ptc

integrity™ lifecycle manager

Installation and Upgrading Guide

11.1

PTC Inc., 140 Kendrick Street, Needham, MA 02494 USA

Contents

I

Installing Integrity Lifecycle Manager

1

Planning Server Installation/ Prerequistites

System and Database Requirements

Supported Operating Systems

Note

For the most current product platform support information, see:

http://support.ptc.com/partners/hardware/current/support.htm

- Integrity Lifecycle Managerserver does not support Microsoft User Account Control (UAC). If you intend to turn off UAC to use Integrity Lifecycle Manager, work with your IT department or Systems Administrator to ensure all necessary steps are taken (for some versions, the Windows registry may be impacted).
- To meet the JRE requirements, you may need to apply upgrades to operating systems that will be running Integrity Lifecycle Manager (including the server, client, and agent).

  Note

  The updates must be applied to the operating system before installing Integrity Lifecycle Manager.

  The list of operating systems and patch levels that support Java 8 can be found in the following location:

  http://www.oracle.com/technetwork/java/javase/certconfig-2095354.html

Supported Browsers

Note

For the most current product platform support information, see http://support.ptc.com/partners/hardware/current/support.htm.

Integrity Lifecycle Manager supports the Microsoft Internet Explorer and Firefox browsers when using the Web clients for workflows and documents, and configuration management.

Supported Languages

Note

For the most current product platform support information, see http://support.ptc.com/partners/hardware/current/support.htm.

The Integrity Lifecycle Manager client, Integrity Lifecycle Manager server, and Integrity Lifecycle Manager Agent are supported in the following languages:

- English
- German
- Japanese
- Simplified Chinese
- Traditional Chinese
- Korean
- French

If a language other than one of those supported languages is specified during installation, English is used for interface text and messages displayed to users.

Creating MS SQL Server Database

Note

For the most current product platform support information, see http://support.ptc.com/partners/hardware/current/support.htm.

This section describes how to create and configure an MS SQL Server database for use with the Integrity Lifecycle Manager server. The instructions assume that you have already set up and tested a supported version of the MS SQL Server database. They also assume that you are familiar with MS SQL Server Administration and the SQL Server Management Studio application.

The following are key considerations when creating an MS SQL Server database:

- Specify the highest supported compatibility level for the version of MS SQL Server you are running.
- If you intend your users to work with large files using the `si add` or `si ci`

commands, it is recommended that you increase the Network Packet Size for the MS SQL database.

- The Integrity Lifecycle Manager server requires the use of row versioning (READ_COMMITTED_SNAPSHOT ON). The server automatically switches to this transaction isolation model for all new server installations.

You can visit the PTC Integrity eSupport portal at http://support.ptc.com/support/integrity.htm. It has useful known issues, tips, and suggestions for installing databases with Integrity Lifecycle Manager server. For example:

- How can I change my Integrity Lifecycle Manager server settings to connect to a different database? (CS215151)

Requirements to Run the Integrity Lifecycle Manager server Installer

In order to run the Integrity Lifecycle Manager server installer, use the `isutil` command line utility or start the Integrity Lifecycle Manager server against a database. The backing database must have:

- a supported version (checked on all databases).
- `ArithmeticAbortEnabled` set to `1` is on or that it can be turned on (MS SQL).
- `TransformNoiseWords` set to `1` or that the database account in use has permission to change the value to 1 (MS SQL).

To create an MS SQL Server database

1. Create an MS SQL Server database, making sure you:
    - Specify a collation that:
        - provides the necessary language processing. For example, if you require to store case and accent information, typically you need a collation that contains the characters `CS_AS`. Or, if you need kana and Japanese width information stored, typically you need a collation that contains the characters `KS_WS`.

        Note

        Integrity Lifecycle Manager Configuration Management requires that you select a collation that stores case and accent (that is, it contains the characters `CS_AS`).

- ○ closely matches the locale your server is running on. If your server needs to store double byte characters, you need to select a locale that is appropriate for the double byte language you need to store.

  For example, an installation of Integrity Lifecycle Manager in a North American environment for SQL Server 2012 could use the `Latin1_General_CS_AS` collation, whereas an installation of Integrity Lifecycle Manager in a Japanese environment could use `Japanese_Unicode_CS_AS_KS_WS`.

- Ensure a reasonable amount of disk space is available for both data files and transaction logs. You can modify this space allocation later, if needed.

2. Create an MS SQL login, making sure you:

   - Specify MS SQL Server authentication.
   - Select the newly created database as the default database for the new login.
   - Assign the `db_owner` database role.

3. Use the Database Maintenance Wizard to establish regular backups for your new database, or refer to MS SQL Server documentation for assistance.

   To improve performance, you can also use the Database Maintenance Wizard to create a database index.

Note

Create a database index only if you are experiencing performance issues. Creating a database index without proper cause may cause database degradation.

4. Run a full backup of the MS SQL Server master database. This ensures your new database is recoverable if anything happens to your server.

Note

You should already have established procedures for regular backups of the master database.

Creating Oracle Database

Note

For the most current product platform support information, see http://support.ptc.com/partners/hardware/current/support.htm.

This section describes how to create and configure an Oracle database for use with the Integrity Lifecycle Manager server. The instructions assume you have already set up and tested a supported version of an Oracle database, and you are familiar with Oracle administration and configuration procedures.

You should already have established procedures for regular backups of your database.

Before you create an Oracle database, note the following:

- Ensure that you have the applicable permissions in order to initialize and maintain the database. For example:
  - The connecting user requires the following system privileges: `CREATE PROCEDURE`, `CREATE TYPE`, `CREATE SEQUENCE`, `CREATE VIEW`, and `CREATE TABLE`.
  - You must also be granted the `execute` permission on the `DBMS_APPLICATION_INFO` system package.
  - Certain diagnostics may also require `execute` on the following system packages but are not required for normal operation of the server: `DBMS_XPLAN`, `DBMS_STATS`, `DBMS_ADDM`, `DBMS_WORKLOAD_REPOSITORY`.
- Your Integrity Lifecycle Manager server Oracle database must generate statistics at sufficiently regular intervals to ensure optimal database performance. If you are unsure if this is being done (or how to configure it to be done), contact your Oracle Database Administrator.
- Before you install an Oracle database on an NT server for use with Integrity Lifecycle Manager, back up the `tnsnames.ora` and `sqlnet.ora` files of your existing applications. If you do not, those files are overwritten and you are not able to access any of your existing applications.
- Ensure that the Oracle Java Virtual Machine (Oracle JVM) is installed as part of the Oracle database installation. Consult your Oracle documentation for more information.

Note

The Oracle Database 12c Release 1 (12.1) JVM supports JDK 6 and JDK 7. The default version is JDK 6. For Oracle 12c, the Integrity Lifecycle Manager platform supports JDK 6. The Integrity Lifecycle Manager platform does not support JDK 7 on Oracle 12c.

- Increase the `open_cursors` initialization parameter from `300` (default) to `2500`.
- When configuring your database, using the Oracle character set with extended characters is recommended, such as `NLS_NCHAR_CHARACTERSET` set to `UTF8`. This ensures that special characters, such as smart quotes, display correctly.
- To ensure that Integrity Lifecycle Manager's duplicate detection works correctly in a Japanese environment, the Oracle database must be configured correctly for the Japanese language/locale. This configuration allows the database lexical analyzer to be aware of boundaries for Japanese words. The machine hosting the Integrity Lifecycle Managerserver must also use a Japanese locale.
- Grant `SELECT_CATALOG_ROLE` to Oracle database users. Doing so ensures that the correct SQL logging and query plan retrieval permissions are available.
- Integrity Lifecycle Manager connects to Oracle using service names. Service names for the Oracle database are available in the `tnsnames.ora` file. Alternatively, you can use the following command to display the service names:

  ```
  SQL> show parameter service_names;
  ```

Note

Before using Integrity Lifecycle Manager server with Oracle 12c, PTC recommends applying Oracle patch 19509982. This Oracle patch addresses a known issue in Oracle 12c (Doc ID 1951689.1, Bug 17376322 “Select Statement Throws ORA-01792 Error”).

To create an Oracle database

1. Set `NLS_NCHAR_CHARACTERSET` to `UTF8`.
2. Create a new tablespace to contain the Integrity Lifecycle Manager data. If you are creating an Oracle database, the tablespace must be created with auto extensions enabled. Allocate sufficient space to contain the data.

   Note

   For text search and text filters to function for your users, the Oracle text package must be installed.

3. Create a new user login name, such as `integrity`, and grant to that user the following:

   Note

   Oracle user names cannot contain dashes (-). If your Oracle database contains user names with dashes, the Integrity Lifecycle Manager server may not start.

   - the role `CTXAPP`.
   - the role `Connect`.
   - system privileges: `CREATE PROCEDURE`, `CREATE TABLE`, `CREATE TYPE`, `CREATE SEQUENCE` (Oracle 11g Release 2 32-bit and 64-bit, Oracle 12c Release 1 32-bit and 64-bit), and `CREATE VIEW` (Oracle 11g Release 2 32-bit and 64-bit, Oracle 12c Release 1 32-bit and 64-bit).
4. Set the default tablespace for the newly created user to the tablespace name created in step 1.
5. Grant the `SELECT` on `CTXSYS.CTX_INDEXES` object privilege to the newly created user.
6. Establish regular backups for your new database tablespace. It is important to ensure your data is properly backed up.

Considerations for Installing Integrity Lifecycle Manager server

Before performing a new installation of the Integrity Lifecycle Manager server, you need to do the following:

- Obtain your required license file from PTC – Integrity Lifecycle Manager Services or PTC Technical Support (`license.dat`) and place it in a directory where it can be picked up during the installation. For more information, see “FlexNet License Files” on page 30.
- When installing Integrity Lifecycle Manager server on Windows, ensure you preserve existing system settings by manually creating a system restore point. Creating a restore point allows for system restoration in the case of error. For information on creating restore points, refer to http://www.microsoft.com and find the restore procedure for the version of Windows you are running.
- Ensure that the time set on the Integrity Lifecycle Manager server is identical to the time set on the database.

  Note

  NTP time adjustments such as Daylight Saving Time are handled by the Integrity Lifecycle Manager server.

  Integrity Lifecycle Manager server does not start if it encounters a significant time shift during startup. If the server finds such a time shift while running, it does not commit the related operation. If any of these problems occur on Integrity Lifecycle Manager, contact PTC Technical Support.
- Consider whether to install Integrity Lifecycle Manager server with the secure port (which uses TLS encryption protocols to secure every connection to the server) or with the clear port (which connects without encryption). PTC recommends using the secure port wherever possible for increased security. For more information about configuring Integrity Lifecycle Manager to use the secure port, see the “Secure Sockets Layer” topic in the *Integrity Lifecycle Manager Help Center*.

If you are installing a Japanese locale Integrity Lifecycle Manager server, consider the following:

As of Integrity 10.4 or later, there is support for English locale Integrity Lifecycle Manager clients connecting to Japanese locale Integrity Lifecycle Manager servers. However, type properties on the server must not explicitly specify the `Structure` field (such as `fields=Structure` and `outlineColumns=Structure` found in the `MKS Solution` type). In such a configuration, the English Integrity Lifecycle Manager client cannot interpret the Japanese version of the word `Structure` (構造). This issue is primarily present in solution implementations, and the following is a list of known type properties whereby it is recommended to remove the key/value pair (do not remove the property) that includes the `Structure` field:

```
MKS.RQ.item.addrelatedtests.findersettings
MKS.RQ.rm.content.createtrace.finderSettings
MKS.RQ.rm.content.includecontent.finderSettings
MKS.RQ.rm.content.includedocument.finderSettings
MKS.RQ.rm.content.insert.content.finderSettings
MKS.RQ.rm.content.insertcontent.finderSettings
MKS.RQ.rm.content.insertdocument.finderSettings
MKS.RQ.rm.document.createfromtemplate.finderSettings
MKS.RQ.rm.document.includedocument.finderSettings
MKS.RQ.rm.document.insertdocument.finderSettings
MKS.RQ.rm.document.open.finderSettings
MKS.RQ.rm.document.viewdocument.viewSettings
MKS.RQ.rm.document.open.viewSettings
```

Note

Field names and messages coming from the Japanese locale Integrity Server will display in Japanese to English locale Integrity Clients, and the default Integrity Help language for the Integrity Lifecycle Manager client is determined by the server language.

System Considerations

- Ensure that the Operating System and all applications running on the machine (s) you are installing Integrity Lifecycle Manager server on have the latest patches and service packs installed or you may encounter issues with your Integrity Lifecycle Manager server installation. To see a list of supported operating systems and applications, see "Supported Operating Systems" on page 10.
- Make sure your system has sufficient disk space and memory. The installation program does not report an error when your system is running out of disk space.
- To meet the JRE requirements, you may need to apply upgrades to operating systems that will be running Integrity Lifecycle Manager (including the server, client, and agent).

Note

The updates must be applied to the operating system before installing Integrity Lifecycle Manager.

For Solaris/Linux/Windows, the list of operating systems and patch levels that support Java 8 can be found in the following location:

http://www.oracle.com/technetwork/java/javase/certconfig-2095354.html

Database Considerations

- PTC recommends that you create and configure the required supported databases prior to beginning the installation process.
- The legacy RCS repository is no longer supported for configuration management as of Integrity 10.0 and later. To support a more robust repository type for configuration management, Integrity exclusively supports the database repository.

  The removal of RCS repository support includes the following changes:

  Installation

  When installing the Integrity Server, RCS is no longer an option as a repository type. An Integrity Server 10.0 or later, configured for the RCS repository fails to start.

  Licensing

  The separate database repository license controlling usage of the database repository is removed as of Integrity 10.0 and later. A configuration management license implies use of the database repository.

  Server Configurations

  There are three basic server configurations available in Integrity 10.0 and later versions:

  - An Integrity Server that functions as a complete server and uses a database repository. This requires an Integrity Server license.
  - An Integrity Server that functions as a complete server and uses a database repository, and an Integrity Server that functions as a complete Integrity Server and a proxy server and uses a database repository. This requires an Integrity Server and Integrity Proxy Server license.
  - An Integrity Server that functions as a complete server and uses a database repository, and an Integrity Server that functions as a proxy-only server. This configuration is new as of Integrity 10.0 and later versions, and the proxy-only server is used exclusively for proxying other servers; it does not use a repository. This requires an Integrity Proxy Server license.

Note

Due to concerns about scalability and robustness, PTC recommends that you do not use the embedded Derby database for production servers.

Commands, Options, Policies, Properties, and Keywords

In Integrity, commands, options, policies, properties, and keywords specific to the RCS repository are unavailable or ignored. For example, when connecting to a proxy-only server, the following nodes are disabled in the Integrity Lifecycle Manager administration client: `ViewSet Distribution`, `Workflows and Documents`, `Configuration Management`, and `Deploy`. All corresponding CLI commands are also disabled for a proxy-only server, for example, `im createtype` and `im editreport`.

Note

In Integrity 10.8 and later, the Deploy node functionality is no longer supported.

If you are currently using an RCS repository for configuration management, you must migrate to a database repository in your current release of Integrity before upgrading to Integrity 10.0 or its later versions. To migrate an RCS repository to a database repository, see the *Integrity Database Repository Migrator Guide* in your current release of Integrity.

If you are currently using a database platform that was supported by the RCS repository in previous releases of Integrity but is not currently supported by the database repository, you must migrate to a supported database platform. For the most current product platform support information, see http://support.ptc.com/partners/hardware/current/support.htm.

- If you are currently using MKS Integrity 2007 at any service pack level and have implemented the MKS Requirements 2007 solution template or a derivative thereof and are upgrading to Integrity 10.0 or its later versions, you must migrate your database to MKS Integrity 2009 first. For more information on migrating an MKS Requirements 2007 solution template to MKS Integrity

2009, see the *MKS Requirements 2007 to MKS Integrity 2009 Solution Migration Guide* on http://support.ptc.com/support/integrity.htm.

- The configuration management database repository relies on the time set on Integrity Lifecycle Manager server. As a result, the time on the server machine should be correctly set prior to installing the Integrity Lifecycle Manager server. Once the Integrity Lifecycle Manager server is installed, the time on the server machine should not be arbitrarily changed. Minor time-keeping corrections, such as Network Time Protocol (NTP) adjustments, are acceptable; however, significant time shifts could cause a future transaction to display to have occurred prior to an existing committed transaction. For example, if the time set on the Integrity Lifecycle Manager server differs from the time set on the database, date and time based queries may fail.

Multiple Server Considerations

Determine how many Integrity Lifecycle Manager servers you require. There are a number of possible multiple server configurations.

- Separate standalone servers

  You can use separate servers for the components for workflows and documents, and for configuration management, or you can use separate servers for different departments in your organization.

- Proxy servers

  Proxy servers are used with the Federated Server™ architecture (FSA) to address the needs of geographically dispersed organizations but can also be configured in organizations with or without geographically dispersed development to improve network access and speed. For more information on FSA proxies, see the “Understanding Federated Server Architecture” in the *Integrity Lifecycle Manager Help Center*.

System Requirements

Before installing the Integrity Lifecycle Manager server or Integrity Lifecycle Manager client, you should be aware of the recommended system requirements. These requirements are designed to give you optimal system performance. However, individual performance may vary depending on actual system components in use. To help determine what your particular system requirements are, contact PTC Technical Support. Many factors, such as number of users and number of projects, need to be assessed in configuring the optimal system.

Server Licensing

License Process

Integrity Lifecycle Manager uses Flexera’s FlexNet® network license server manager to control the use of the Integrity Lifecycle Manager components. For new installations, you must obtain your `license.dat` file and place it in a temporary directory before installing the Integrity Lifecycle Manager server so that it can be checked during the installation process.

Note

During the installation process, the license file is copied to the *`installdir`*`/data/license` folder and is used from there unless you specify another location in the `mksis.licensePath` in the `is.properties` file.

The actual number of licenses you receive depends on your particular requirements and what you have purchased.

Your FlexNet License Files

Your license administrator controls who uses the licensed application(s) and the seat(s) where the licenses are available. For details on the various types of license files, see “FlexNet License Files” on page 30, and “Supported FlexNet License Types” on page 28.

FlexNet Model

FlexNet has four main components: the license server manager, the Integrity Lifecycle Manager daemon, the license file, and the application program.

License Server Manager

The FlexNet license server manager (`lmadmin`) handles the initial contact with the Integrity Lifecycle Manager components passing the connection on to the appropriate Integrity Lifecycle Manager daemon. It also starts, stops, and restarts the Integrity Lifecycle Manager daemons.

Integrity Lifecycle Manager Daemon

In FlexNet, licenses are granted by a running process. For Integrity Lifecycle Manager products, this process is called the Integrity Lifecycle Manager daemon. The Integrity Lifecycle Manager daemon keeps track of how many licenses are checked out, and who has them.

The licensed application communicates with the Integrity Lifecycle Manager daemon. The application and daemon process (the license server) can run on separate nodes on your network across any size wide-area network. The format of the traffic between the client and the Integrity Lifecycle Manager daemon is also machine independent allowing for heterogeneous networks. This means the license server and the computer running the Integrity Lifecycle Manager server can be run either on different hardware platforms or even on different operating systems (Windows and UNIX, for example).

If you install your license server on a separate machine, you need to redirect your server to that machine to find the license file.

Note

If you want to use FlexNet to support multiple license servers, the recommended approach is to have one process for each vendor who has FlexNet-licensed components on the network. This also means you have to manage multiple vendor daemons.

License File

Licensing data is stored in a text file called the license file. You must place it on the license server in `*installdir*/data/license`.

If you change the location of the license file, you need to change the following in the `*installdir*/config/properties/is.properties` file:

```
mksis.licensePath=installdir/data/license/license.dat
```

If you install the license server on a machine other than the Integrity Lifecycle Manager server, you must request a redirector license file. The redirector license file must be placed on the Integrity Lifecycle Manager server and the standard license file must be placed on the remote license server. The redirector license file points the Integrity Lifecycle Manager server to the remote license server for licensing information.

The license file contains the following information:

- host IDs specified in the `SERVER` line(s)
- vendor line
- at least one line of data (called `FEATURE` or `INCREMENT` lines) for each licensed Integrity Lifecycle Manager component (the data in each `FEATURE` line has a license key)

Application Program

The Integrity Lifecycle Manager server (the application program) is linked with the program module that provides the communication with the license server. During execution, the Integrity Lifecycle Manager server communicates with the FlexNet license server manager to request a license.

Supported Platforms

The platforms that support the Integrity Lifecycle Manager server may not be supported by the FlexNet server; consult Flexera for a list of supported platforms. For more information on supported platforms for the Integrity Lifecycle Manager server, see "Supported Operating Systems" on page 10. If FlexNet does not support the platform your Integrity Lifecycle Manager server is running on, run the FlexNet server on a separate system.

Installing FlexNet License Server

Integrity Lifecycle Manager licenses are managed using FlexNet Publisher 11.13.1.3, which includes the `lmadmin` license server manager. The FlexNet license server manager supports a client connection over HTTP and also provides a Web-based interface for administrative tasks.

Note

This version of the FlexNet Publisher License Server installer does not include a Java Runtime Environment (JRE). You must install JRE 1.5 or higher before installing FlexNet. The JRE can be downloaded from http://www.java.com. For more information, refer to the FlexNet Publisher documentation included with the FlexNet installer.

1. Insert the Integrity Lifecycle Manager DVD from your package into the DVD drive.

   On Windows and on certain UNIX systems, the Integrity Lifecycle Manager DVD browser starts automatically and allows you to select from the following options:

- **Installation** allows you to install the Integrity Lifecycle Manager server, Integrity Lifecycle Manager client, FlexNet license server, and Integrity Lifecycle Manager Agent.
- **Documentation** allows you to view the product documentation in Adobe Acrobat PDF format.
- **Release Notes** allows you to view the release notes for Integrity Lifecycle Manager.

  To manually start the browser, on the Integrity Lifecycle Manager server DVD open `index.html` in the `/documentation` directory.

2. To start the installation process, from the Integrity Lifecycle Manager DVD browser click **INSTALL**. A page displays allowing you to select a component to install. In the **License Server** section, do one of the following:

   - For Windows, click the link, then specify whether you want to extract or save the `flexnet.zip` file.
   - For UNIX platforms, follow the written instructions.

   > **Note**
   >
   > The FlexNet installers for UNIX are graphical. Before starting the installation on UNIX you must set the `$DISPLAY` environment variable.

   If you do not want to use the browser you can extract `flexnet.zip`, which is located on the Integrity Lifecycle Manager DVD in the platform-specific `/installs/flexnet` directory.

3. Follow the setup instructions for your particular platform.

   For Windows, run `setup.bat`.

   For UNIX platforms, run `setup.sh` to setup the FlexNet server.

Setting Up the FlextNet License Server

To set up a simple FlexNet License Server, perform the following steps:

1. If you have not already done so, unpack the `flexnet.zip` file into a secure directory on the license server machine.
2. Run the setup script `setup.bat` on Windows, or `setup.sh` on a Posix compatible OS.

On Windows, the setup script launches the `lmadmin` installer. Once the installation completes, copy `lmutil.exe` and `MKS.exe` into the directory you chose for the `lmadmin` installer.

On UNIX, the setup script launches the `lmadmin` installer, then prompts you for information to configure the appropriate files in the `lmadmin` installation directory, including `lmutil` and `MKS`.

Note

FlexNet does not require root permissions and thus should not be run as root.

3. Once the installation completes, start the FlexNet license server if it is not already running.

   On Windows, start the FlexNet license server by running the following executable in the `lmadmin` installation directory:

   ```
   lmadmin.exe
   ```

   On UNIX, start the FlexNet license server by running the following executable in the `lmadmin` installation directory:

   ```
   lmadmin
   ```

Note

To configure FlexNet to start automatically on UNIX, refer to the FlexNet administration documentation included in the `flexnet.zip` file.

4. Launch the Web configuration tool to configure the license server. By default, the server listens on port 8090 (this can be customized in the installer). The default login and password is `admin/admin`. When you login for the first time, you are prompted for a new password.

FlexNet Licenses

Now that you have installed and set up the FlexNet license server, you can start running the Integrity Lifecycle Manager components using the licenses you received from your Integrity Lifecycle Manager Sales Representative. The number of licenses you receive depends on your particular requirements and what you have purchased.

Obtaining FlexNet Licenses

If you did not receive your license file with Integrity Lifecycle Manager, you should do the following:

1. Make sure you know which machine you intend to run the FlexNet server on.

   Before running any FlexNet licensed program, you need to set up your license server machine. You must select which machine to run your license server on and provide the host ID to PTC Technical Support.

   Note

   - To obtain the host ID of the server machine, run the FlexNet License Server Manager Web interface (`lmadmin`). Under the **System Information** tab, the required host ID displays in the **Ethernet Address** field.
   - If the FlexNet server is running on Solaris, the IP address is required instead of the host ID.

2. Provide your Integrity Lifecycle Manager Sales Representative or PTC Technical Support with your host information in order to get a license file. You can use any of the following methods to request your licenses from PTC Technical Support:

   - Log on to the PTC Integrity eSupport portal and complete the online request.
   - Call 1-800-219-4842.

     After your request is received and your payment confirmed, a license file is provided to you (generated using the host ID information you provided in your request).

3. Install your license file in the license file location on your system. By default, the location is:

   `*installdir*/data/license/license.dat`

   You can also consider combining the new license file with any existing license files.

4. Determine if an options file is desired, and if so, set it up. The options file controls various options such as reservations and timeouts of licenses.

Tip

Most installations can run without an options file; however, you may decide you want to use certain options. For example, administrators may use an option to limit the quantity and content of logged messages.

5. To start the license server, start up the license server manager, `lmadmin`. Under the **Vendor Daemon Configuration** tab, select `MKS` from the list. The **Vendor Daemon:MKS** panel displays.
6. Under **Vendor Daemon Actions**, click **Start**.

Tip

On Windows, you can also run the license server from the Windows Services Control Panel.

7. To use the license file, start the Integrity Lifecycle Manager server.

Supported FlexNet License Types

Integrity Lifecycle Manager supports concurrent and seat licenses. In general, seat licenses are bound to individual users, whereas concurrent licenses can be shared by multiple users. Seat and concurrent licenses can be mixed in the same license file.

Concurrent Licenses

A concurrent license permits anyone with login permission for the application to use the licensed software component up to the limit specified in the license file. Concurrent licenses can be used for the graphical user interface (GUI) and command line interface (CLI) clients, as well as for the Web clients made available by the Integrity Lifecycle Manager server. The GUI and CLI clients include both the user and administration clients.

Application Timeout and Idle Disconnect

You can specify the number of minutes without any transactions that the Integrity Lifecycle Manager server waits before deciding the Integrity Lifecycle Manager client is no longer active. When the time expires, each concurrent license in use and connection to the server are released, independent of one another. By default, an idle client does not release licenses or disconnect from the server. The minimum idle disconnect time for the client is set at 60 minutes.

To force the clients to release the concurrent license and disconnect from the server when the client is idle, configure the `idleDisconnectTimeout` property found in the Integrity Lifecycle Manager client.

The following property sets the idle timeout on all clients:

```
mksis.idleDisconnectTimeout=value
```

where *value* is a static value in minutes, loaded at boot time, applying to both the user and administration clients.

You can accept the default value or increase the idle disconnect time by uncommenting the property and setting the desired value. If you set a value of less than 60 minutes, the minimum default of 60 minutes still applies.

Web Client Timeouts

The Web clients made available by the Integrity Lifecycle Manager server include one for workflows and documents, and one for configuration management. By default, Web clients remain connected as long as the Web browser session remains active.

Timeout settings for the Web clients are configured in the Integrity Lifecycle Manager client.

To configure the Web client for workflows and documents (default is set to 300), edit the following property under **Workflows and Documents ▸ Configuration ▸ Properties**:

```
mksis.im.httpSessionTimeout=value
```

To configure the Web client for configuration management (default is set to 3600) edit the following property under **Configuration Management ▸ Configuration ▸ Properties**:

```
mksis.si.httpSessionTimeout=value
```

where *value* is the static value in seconds.

Pooling Concurrent Licenses Through FlexNet

You can create your own license pools for concurrent licenses through FlexNet. For example, you can use license pooling in the following scenarios:

- A common Integrity Lifecycle Manager server is shared among several departments or business units, with each department or business unit paying for a portion of the concurrent licenses.
- Several groups of users share concurrent licenses, with users in the lower priority group allocated a smaller number of licenses.

To implement this type of license pooling, you need to do the following:

- request `INCREMENT` lines to be added to your license file for each group of licenses
- add an `INCLUDE` line to your options file to associate a group of licenses with a defined group of users

For more information, see the FlexNet documentation on the Flexera Web site at:

http://www.flexerasoftware.com

Seat Licenses

Seat means the licensed software component can only be used by the licensed user. The groups for workflows and documents, and configuration management used to acquire the appropriate seat licenses are specified in the file:

```
installdir/config/properties/is.properties
```

where *installdir* is the path to the directory where you installed the Integrity Lifecycle Manager server.

You can control who gets seat licenses by setting the license group properties in `is.properties` to a single group in the authentication realm that contains everyone in the company requiring a seat license. For example:

```
mksis.imLicenseGroup=SeatLicenseGroup
mksis.siLicenseGroup=SeatLicenseGroup
```

Provided you have sufficient seat licenses, you do not need to restart the Integrity Lifecycle Manager server when you add users to the group. The required files are dynamically updated on the Integrity Lifecycle Manager server.

Note

If your security realm is case sensitive, the naming of seat licenses is case sensitive.

FlexNet License Files

Your license administrator controls who uses the licensed application(s) and the seat(s) where the licenses are available. Depending on the products you purchase, you receive:

- Standard license file
- Redirector license file (if using a remote license server)
- Integrity Lifecycle Manager server license
- Integrity Lifecycle Manager server proxy license
- Concurrent licenses:

    - Integrity Lifecycle Manager for workflows and documents
    - Integrity Lifecycle Manager for configuration management
- Seat licenses:
    - Integrity Lifecycle Manager for workflows and documents
    - Integrity Lifecycle Manager for configuration management
    - Integrity Lifecycle Manager for Application Service Management

As part of your initial purchase, you start with a set of fully functional licenses that allows you full use of the products you purchased. For more information on licenses, see “FlexNet Licenses” on page 26.

Note

- The embedded Derby database does not require a license.
- Licenses are assigned to a particular host machine. If for any reason the host machine needs to be changed or becomes disabled, you need to get a replacement license.

Server Security

Security Environment for Configuration Management Overview

The security environment for configuration management is divided into three major components:

- server-based security scheme for authenticating users and authorizing access based on permissions
- data management subsystem for maintaining the underlying security ACL database
- distributed management application (the Authorization Administration program) for providing a variety of administrative user interfaces to the security ACL database, including the Integrity Lifecycle Manager administration client interface

Choosing Your Security Policy

Your security policy specifies the security scheme for your users. If you are using the same security scheme for all users, all you need to do is specify the authentication domain and transport protocol you want to use as your default.

In `security.properties`, uncomment `mks.security.policy.scheme.default` and specify your default policy, for example:

```
mks.security.policy.scheme.default=mksdomain_clear
```

Authentication Domain	Transport Protocol	Security Policy
Kerberos	Clear	`windows_clear`
Kerberos	Private	`windows_private`
Kerberos Single Sign-on	Clear	`windowsSSO_clear`
Kerberos Single Sign-on	Private	`windowsSSO_private`
MKS Domain	Clear	`mksdomain_clear`
MKS Domain	Private	`mksdomain_private`
LDAP	Clear	`ldap_clear`
LDAP	Private	`ldap_private`
UNIX	Clear	`unix_clear`
UNIX	Private	`unix_private`

Note

- The default security scheme for Windows systems is `nt_clear`. The NT and NTSS security realms are supported for this release but will be dropped in a future release.
- The default security scheme for UNIX systems is `unix_clear`.
- If you only have a `clear` or `secure` port specified in `is.properties`, your security scheme must use the appropriate communication method for that port.

MKS Domain Security Realm

The MKS Domain is an Integrity Lifecycle Manager security realm. The specific database scheme stores users and group names in the database.

Possible uses for this security realm are as follows:

- Use MKS Domain groups when you want to include users and groups from multiple realms as group members.
- Use MKS Domain users for small teams, or for temporary or visiting users.

Enabling the MKS Domain

Before using the MKS Domain, you must enable it in the following file:

```
installdir/config/properties/security.properties
```

For more information on the contents of the `security.properties` file, see the “Security Schemes” topic in the *Integrity Lifecycle Manager Help Center*.

Administering the MKS Domain

Integrity Lifecycle Manager server imports your existing users and groups into the MKS Domain. However, if you do not have existing users and groups, you can administer the MKS Domain (setting up your users and groups) through the Integrity Lifecycle Manager client(using a “bootstrap” method).

First, start the Integrity Lifecycle Manager server in safemode; then create users and groups.

Importing Users and Groups

Each time a server is restarted, the Integrity Lifecycle Manager server automatically compares lists of users and groups stored in text files with MKS Domain data. The server then imports users and groups that are in the files but are not existent in MKS Domain.

The users and groups are respectively located in the following files `installdir/data/password.properties` and `installdir/data/group.properties`, where *installdir* is the directory the Integrity Lifecycle Manager server is installed in. Those files are not required for setting up the MKS Domain. The ability to import users and groups is provided as a convenience to import large numbers of them at a time.

The syntax of the `password.properties` file is:

```
user1=password1
user2=password2
```

The syntax of the `group.properties` file is:

```
group1=user1,user2,...
group2=user3,user4,...
```

Note

After being successfully imported, the server deletes both `password.properties` and `group.properties`. A property file is not deleted if a user or group in it fails to import.

Spaces used in user and group names must be preceded by the escape backward slash (\), for example, `John\ Riley=Password`, or they do not import correctly.

The MKS Domain is an Integrity Lifecycle Manager security domain. This database scheme stores users and group names in the database.

MKS Domain can be administered through the GUI and the CLI interfaces. For information on administering the MKS Domain through the CLI, see the CLI man pages.

Consider the following when importing users and groups:

- All user and group information is stored in the Integrity Lifecycle Manager server database.
- Empty passwords for users are not permitted.

MKS Domain Passwords

Passwords for users in the MKS Domain are securely stored in the Integrity Lifecycle Manager database as one-way, salted, hash values.

The salt value is automatically generated during install, upgrade, or service pack application, and located in the `mksis.serversalt` property in the file *`installdir`*`/config/properties/encryption.properties`, where *installdir* is the directory the Integrity Lifecycle Manager server is installed in.

Key Considerations

- When installing or upgrading to Integrity 10.1, all existing passwords in the MKS Domain tables will be decrypted and then hashed using the server salt and existing password. The new hash values will be stored in the existing Password column. These actions will occur during the first startup of the Integrity Lifecycle Manager server after upgrading or installing a service pack.
- If the server salt is lost or modified, all passwords in the MKS Domain are set to invalid. The only way to reset the invalid passwords is to generate a new server salt.

⚠ **Caution**

The `mksis.serversalt` property is not a user-maintained property. Attempting to modify an existing salt value will invalidate all existing passwords and thereby require all passwords to be reset.

- The server salt value can be obscured by using the `encryptPassword` command.

Generating a New Server Salt

Generating a new server salt requires an Integrity Lifecycle Manager administrator.

- If the administrator authenticates using the MKS Domain, then the new salt must be generated with the Integrity Lifecycle Manager server in safe mode. For more information on this topic, see “Starting Server in Safe Mode” on page 81.
- If the administrator authenticates using LDAP, then safe mode is not required.

  To generate a new server salt

Use the following `isutil` command:

```
-c generateserversalt
```

A prompt displays and requires you to confirm your choice to generate a new server salt. A message confirms that generating a new server salt invalidates all existing passwords and could affect Integrity Lifecycle Manager server operations.

For more information on `isutil` commands, see “Using isutil to Manage the Database Repository” on page 119.

MKS Domain Permission

Administering the MKS Domain requires the `mks:system:mksdomain` ACL with the `AdminServer` permission. By default, the `mks:system:mksdomain` ACL is denied to the everyone group.

To set this permission from the Integrity Lifecycle Manager client, in the tree pane select **MKS Domain** ▸ **Permissions** ▸ **Global**. Then add a principle. For more information, see “Adding a New Principal and ACL Entry” topic in the *Integrity Lifecycle Manager Help Center*.

To set this permission from the CLI, use the following command:

```
aa addaclentry --acl=mks:system:mksdomain u=<username>:AdminServer
```

where `<username>` is the user name of the user who is assigned permission to view the MKS Domain node in the Integrity Lifecycle Manager client.

For more refined control over MKS Domain groups, Integrity Lifecycle Manager allows you to create restricted groups which cannot be administered using the `AdminServer` permission. Administering restricted groups requires the `RestrictGroup` permission under the `mks:system:mksdomain` ACL. This permission can be granted to a principal along with the `AdminServer` permission. Only those principals who have the `RestrictGroup` permission can create, edit, and delete the MKS domain groups marked as **Restricted**. For more information, see the "Managing MKS Domain Groups" topic in the *Integrity Lifecycle Manager Help Center*.

Note

Assigning the `AdminServer` permission to a user gives that user permissions to view, edit, and delete both users and groups which are not marked as **Restricted** in the MKS Domain.

A user does not need the `RestrictGroup` permission to view a restricted group. However, for creating, editing, and deleting a restricted group, the `RestrictGroup` permission must be granted to the principal.

LDAP

If you are using a security scheme with a Kerberos, Kerberos Single Sign-on, or LDAP authentication domain, you must set up your security realm. To set up your security realm, do the following:

- Set up the properties for your realm.
- Review the batch size for the number of entries returned by the directory server.
- If required, set up the realm to use a Secure Sockets Layer.
- Review the failover settings.

Setting Up Security Realm Properties

You set up properties for your LDAP-compliant security realm in the `security.properties` file. Typical settings are preconfigured for each of the following supported realms:

- OpenLDAP server
- Microsoft Active Directory Services (ADS)

- Sun ONE Directory Server
- RFC 2307-based schemas on all supported servers
- Novell Directory Services

To set up the Integrity Lifecycle Manager server to communicate with your security realm, uncomment the properties that correspond to your security realm, and then edit the following properties for server, user, group, and membership.

You should already be familiar with how the LDAP-compliant security realm is implemented on your system. At a minimum, you should be familiar with Distinguished Names (DN), LDAP search filters, and LDAP schemas.

For more information on the properties for your security realm, refer to the LDAP documentation or to some of the resources available on the Web.[1]

Note

The Integrity Lifecycle Manager server provides support for the password expiry feature of the LDAP v3 security realm. The only LDAP server that supports this functionality is Sun ONE Directory Server.

Note

When configuring the LDAP authentication realm in the Integrity Lifecycle Manager server `security.properties` file, you must use quotation marks to enclose the `.dn` value if any of the user or group names contain spaces. The following provides an example of the correct syntax to use if the defining list contains spaces: `ldap.user.dn="ou=Engineering,ou= Quality Assurance"`.

[1] General LDAP documentation
http://www.umich.edu/~dirsvcs/ldap/doc/
Microsoft Active Directory Server
http://www.microsoft.com/windows2000/technologies/directory/ad/default.asp#section8
OpenLDAP
http://www.openldap.org/
Novell Directory Services for Windows
http://developer.novell.com/edirectory/ndsldap.htm
RFC 2307
http://www.ietf.org/rfc/rfc2307.txt

LDAP Bind Credential Properties

These settings establish the LDAP bind credential used when enumerating users and groups.

Property	Description
`ldap.host`	Host name (or IP address) of LDAP server.
`ldap.port`	LDAP server port to connect to. By default, `server.port` is `389` for connections using clear protocol and `636` for connections using private protocol.
`ldap.principal`	Distinguished Name (DN) of user/principal used to connect to LDAP server. Principal should be an unprivileged user (that is, principal should have read-only access).
`ldap.credential`	Password of above user/principal.

Following LDAP Referrals

If you want to follow LDAP referrals, you need to specify the additional server addresses using the following format:

```
ldap.host.1=<host1>
ldap.principal.1=<principaluser>
ldap.credential.1=<principleuserpassword> ldap.host.2=<host2>
ldap.principal.2=<principaluser>
ldap.server.credential.2=<principaluserpassword>
```

If a referral is not specified in this list, it is not followed.

The host name is looked up using DNS, and the failover mechanism applies.

User Properties

These settings define where to find users in the directory.

Property	Description
`ldap.user.dn`	One or more base Distinguished Names (DN) for searching users.
`ldap.user.filter`	LDAP search filters to match user entries (where `%u` is substituted for user).
`ldap.user.scope`	Range for searching users. Allowed values are `subtree`, `one-level`, or `base`. By default, `ldap.user.scope=subtree`.
`ldap.user.name`	Name or user ID of user.

Property	Description
	The default is `samAccountName`. For User Principle Name (UPN), specify `userPrincipalName`.
`ldap.user.displayname`	Full name for user. Specifying this property turns on full name. For more information, see the “Full User Name Support” topic in the *Integrity Lifecycle Manager Help Center*.
`ldap.user.e-mail`	E-mail address for user. For more information, see the “E-mail Address Support” topic in the *Integrity Lifecycle Manager Help Center*. Property not defined for rfc 2307 realms.
`ldap.user.objectclass`	Object class value that indicates object is user.

Limiting Number of Users Returned

Some LDAP-compliant security realms do not allow queries with more than 1000 results. If you have a large number of users, you should set up multiple DNs to send multiple queries with smaller results. For example, if there are smaller contexts with less than 1000 users, you could create several, more specific DNs:

```
   user.dn.1=ou=support,ou=users,dc=northamerica,dc=support,dc=com
   user.dn.2=ou=boston,ou=users,dc=northamerica,dc=support,dc=com
   user.dn.3=ou=qa,ou=users,dc=northamerica,dc=support,dc=com
   user.dn.4=ou=development,ou=users,dc=northamerica,dc=support,dc=com
```

You only need to list users connecting to the Integrity Lifecycle Manager server.

Configuring LDAP for Multiple Domain Name Definitions

If you have multiple domain name definitions, you must configure your LDAP settings correctly to avoid potential security vulnerabilities. This section describes two scenarios for configuring LDAP for multiple domain name definitions.

Scenario 1: All domain name definitions contain the same filter and scope

In this scenario, the administrator defines the filter and scope once and then Integrity Lifecycle Manager automatically configures all domain names to use that filter and scope.

For example:

```
ldap.user.dn.1=OU=Developers,DC=company,DC=com
ldap.user.dn.2=OU=Management,DC=company,DC=com
ldap.user.dn.3=OU=Contractors,DC=company,DC=com
ldap.user.filter="(&(samaccountname\=%u)
     (objectclass\=user)(objectcategory\=person))"
ldap.user.scope=subtree
```

Scenario 2: Some domain name definitions contain different filters or scope

In this scenario, the administrator defines a separate filter and scope for each domain name, with the filter and scope numbered to match the appropriate domain name.

Note

If you do not provide a matching filter or scope for a domain name, a domain name number, or the numbering is non-sequential (`dn.1`, `dn.2`, `dn.4`, where `dn.3`is missing), the Integrity Lifecycle Manager server does not start.

For example:

```
ldap.user.dn.1=OU=Developers,DC=company,DC=com
ldap.user.dn.2=OU=Management,DC=company,DC=com
ldap.user.dn.3=OU=Contractors,DC=company,DC=com

ldap.user.filter.1=(&(samaccountname=%u)
     (objectclass=user)(objectcategory=person))
ldap.user.filter.2=(&(samaccountname=%u)
     (objectclass=user)(objectcategory=person))
ldap.user.filter.3=(&(mail=%u)
     (objectclass=user)(objectcategory=person))

ldap.user.scope.1=subtree
ldap.user.scope.2=subtree
ldap.user.scope.3=base
```

Group Properties

These settings define where to find groups in the directory.

Property	Description
`ldap.group.dn`	One or more base Distinguished Names for searching groups.
`ldap.group.filter`	LDAP search filters to match group entries (where `%g` is substituted for group).
`ldap.group.scope`	Range for searching groups. Allowed values are `subtree`, `one-level`, or `base`. By default, `ldap.group.scope = subtree`.
`ldap.group.name`	Name of group.

Property	Description
`ldap.group.members`	Root LDAP directory that contains list of members for group.
`ldap.group.objectclass`	Object class value that indicates object is group.

Group Member Settings for Kerberos and Kerberos Single Sign-on

If you are using a security scheme with a Kerberos or Kerberos Single Sign-on authentication domain (`windows` or `windowsSSO` security policy in the `security.properties` file), keep the following points in mind:

- If you are using ADS for group lists, do not use group names that start with the "#" symbol.
- Do not set the seat license group as the user's primary group. Setting the seat license group as the primary group means the search filter cannot find the user and, therefore, cannot assign a seat license. The user does not get a seat license, but does get a concurrent license, if one is available.

Member Properties

These settings define where to find members of groups in the directory.

Property	Description
`ldap.member.dn`	One or more base Distinguished Names for searching group members (where `%M` is substituted with value of member name/DN for group).
`ldap.member.filter`	Filter to resolve member (where `%M` is substituted with value of member name/DN for group).
`ldap.member.scope`	Range for searching members. Allowed values are `subtree`, `one-level`, or `base`.

Organizational Unit Properties

These settings define the object class name and display name for an organizational unit.

Property	Description
`ldap.ou.objectclass`	Object class name for organizational unit.
`ldap.ou.name`	Display name for organizational unit.

Performance Tuning

If you are using an LDAP-compliant security realm, the following property in `security.properties` could affect the performance of your directory server:

```
ldap.batchsize
```

This property specifies the maximum number of entries that are passed back at one time from the directory server to the Integrity Lifecycle Manager server. The default value is `100`.

Using SSL

If you are using an LDAP-compliant security realm, the following property in `security.properties` specifies if a Secure Sockets Layer (SSL) is used to communicate with the directory server:

```
ldap.ssl
```

By default, this property is set to `false`.

For more information on using SSL, see the “Secure Sockets Layer” topic in the *Integrity Lifecycle Manager Help Center*.

Failover of Directory Servers

If you are using an LDAP-compliant security realm, the Integrity Lifecycle Manager server supports the use of multiple directory servers to handle authentication when one server fails. The Integrity Lifecycle Manager server uses the Domain Name Service (DNS) list to find all directory servers associated with the server host name and authenticates to the first server in the list that responds to the connection request. To minimize waiting time, the Integrity Lifecycle Manager server maintains a pool of the directory servers that it has connected to.

If you are using failover, you should review the following cache settings in `is.properties:`

```
java.security.property.networkaddress.cache.ttl
java.security.property.networkaddress.cache.negative.ttl
```

For more information on working with `is.properties`, see the “Integrity Lifecycle Manager Server Configuration Properties” topic in the *Integrity Lifecycle Manager Help Center*.

You should also review the timeout settings used by Integrity Lifecycle Manager server when trying to connect to a directory server. These settings are specified in the `security.properties` file in the following properties.

Property	Description
`ldap.connect.timeout`	Number of seconds Integrity Lifecycle Manager server waits when connecting to directory server before deciding it is not responding. Default is `5`.
`ldap.blacklist.timeout`	Minimum number of seconds Integrity Lifecycle Manager server waits before trying to reconnect to inactive directory server. Default is `300`.
`ldap.pool.timeout`	Number of milliseconds active directory server connection remains in pool before its connection removed. Default is `600000`.

Alternate Login Names

If you are using a security scheme with an LDAP authentication domain, you can use alternate login names. The alternate login name feature lets you specify a pair of login names that each user can use to authenticate with. It then associates those two login names with a single user name that Integrity Lifecycle Manager uses to uniquely identify the user. Integrity Lifecycle Manager behaves as though the user had authenticated with that name.

For example, you want to let the user Elizabeth Smith authenticate using either of the following:

- Her employee number (for example, `119208`)
- Her common name (`cn`) in the LDAP database (for example, `esmith`)

However, regardless of how she logs in, you always want Integrity Lifecycle Manager to behave as though she logged in using the common name.

To do this, you can configure the `security.properties` file to use alternate login names.

One use for alternate login names is a security situation where you do not want Integrity Lifecycle Manager to display one of the possible authentication names. For example, you may want Integrity Lifecycle Manager to always identify Elizabeth by her common name because employee numbers are confidential.

Another use is when you need to merge two LDAP databases and both groups authenticate with different user attributes. For example, Elizabeth’s division is merging with Harry Jones’ division. Elizabeth’s division authenticates using employee numbers while Harry’s division authenticates using LDAP common names. In the future, both groups will use common names. To smooth the transition, you can use alternate login names to let users authenticate with either their employee number or their LDAP common name. Thus, Elizabeth’s division can still log in with employee numbers and Harry’s division can still use LDAP common names. However, in both cases, Integrity Lifecycle Manager uses the LDAP common name to uniquely identify each user.

Note

To use alternate login names, you must be running version 10.4 or later of Integrity Lifecycle Manager server, Integrity Lifecycle Manager client, and all Integrity Lifecycle Manager proxies. In addition, you should use version 10.4 or better of the Microsoft Visual Studio and Eclipse integrations.

Configuring LDAP to Use Alternate Login Names

In the LDAP user database, each user entry has various attributes which provide additional data about the user. The schema for the version of the LDAP database on your system defines the names and valid content for those attributes.

In the `security.properties` file, the `ldap.user.filter` property defines how LDAP searches for a user when that user authenticates. It specifies which user attributes are looked at when seeking a match for the login name used to authenticate with (that is, the authentication name). The `ldap.user.name` property defines the user attribute to use as the Integrity Lifecycle Manager user name. For example:

```
ldap.user.filter=(&(cn\=%u)(objectclass\=inetOrgPerson))
ldap.user.name=cn
```

In this example, LDAP searches the database for a user entry in the `inetOrgPerson` object class where the `cn` attribute matches the authentication name (`%u`). When a match is found, the `ldap.user.name` property tells Integrity Lifecycle Manager to use the `cn` attribute as the Integrity Lifecycle Manager user name.

Note

The Integrity Lifecycle Manager user name must be a name that can be used to authenticate with. Thus, the `ldap.user.name` property must be set to a user attribute that is searched for by the `ldap.user.filter` property.

For alternate login names to work, you must edit the `security.properties` file:

- Set `ldap.user.filter` to search for LDAP user attributes that might match the authentication name. You can match against two user attributes. Both attributes must have unique values for each user.
- Set `ldap.user.name` to the user attribute (one of the two searched for by

`ldap.user.filter`) that you want to use as the Integrity Lifecycle Manager user name.

- Set `ldap.user.alternatelogin` to the other LDAP user attribute with which the user can log in.

> **Note**
>
> If you do not set the `ldap.user.alternatelogin` property, the alternate login name feature may appear to work correctly. However, Integrity Lifecycle Manager may create additional sessions, leading to session limit and licensing issues.

So, to let Elizabeth Smith log in using either her employee number or her LDAP common name (`cn`), you would set these properties to something like:

```
ldap.user.filter=(|(&(cn\=%u)(objectclass\=inetOrgPerson))
     (&(employeeNumber\=%u)(objectclass\=inetOrgPerson)))
ldap.user.name=cn
ldap.user.alternatelogin=employeeNumber
```

Using these settings, LDAP searches for a user entry where either the common name (`cn`) or the employee number (`employeeNumber`) matches the name used to authenticate with. When a match is found, the `ldap.user.name` property sets the common name for the matching user entry as the Integrity Lifecycle Manager user name. The `ldap.user.alternatelogin` property sets the employee number as the other attribute that the user can authenticate with.

See your LDAP documentation for more information about the LDAP concepts and syntaxes discussed here.

Alternate Login Names and Proxies

When using the alternate login name feature with Integrity Lifecycle Manager proxies, the behavior of the feature depends on the proxy's configuration.

When connecting with a proxy that does not use LDAP, two issues may result.

- Connection lists in the Integrity Lifecycle Manager client display the actual authentication name.
- Users must disconnect using the actual name they authenticated with.

For proxies which do use LDAP, the `security.properties` file on the proxy should be configured to provide the same alternate login name behavior as the Integrity Lifecycle Manager server.

Note

Exact details on the proxy's configuration depend on the schema for the version of the LDAP database on the proxy.

Alternate Login Names and Production Data

It is always best to set up alternate login names before a server contains any production data.

For servers with production data, you can change the `ldap.user.filter` property to search against a pair of LDAP user attributes, thus allowing users to authenticate as either. However, you should not change the `ldap.user.name` property to use a different user attribute as the Integrity Lifecycle Manager user name. To change the Integrity Lifecycle Manager user name to a different user attribute, contact Integrity Lifecycle Manager Services for assistance in migrating your existing data.

Once you have migrated your data, you must set the `ldap.user.name` property to the user attribute being used as the new Integrity Lifecycle Manager user name. You must migrate all existing data to use the new Integrity Lifecycle Manager user name before users can safely log in using the new `ldap.user.name` value.

When you change the setting for the `ldap.user.name` property without first migrating your existing data, Integrity Lifecycle Manager creates a new user to match the new Integrity Lifecycle Manager use name. As a result, all existing data associated with a user is no longer be consistent with that user's account. You cannot cleanly migrate that data once the user has logged in using the new `ldap.user.name` value.

UNIX Security Realm

The UNIX security realm is supported for Solaris, SuSE Linux, and Red Hat Linux.

Note

When setting up the UNIX security realm, you must specify all the users for primary and secondary groups in the UNIX groups file. You cannot rely on the primary group ID for the user.

To install the UNIX security realm

Note

Administrator must be logged in as root.

1. Move to the following directory:
   ```
   installdir/server/mks/bin
   ```
2. Make sure the correct file owner and permissions are assigned to the `verify` file:

   Note

   If you are administering the UNIX security realm under Solaris or Linux, the `verify` program only needs to have `setuid` applied if shadow passwords are used.

   ```
   chown root verify
   chmod +xs verify
   ```

   - For Solaris, add the following lines to your `/etc/pam.conf` file:
     ```
     # Setup for authentication on Solaris machines
     #
     mksUnixRealm auth requisite pam_authtok_get.so.1
     mksUnixRealm auth required pam_unix_cred.so.1
     mksUnixRealm auth required pam_unix_auth.so.1
     ```
   - For Red Hat Linux, create a file `/etc/pam.d/mksunixrealm` containing the following:
     ```
     #%PAM-1.0
     #
     auth        include     system-auth
     account     include     system-auth
     password    include     system-auth
     session     include     system-auth
     ```

Note

To use the UNIX security realm on Red Hat Linux, you must first ensure the 32-bit versions of some libraries are installed. To do this, use the system tools `rpm` or `yum` to verify, and if necessary, install the `pam.i686` package and related dependencies. Note that `rpm` and `yum` are system tools—they are not included with Integrity Lifecycle Manager.

- For SuSE Linux, create a file `/etc/pam.d/mksunixrealm` containing the following:

  ```
  #%PAM-1.0
  #
  auth        required     pam_unix2.so
  account     required     pam_unix2.so
  password    required     pam_unix2.so
  session     required     pam_unix2.so
  ```

3. Restart the Integrity Lifecycle Manager server and log in using UNIX user IDs.

2

Server Installation

Server Installation

Installing Integrity Lifecycle Manager server

The Integrity Lifecycle Manager DVD in your package includes everything you need to install Integrity Lifecycle Manager. The components of Integrity Lifecycle Manager you can run depend on what you have purchased.

Note

For details on installing your solution template, consult the documentation provided with it.

Integrity Lifecycle Manager includes the Integrity Lifecycle Manager server, which is the common server for the Integrity Lifecycle Manager client.

The Integrity Lifecycle Manager client comprises the Administration Client and the feature sets for workflows and documents, and configuration management.

The procedures in this section describe the steps required to perform a new installation of the Integrity Lifecycle Manager server on Windows and UNIX platforms. For information on the operating systems that Integrity Lifecycle Manager server supports, see "System Requirements" on page 21.

Installing Integrity Lifecycle Manager server From DVD

This section describes how to install the Integrity Lifecycle Manager server from the DVD. The detailed steps of the installation depend on the server installation type you select (database repository or proxy server).

To install Integrity Lifecycle Manager server from a DVD

On Windows and on certain UNIX systems, the Integrity Lifecycle Manager DVD browser starts automatically and allows you to select from the following options:

- **Installation** allows you to install the Integrity Lifecycle Manager server, Integrity Lifecycle Manager client, FlexNet license server, and Integrity Lifecycle Manager Agent.
  - Before you begin the installation process, ensure that you have sufficient privileges to install Integrity Lifecycle Manager server.

    - The silent installation option is also available.
- **Documentation** allows you to view the product documentation in Adobe Acrobat PDF format.
- **Release Notes** allows you to view the Integrity Lifecycle Manager release notes.

Integrity Lifecycle Manager server Installation Process

The following list outlines the installation steps by installation panel in the Integrity Lifecycle Manager server installation wizard. The purpose of this list is to provide key details not included in the full installation procedure or the GUI descriptions. It contains links to supporting information and documentation but is not intended as a complete guide to installation.

1. **License Agreement** panel:
    - Integrity Lifecycle Manager includes the FlexNet network license manager. If you already have a FlexNet license server on your system, you can use it instead to manage your Integrity Lifecycle Manager licenses.
    - For new installations of the Integrity Lifecycle Manager server, you must obtain your license file before installing the Integrity Lifecycle Manager server so that it can be checked during the installation process.

      During the installation process, the license file is copied to `*installdir*/data/license` and will be used from there unless you specify another location in the `mksis.licensePath` in the `is.properties` file.

      Note

      When installing Integrity Lifecycle Manager on Windows, ensure you preserve existing system settings by manually creating a system restore point. Creating a restore point allows for system restoration in the case of error. For information on creating restore points, refer to http://www.microsoft.com and find the restore procedure for the version of Windows you are running.

    If your system language is supported (currently English, German, Japanese, Chinese Simplified, Chinese Traditional, Korean and French are supported), the installer program runs in that language. You can change the installer language when prompted by the **Run installer in** dialog box that displays before the license agreement. Once the install language is set, all installer panels appear in that language. License agreements also appear in the installer language.

For more information, see “License Process” on page 22

2. **Integrity Server Installation Location** panel:

   - The installation will fail if you have not uninstalled the previous version of Integrity Lifecycle Manager server. A panel will indicate that it detects an existing version and either force you to specify a new directory or uninstall the existing version.
   - The Integrity Lifecycle Manager server and/or Integrity Lifecycle Manager Agent installers will follow the expected behavior of Windows applications defaulting to install under `c:/Program Files/Integrity/ILMServer11`. The release is appended to the Integrity Lifecycle Manager server directory location.
   - If you have uninstalled prior to installing a new version or upgrading an existing version, files remaining in the directory be manually deleted in order to provide a clean install and prevent issues.
   - The path location cannot include special characters, such as #.

   For more information, see the “Integrity Lifecycle Manager Server Configuration Properties” topic in the *Integrity Lifecycle Manager Help Center*.

3. **Select Product Language** panel:

   The **Use Integrity LM in** dropdown will be unavailable, but the default value will be the default system language. Click **Install** to continue.

4. **Integrity Lifecycle Manager Server Notification** panel:

   Note

   This panel only displays for new server installations. If you are upgrading an existing Integrity Lifecycle Manager server, it will not display.

   - **Mail Server** is the name of the mail server.
   - **Mail Server Port** is the port number of the mail server.
   - **Email To** is a single e-mail address or a comma-delimited list of e-mail addresses to deliver server e-mail notifications to.

5. **Integrity Lifecycle Manager Server Install Type** panel:

- **New server**
- **Upgrade of an existing server**: Refer to the applicable Integrity Lifecycle Manager upgrading documentation on the PTC Integrity eSupport portal.

6. **Integrity Lifecycle Manager Server License File Location** panel:

   `License.dat` is provided to you or available in the PTC Integrity eSupport portal. Copy it to your Windows user directory before beginning the installation process.

7. **Integrity Lifecycle Manager Server Type** panel:

   - **Full server** (includes proxy capabilities).
   - **Proxy only server**. Proxy servers are a part of the Federated Server Architecture. Installation and configuration are fully described in the "Configuring FSA" topic in the *Integrity Lifecycle Manager Help Center*.

8. **Integrity Lifecycle Manager Full Server Capabilities** panel:

   Select **Workflows and Documents**, **Configuration Management**, or **Both**, as applicable to your environment.

   The options are provided to support clustered configurations where only Workflows and Documents are supported.

   For more information, see the "Database Creation Options" topic in the *Integrity Lifecycle Manager Help Center*.

   Note

   Detailed descriptions of all database options and properties are located in `*installdir*/config/install/mksserver.properties`.

9. **Integrity Lifecycle Manager Server Database** panel:

   The database options that display on this panel depend on the options selected in steps 8 and 9.

   For the most current product platform support information, see http://support.ptc.com/partners/hardware/current/support.htm.

   Note

   The **Embedded** (proxy-only server) option is used exclusively for proxying other servers; it does not use a repository. The proxy-only server uses the Derby embedded database to track cache-related operations.

For more information, see the “Integrity Lifecycle Manager Server Configuration Properties” topic in the *Integrity Lifecycle Manager Help Center*.

10. **Migrate Existing Data** panel:

    Choose to migrate existing data. Depending on the amount of data you are migrating it may take several hours to complete this step.

    If you choose **No**, the install is aborted.

11. Restart your computer to have the system environment changes take effect.
12. Configure the Integrity Lifecycle Manager server. For more information, see the “Integrity Lifecycle Manager Server Configuration Properties” topic in the *Integrity Lifecycle Manager Help Center*.

Installing Integrity Lifecycle Manager Server Using Silent Install

If you have a number of Integrity Lifecycle Manager servers to install, using a silent install saves time by enabling you to configure the server installation options once, and then run the install on multiple servers without the need for any further interaction.

If you are installing on a UNIX platform, using a silent install means that you do not need to set the environment variable `$DISPLAY`.

There are two components to installing the server using a silent install:

- configuring the `mksserver.properties` file
- running the silent install command on the Integrity Lifecycle Manager servers

Note

Errors that occur during the silent install are not displayed in the GUI or text output. To verify that the silent install was successful, check the return code by using the `echo $?` command from a command line. Non-zero exit code indicates the silent install failed.

Configuring Server Properties File

The `mksserver.properties` file is located in the *`installdir`*`/config/install` directory. Silent installer property files are also available on the DVD in the `support/install-properties` folder. Review the following properties and update as required.

Note

Since the silent install is intended to be used without any user intervention, it is important that all information provided in the `mksserver.properties` file be valid.

Property	Description
`INSTALLER_UI`	Specifies what mode to use for installation. By default, property is set to `silent`, which installs server based on values in this properties file. To run GUI installation and display prompts for server installation parameters, set property to `gui`.
`MKS_LICENSE_AGREEMENT`	Specifies whether you accept terms of license agreement. A copy of license agreement can be found on DVD in `support/install-properties` folder. By default, property is set to `false`. If not set to `true`, installation is canceled.
`MKS_ORGANIZATION`	Name of your company or organization.
`USER_INSTALL_DIR`	Where to install Integrity Lifecycle Manager server. If server is running Windows, use double backslashes for paths in properties. Spaces in path should be preceded by backslash. For example, the default install location on Windows is `C:\\Program\ Files\\Integrity\\ILMServer11`.
`MKS_UPGRADE`	Specifies whether the installation is an upgrade of an existing server.
`MKS_EXISTING_INSTALL_DIR`	If `MKS_UPGRADE` is set to `true`, specifies the path of the existing server installation to upgrade.
`MKS_LICENSE_FILE`	Specifies path to directory for `license.dat` file. You can specify actual license file in path or just specify path.

Property	Description
MKS_DATABASE_TYPE	Type of database to use with Integrity Lifecycle Manager server. Valid values are: • MSSQL (Microsoft SQL Server) • ORACLE • EMBEDDED (Proxy server only) By default, property is set to EMBEDDED.
MKS_COPY_OLD_EMBEDDED_DATA	If your previous server installation and the new one are using the embedded database and you want to copy over the data from the previous installation, set MKS_COPY_OLD_EMBEDDED_DATA=true.
MKS_OLD_INSTALL_LOCATION	If MKS_COPY_OLD_EMBEDDED_DATA=true, set MKS_OLD_INSTALL_LOCATION to the path where the old server is installed, for example, MKS_OLD_INSTALL_LOCATION=C:\Program Files\MKS\IntegrityServer.
MKS_DATABASE_HOST_NAME	Database server host name. If MKS_DATABASE_TYPE=EMBEDDED, property not required.
MKS_DATABASE_PORT	Database server port. If MKS_DATABASE_TYPE=EMBEDDED, property not required.
MKS_DATABASE_NAME	Database name for MS SQL database or service name for Oracle database. If MKS_DATABASE_TYPE=EMBEDDED, property not required.
MKS_DATABASE_LOGIN_NAME	User name of authorized database user (with permission to create and modify privileges). If MKS_DATABASE_TYPE=EMBEDDED, property not required.
MKS_DATABASE_LOGIN_	Password of authorized database user

Property	Description
PASSWORD	(with permission to create and modify privileges). Password must be in plain text. Does not get encrypted. If MKS_DATABASE_TYPE=EMBEDDED, property not required.
MKS_INSTANCE_NAME	Optionally, sets the database instance name for an MS SQL database.
MKS_MIGRATE_EXISTING_DATA	Specifies whether existing Integrity Lifecycle Manager server data detected during installation should be migrated. By default, property set to true. If set to false, installation canceled.
MKS_CASE_INSENSITIVE	If a new installation, specify whether database is case sensitive (false) or case insensitive (true). By default, property to true.
MKS_NOTIFY_MAIL_SERVER	Specifies the mail server name used for sending server e-mail notifications.
MKS_NOTIFY_MAIL_SERVER_PORT	Specifies the mail server port used for sending server e-mail notifications.
MKS_NOTIFY_EMAIL_TO	Specifies a single e-mail address or comma-delimited list of e-mail addresses to deliver server e-mail notifications to.
MKS_PRODUCT_LANGUAGE	Specifies the language of the Integrity user interface. You can display the user interface in one of the supported languages (currently, English and Japanese are supported). When using Integrity on a non-English or non-Japanese system, it is possible to display the user interface in English and retain locale-specific formatting for time, number, and currency values. Set this property to a valid two-letter ISO-639 code to specify how to display the user interface. Options are as follows:

Property	Description
	• If the ISO-639 code corresponds to a supported language, then both user interface and formatting display in that language. • If the ISO-639 code corresponds to an unsupported language, then user interface displays in English and formatting displays according to the ISO–639 code specified. • If the property is left blank or set to an invalid ISO-639 code, then user interface and formatting display in the system language, if supported. If not supported, then user interface displays in English and formatting displays according to the system language. For example: ○ If `MKS_PRODUCT_LANGUAGE=ja`, then both user interface and formatting display in Japanese because it is a supported language. ○ If `MKS_PRODUCT_LANGUAGE=de`, then user interface displays in English because German is not a supported language, but formatting displays according to German rules. ○ If `MKS_PRODUCT_LANGUAGE=`*`invalid_code`*, then user interface displays in system language, if supported. If not supported, then user interface displays in English. In either case, formatting displays according to the system language.

Property	Description
MKS_PROXY_SERVER	Installs the proxy server that works for both Workflows and Documents and Configuration Management. After installation, a full server can also run as a proxy server. For more information, see the “Global Development Using a Proxy” topic in the *Integrity Lifecycle Manager Help Center*. By default, this property is set to false, and the full server is installed. To install the server as a proxy server, set the property to true.
MKS_WORKFLOWS_AND_ DOCUMENTS_ENABLED	Specify the Workflows and Documents capabilities of the Integrity Lifecycle Manager server. By default, this property is set to true. To disable the Workflows and Documentation capabilities, set the property to false.
MKS_CONFIGURATION_ MANAGEMENT_ENABLED	Specify the Configuration Management capabilities of the Integrity Lifecycle Manager server. By default, the value is set to true. If set to false, the Configuration Management capability is disabled.

Running Silent Install

1. Copy any files that are referred to in the `mksserver.properties` file to the server before installation (for example, the license file).
2. Copy `mksserver.properties` to a temporary location on the server.
3. Install the Integrity Lifecycle Manager server by doing one of the following:
   - If the location of `mksserver.properties` is not in the same directory as the `mksserver.exe`, run:

     ```
     mksserver -f mksserver.properties
     ```

     or

     ```
     ./mksserver -f mksserver.properties
     ```

where -f flags the location of mksserver.properties.

- If the location of mksserver.properties is in the same directory as mksserver.exe, run either mksserver.exe or mksserver.bin as appropriate for your platform.

Running Integrity Lifecycle Manager server

Before you start the Integrity Lifecycle Manager server, note the following:

- You must have FlexNet licensing set up properly and running before the Integrity Lifecycle Manager server can start. For more information on using FlexNet licensing, see "License Process" on page 22.

When the server is started for the first time, Integrity Server started is written to server.log and startup.log. If the server is restarted after a temporary locked state, Integrity Server transitioned to Operating state is written to server.log and startup.log.

Note

By default, a maximum of 50 server.log and startup.log files are retained on the Integrity Lifecycle Manager server. Each log file has a maximum 10 MB default size limit. Versions are named incrementally, for example, server.1.log, server.2.log, server.n.log. To configure the size of the log file and the number of logs, see the "Integrity Lifecycle Manager Server Configuration Properties" topic in the *Integrity Lifecycle Manager Help Center*.

To run the Integrity Lifecycle Manager server from the CLI

To start the Integrity Lifecycle Manager server, use the mksis.bat (mksis in UNIX) file with the start subcommand. The file also provides additional administrative functionality for running the Integrity Lifecycle Manager server.

The file is located in the following directory:

```
installdir/bin/mksis[.bat]
```

where *installdir* is the Integrity Lifecycle Manager server installation directory.

The following is the syntax for the command:

```
mksis <subcommand>
```

where <*subcommand*> is one of the following:

- console starts the Integrity Lifecycle Manager server in console mode as an application only without the service, as specified in *installdir*/config/mksservice.conf

- `install` installs the Integrity Lifecycle Manager server as a Windows NT service.
- `remove` removes the Integrity Lifecycle Manager server server service.
- `restart` restarts the Integrity Lifecycle Manager server service.
- `start` launches the Integrity Lifecycle Manager server service (Windows) or daemon (UNIX).
- `stop` stops the Integrity Lifecycle Manager server service (Windows) or daemon (UNIX).

To set up the Integrity Lifecycle Manager server as a UNIX service

Note

The UNIX system administrator should complete the following steps.

1. Test the service by connecting to the appropriate port.
2. Check the contents of the startup scripts for errors and system compatibility. The startup scripts are located in the following directory:

   `/etc/init.d`
3. Update the appropriate directories with the `rc.d` scripts.

   Note

   For Linux and Solaris platforms, runlevel specifics go in `/etc/rc.d/rcx.d` where `x` is `1`, `2`, `3`, `4`, `5`, `6`. These are symbolic links to the scripts in `/etc/init.d`. You can use the program `chkconfig` to update the `/etc/rc.d/rcx.d` directories.

   See article CS208955 on the PTC Integrity eSupport portal for details about configuring UNIX/Linux to start Integrity Lifecycle Manager server on system startup.
4. To launch the daemon, run `rc.mksis` or `init.d/mksis` start.

Installing Integrity Clients

For information on installing and configuring Integrity Clients, see the *Integrity Lifecycle Manager Help Center*.

3

Server Configuration

Common Server Operations

Configuring Context Based Text Searching

The text searching feature enables the use of context search indexes. Context search indexes are automatically built for text searches, if the underlying database supports it; however, the text search queries perform differently using a context search.

The **Search** function allows users to carry out simple text searches of the Integrity Lifecycle Manager item database. The text search uses a search syntax similar to many common Web search engines. The search is carried out by the underlying database and the results are passed back to Integrity Lifecycle Manager.

Text searches are available to users through both the GUI and the Web interface using the **Search** field on the toolbar. Text searches look only for information that is in short or long text fields. The search does not capture information in other types of fields, such as integer, pick, floating point, logical, date, user, or group fields. For more information on the text search feature, see the *Integrity Lifecycle Manager Help Center*.

The property for configuring context based text searches is `mksis.im.contextSearchOn` set through **Workflows and Documents ▸ Configuration ▸ Properties** in the Integrity Lifecycle Manager administration client. By default, `mksis.im.contextSearchOn` is set to `true`.

Note

For information on the databases that support the text search feature, see: http://support.ptc.com/partners/hardware/current/support.htm. If you are migrating an Integrity Lifecycle Manager database from a previous release, certain additional steps are required. For more information, see the Integrity Lifecycle Manager upgrading documentation on the PTC Integrity eSupport portal.

If your users have Integrity Lifecycle Manager queries from a previous release that rely on the old text searching behavior, you may want to disable the use of the context search indexes temporarily. Otherwise, you can leave context searching enabled if the underlying database supports it.

Configuring an Oracle Database for Special Character Searches

If the Integrity Server is running on an Oracle database, you can configure text searches to include certain special characters, such as the percent (`%`) and underscore (_) symbols, using the following Integrity Server property:

```
mksis.im.additionalTokenCharacters
```

In Oracle, you must also configure a custom lexer to have the special characters included as part of a search term. The database lexer can be configured using Oracle DDL when creating user-defined text tables. For more information on configuring user-defined text tables, see the *Integrity Performance Tuning Guide*.

Note

You can also change the index on an existing Integrity indexed text tables (for example, `Text0`, `Text1`, `Text3`); however, you may need to recreate that index if you later decide to upgrade the Integrity Server.

The following example provides a basic lexer definition that supports the `%` and _ symbols:

```
begin
ctx_ddl.create_preference('INTEGRITY_LEXER', 'BASIC_LEXER');
ctx_ddl.set_attribute('INTEGRITY_LEXER', 'PRINTJOINS', '_%');
end;
```

After configuring the lexer, you must create (or re-create) the appropriate text index.

For text tables that represent rich content (HTML) (for example, `Text0`):

```
drop index text0_value;
create index text0_value
on text0(value)
indextype is ctxsys.context
filter by IssueID,TypeID
parameters('filter ctxsys.null_filter section group
    ctxsys.html_section_group lexer INTEGRITY_LEXER');
begin
ctx_ddl.sync_index('text0_value');
end;
```

For text tables that represent plain text (for example, `Text1`):

```
drop index text1_value;
create index text1_value
on text1(value)
indextype is ctxsys.context
filter by IssueID,TypeID
parameters('lexer INTEGRITY_LEXER');
begin
```

```
    ctx_ddl.sync_index('text1_value');
    end;
```

After (re)creating the indices, you configure the Integrity Server to include the additional characters by modifying the`mksis.im.additionalTokenCharacters` property. The property is configured through the Integrity Administration Client under the **Workflows and Documents ▸ Configuration ▸ Properties** node.

The default value of this property includes the underscore (_) and period (`.`) characters. These characters are ignored by Oracle's default lexer.

Modify the property as follows:

```
mksis.im.additionalTokenCharacters=_%
```

The property is dynamically updated on the Integrity Server. A server restart is not required to have the changes take effect.

After completing the changes on the server, new searches for terms such as `auto_system` and `75%` will return items containing these terms. The functionality is not limited to the underscore and percent symbols. Provided the database lexer supports it, other special characters can be included in text searches. For more information, refer to your Oracle product documentation.

Note

When specified in `mksis.im.additionalTokenCharacters`, certain special characters are ignored because they are part of the documented query language. The ignored characters are: asterisk `*`, pipe `|`, double-quote `"`, plus sign `+`, minus sign `-`, and tilde `~`. For the purposes of searching, the asterisk symbol (`*`) continues to represent a wildcard (zero or more characters).

Synchronizing a Database

If your database does not support the automatic tracking of text field updates in the context search indexes, you can set a property to specify an interval for synchronizing that database. The property is configured through **Workflows and Documents ▸ Configuration ▸ Properties** in the Integrity Lifecycle Manager administration client as follows:

```
mksis.im.contextSearchSync=30
```

Note

This property is ignored for databases that do not require it. Currently only the Oracle database uses it.

Setting the `mksis.im.contextSearchSync` property forces an update within the time interval (in seconds) that you specify. By default, this value is set to 30 seconds—that is, a synchronization every 30 seconds. The minimum value is 10 seconds.

Configuring Attachment Size Limits

As administrator, you can control the maximum size of file that users can attach to any item in Integrity Lifecycle Manager using the Integrity Lifecycle Manager administration client. Select **Workflows and Documents** ▸ **Configuration** ▸ **Properties**. You can set a higher or lower limit by modifying the value for `mksis.im.maxAttachmentSize`.

For example, the following setting limits the maximum size of an attached file to 4 MB:

```
mksis.im.maxAttachmentSize=4
```

The minimum value for the attachment size is a limit of 1 MB, while the maximum value is a limit of 250 MB. By default, the limit for attachment file size is 4 MB.

Note

For Microsoft SQL Server, the attachment size should not exceed 25 MB.

For more information on configuring properties, see the Properties topics.

Configuring Limits for Queries

A query is a request to select and list the items that meet specific selection criteria. The selection criteria are a logical expression of specific values, or ranges of values, of the standard and custom fields of the item type.

Integrity Lifecycle Manager users have the ability to create complex queries that can demand significant system resources when run against a large database of items. Queries that return a large result count can also demand significant memory resources from both the Integrity Lifecycle Manager server and client.

To address the item of complex queries, Integrity Lifecycle Manager includes the following options:

- system query absolute time-out property (`mksis.im.queryTimeOut`)
- system query default time-out property (`mksis.im.queryTimeOutDefault`)
- query time-out setting for groups configured through the Groups view
- maximum query item count property (`mksis.im.queryGovernor`)

If a user runs a query exceeding the timeout limits, Integrity Lifecycle Manager displays an error message.

If a user runs a query exceeding the item count limit, Integrity Lifecycle Manager displays a warning message along with a partial set of results (that is, all items up to the item count limit). This behavior does not apply to queries that are not directly run by a run (for example, a query behind a report). In such cases, Integrity Lifecycle Manager displays an error message.

Note

For versions of Integrity before 10.6, Integrity Lifecycle Manager still displays an error message when a user's query exceeds the item count limit.

As administrator, you can also stop a query that is actively running on the Integrity Lifecycle Manager server. For more information on viewing and stopping query processes, contact PTC Technical Support.

All query time-out properties are set through the Workflow and Document configuration properties.

How Query Time-out Settings Work

The query time-out properties are used in the following order:

- If the user is a member of a group with a time-out setting, that time-out is used. If the user is a member in more than one group, the highest group time-out setting is used.
- If there is no group time-out setting, or if the group time-out setting is 0, the system default time-out is used.
- If there is no system default time-out, or if the default time-out setting is 0, the system absolute time-out is used.

Note

A system time-out value of 0 means there is no limit on the time allowed for the query and queries are allowed to run until all results are returned.

Improving Queries to Avoid Increasing Query time-outs

When a query uses a database index, it usually completes in less than a second. When a query does not use a database index, it may need to scan a very large amount of data in the database. It is not possible to have appropriate database indexes for all queries that a user might create, but several fields have the capabilities for indexing during field creation. If a particular query is hitting the time-out limit, instead of increasing the time-out value, consider the following ways to improve the query performance:

- Add more filters to the query, for example, filtering for a specific type or state. Users might not add this type of filter if another field enables them to limit their results as required. Adding additional filters makes queries run much faster. For more information on adding query filters, see the *Integrity Lifecycle Manager Help Center*.
- You can add indexes on fields with **Data Type** as **Long Text** or **Short Text**. The fields with **Data Type** as **Long Text** or **Short Text** have the option to create a text search index while creating or editing the fields. The checkbox is available on **Values** tab.

  You can also create indexes on user and group fields. User and group fields are frequent search targets. If a user or group field is not indexed, queries on them are satisfied only by scanning other field indexes which is a very slow process. You should create an index on user and group fields unless you are certain the user or group field is not used very often.
- For a frequently run query that is shared to a group, consider adding an index to make that specific query run better.
- For a query that contains fields that are frequently used in other queries, consider indexing those fields.

Configuration Information

Configuration information must reside on the Integrity Lifecycle Manager server. The proceeding table lists the path and file names, and a brief description for each of the policy and property configuration files included in Integrity Lifecycle Manager. Additional properties are stored in the database and configured from the Integrity Lifecycle Manager administration client or the CLI. For more information, see the “Using Properties” topic in the *Integrity Lifecycle Manager Help Center*.

Note

References to *installdir* represent the path to the directory where you installed the Integrity Lifecycle Manager server.

Path and File Name	Description
installdir/config/properties/im.properties	Configures workflows and documents feature set.
installdir/config/client/IntegrityClientSite.rc	The IntegrityClientSite.rc file on the client side is used to configure a Integrity Lifecycle Manager client installation with default preferences that can be picked up by all Integrity Lifecycle Manager client users. This file is used to define logging, diff tools, merge tools, and SI bulk data cache sizes. **Note** Generally, multiple users of the same Integrity Lifecycle Manager client apply to Unix setups but can occur on Windows in the case of Citrix.
installdir/config/properties/is.properties	Configures Integrity Lifecycle Manager server, including what server components you want to execute, and absolute path to FlexNet license file where you set seat license groups.
installdir/config/properties/logger.properties	Configures logging for Integrity Lifecycle Manager server. Includes properties that configure logging for SQL operations, SMTP operations, e-mail notification, ACL resolution, configuration management transactions for cache and users, and LDAP-compliant security realms. For more information on security, see the Administrator documentation.
installdir/config/policies/servicepack.policy	Specifies minimum version of client that is allowed to connect to Integrity Lifecycle Manager server. Default behavior is to follow the published client-server compatibilities. For more information on client-server compatibilities, see the Integrity Lifecycle Manager upgrading

Path and File Name	Description
	documentation on the PTC Integrity eSupport portal. **Note** `minClientBase` property in `servicepack.policy` file specifies minimum version of Integrity Lifecycle Manager client allowed to connect to Integrity Lifecycle Manager server. Property only works with Integrity 4.6 and greater. For more information on applying and distributing service packs, see "Installing Integrity Server" on page 50.
`installdir``/config/properties/security.properties`	Configures Integrity Lifecycle Manager server authentication mechanism. For more information, see "Configuration Information" on page 69.
`installdir``/config/properties/si.properties`	Configures configuration management feature set.
`installdir``/data/triggers/env.properties`	Defines environment settings used in JavaScript triggers. For more information, see the triggers information in the Administrator documentation.

Additional Configuration Options

Integrity Lifecycle Manager also supports personal configuration information in the form of preferences set on each client. For details on preferences, see the User documentation.

Project files can also hold configuration options. These settings persist while the project is opened using the GUI, Web, or CLI. Configuration options in project files are sometimes called attributes. Within project files, you can set your own attributes as well as those predefined as options. You can set these attributes from the graphical user, Web, or command line interfaces.

Environment Variables

Integrity Lifecycle Manager recognizes the environment variables listed in this section. The environment variables perform similarly to the same-named configuration options within Integrity Lifecycle Manager.

If both the configuration option and environment variable are defined, the environment variable takes precedence. When setting these variables in an initialization file, the variable names must be uppercase. Integrity Lifecycle Manager does not recognize them otherwise.

Variable	Value
`PATH`	Standard search path.
`TMPDIR`	By default, Integrity Lifecycle Manager commands store temporary files under `%TMPDIR%` on Win32 and under `$TMPDIR` on UNIX. To use different directory, set `TMPDIR` to name of directory you want to use.

Setting Up Integrity Help

Integrity product help is accessed through the Integrity Server. To retrieve help on the server, the server's default connection must be defined in the Integrity Client's **Default Server Connection and Online Help** preference, and the server must be available to display the requested help topic in a Web browser.

In some cases, Integrity help may be accessed through a proxy server or a non-Integrity Server:

- If you have a proxy server defined, then help is retrieved from the proxy.
- If you have an older Integrity Client and you request help from a newer Integrity Server, then help requests may be redirected to a non-Integrity Server. In this case, help for the older client version may be retrieved from another server or from the PTC Web site; this may require additional logon information. Your administrator defines the server configuration on the Integrity Server. The client configuration is defined to the Integrity Server, even if requests are subsequently redirected to a different server.

To assist with maintaining a valid configuration, the help server is validated when you start the client and when you change the client’s preferences. If the server was not reached successfully, a subsequent help request displays a warning message to notify you of the problem; either the specified server is invalid or currently unavailable, or the default server connection is not defined.

- If the server is not defined, you can click **Help** in the warning message to access this page locally. Follow the instructions described later in this document to set up a default server connection.
- If the server is invalid or unavailable and it may be available now, you can click **OK** to ignore the warning message. If the server is available, the requested help displays in a Web browser. If the server is not available, a Web browser error displays.

  Clicking **OK** to ignore the warning message disables it from redisplaying again until the server is revalidated; either during a client restart or after clicking **OK** in the **Preferences Configuration**window.

You can display this page at any time in the Integrity Client or Administration Client by selecting **Help** ▸ **Integrity Help Setup**, or by clicking **Help** in the warning message that displays when help on the server is unreachable.

To set up a default server connection for the Integrity Lifecycle Manager server and online help

1. Do one of the following to open the **Preferences Configuration** dialog box:

   - In the Integrity Lifecycle Manager client GUI, select **File** ▸ **Preferences Configuration**. The **Preferences Configuration** dialog box displays.

     Note

     If this is your first time starting the client or there are no open ViewSets, a selection dialog box displays. Click **Preferences**.

   - In the Administration Client, select **Tools** ▸ **Preferences**. The **Preferences Configuration** dialog box displays.

   CLI EQUIVALENT

   ```
   im viewprefs -g
   ```

2. In the tree pane, select **Integrity Client** ▸ **Connection**. The **Connection** pane displays.

3. Under **Default Server Connection and Online Help**, specify the following options:
   - In the **Host Name** field, type the name (for example, `xyz-server`) or numerical IP address (for example, `1.2.34.56`) of the defaultIntegrity Lifecycle Manager server.
   - In the **Port** field, type the port number of the default Integrity Lifecycle Manager server, for example, `7001`.

   Note

   If a proxy server is defined for the default Integrity Lifecycle Manager server, then help is retrieved from the server defined in your client's **Proxies** preference. Do not define a proxy in the **Default Server Connection and Online Help** fields.

4. To save your changes, click **OK**.

   The host name and port are tested to ensure the connection is valid and the server is available (this may take a few minutes to complete). Once the validation completes successfully, help is accessible on the server.

Integrity Lifecycle Manager Help Backward Compatibility

As of Integrity 10.3 and later, Integrity clients access help that is hosted on the Integrity server. Integrity clients can only access the version of help that matches their release version. Although Integrity Lifecycle Manager supports older clients connecting to new servers, by default only the newest version of the help is hosted on the Integrity Lifecycle Manager server. For example, by default, Integrity 10.4 does not provide help to Integrity client 10.3.

Note

Integrity at version 10.2 and earlier contain their own help and are not impacted by the compatibility issues described in this topic.

If your implementation of Integrity Lifecycle Manager includes older clients connecting to a newer Integrity Lifecycle Manager server, consider one of the following configurations to resolve Integrity Lifecycle Manager Help compatibility issues:

- **Maintain default configuration and provide users with Internet access.**

By default, when an Integrity Lifecycle Manager client requests a version of help that is not available, the Integrity Lifecycle Manager server redirects the Integrity Lifecycle Manager client to the Integrity Lifecycle Manager Help hosted on the PTC Integrity eSupport portal. Your users must have an Internet connection and an account with credentials that can authenticate on the PTC Integrity eSupport portal.

- **Upgrade all Integrity Lifecycle Manager clients to match the version of the Integrity Lifecycle Manager server.**

  Configure the Integrity Lifecycle Manager server to require all clients connecting to it to upgrade (by setting the minimum permitted version of the client). For more information, see the documentation for service pack policies. This configuration avoids the default help redirect by requiring that all users use the matching version of the Integrity Lifecycle Manager client.

- **Upgrade multiple Integrity Lifecycle Manager servers at different times.**

  If your implementation of Integrity Lifecycle Manager includes multiple Integrity Lifecycle Manager servers, plan to upgrade them at different times. If at least one of the servers contains the older version of the Integrity Lifecycle Manager Help, you can modify the newer servers to redirect the help requests of older clients. When all of the servers are upgraded, the redirect can be reset to the default PTC Integrity eSupport portal URL `http://support.ptc.com/cs/help/integrity_hc/integrity110_hc/`. For example, an Integrity 10.4 hosting Integrity 10.4 help can be configured to redirect Integrity 10.3 help requests to another server that is still at version 10.3. To redirect help requests, see the documentation for the `mksis.helpSystemRedirectURL` property. Note that this configuration option only applies to Integrity servers that are in active use for production: Integrity servers cannot be configured to only host Integrity Help.

- **Host the Integrity Lifecycle Manager Help on its own server without Integrity Lifecycle Manager.**

  The Integrity Lifecycle Manager Help is available as a compiled `.war` file that can be hosted on a J2EE compliant Web server (such as Tomcat or JBoss, but not Apache). This means if you have an available Web server running within your organization you can deploy the `.war` to that Web server. Then you can redirect your Integrity Lifecycle Manager server to the URL for the Web server. To redirect help requests, see the documentation for the `mksis.helpSystemRedirectURL` property.

⚠ **Caution**

The version of the `.war` file running on your Integrity Lifecycle Manager server will only function on Integrity Lifecycle Manager's Web server. To obtain a version of the `.war` file that will run on other Web servers, go to the Support section of the PTC Web site (http://www.ptc.com). The `.war` file is available from the documentation area of the software download section that corresponds to the desired Integrity Lifecycle Manager release. The file name will include `Integrity-Help-On-Third-Party-Server`. After downloading the file, rename it to the form `integrity`*`version`*`_hc.war`, where *version* is the release version without punctuation. For example, Integrity 10.3 is `integrity103_hc.war`.

- **Host multiple versions of the Integrity Lifecycle Manager Help on the same Integrity Lifecycle Manager server.**

The Integrity Lifecycle Manager server can host multiple versions of the Integrity Lifecycle Manager Help, and serve the version of the help that matches the version of the Integrity Lifecycle Manager client connecting. For example, both the 10.3 and 10.4 versions of the Integrity Help can be hosted on Integrity 10.4. Additional Integrity Help `.war` files must be copied to the following location:
`InstallDir/server/mks/deploy/`

Note

An Integrity Lifecycle Manager server restart is required for copied `.war` files to take effect.

To obtain the correct version of the Integrity Lifecycle Manager Help `.war` file, go to the Support section of the PTC Web site (http://www.ptc.com). The `.war` file is available from the documentation area of the software download section that corresponds to the desired Integrity Lifecycle Manager release. The file name will include `Integrity-Help-On-Integrity-Server`. After downloading the file, rename it to the form `integrity`*`version`*`_hc.war`, where *version* is the release version without punctuation. For example, the file name for Integrity 10.3 is `integrity103_hc.war`

Alternatively, the `.war` file can be located in installation directory of the older version of the Integrity Lifecycle Manager server. For example, Integrity server 10.3 version of the `.war` file can be located here:

```
InstallDir/server/mks/deploy/integrity103_hc.war
```

Note

Applying an Integrity Lifecycle Manager service pack deletes any `.war` files currently on the server that are from the same major release series as the Integrity Lifecycle Manager server installation. For example, installing an 11.2 release version service pack would delete the 11.0 and 11.1 release version `.war` files, but not the 10.7 release version `.war` file. To retain the `.war` files in such a scenario, back up the `.war` files to another location and then copy them back into the appropriate server directory after the service pack is applied to the Integrity Lifecycle Manager server installation.

Caution

Each additional `.war` file requires 15 MB heap memory and 12 MB PermGen memory. If the Integrity Lifecycle Manager server cannot re-start after copying the war file due to out-of-memory errors, consider another configuration described in this topic. The copied `.war` file will consume heap and PermGen memory from the memory already allocated for Integrity Lifecycle Manager server, potentially causing the server to reach the memory limits or run out of memory. There is higher potential for these errors to occur in 32-bit systems when hosting multiple `.war` files.

Server Troubleshooting

Troubleshooting Kerberos and Kerberos Single Sign-On

The following error messages may appear in log files or debugging information if the Kerberos or Kerberos Single Sign-On has not been set up correctly.

Note

To turn debugging on, add `mks.security.debug=true` to `security.properties`.

- `ERROR(0): No valid credentials provided (Mechanism level: Server not found in Kerberos database (7))`

  If this error message appears in the client side log file, your `mks.security.clientServiceName` setting is not correct. Make sure it is set to be the name of the user the Integrity Lifecycle Manager server is running as.

- `WARNING(0): The registry key required to support Kerberos Single-Sign-On is missing. You may wish to add them manually.`

  `WARNING(0): Integrity Server does not allow registry key to be automatically added. Either add the key manually or consult your Integrity Server administrator`

  `WARNING(0): On Windows Server 2000 and 2003 the key "allowtgtsessionkey" in HKEY_LOCAL_MACHINESystem\ CurrentControlSet\ Control\Lsa\Kerberos\Parameters with value 1 should be added (reboot may be required).`

  If any of these error messages appear in the client side log file, the client side registry key is missing.

- `ERROR(0): No valid credentials provided (Mechanism level: Failed to find any Kerberos Ticket)`

  If this error appears in the client side log file, either the `mks.security.kerberosRealmName` or `mks.security.kdcAddress` setting is wrong, for example, the realm name is not entered in uppercase.

- `15:52:35,661 INFO [IntegrityServer] DEBUG(10): Login exception encountered while attempting authentication of user kmorton via policy default-policy. Details of exception No valid credentials provided (Mechanism level: Failed to find any Kerberos Key)`

  If this error message appears in the server side log file, the `mks.security.SPN` setting does not match the `-princ` option given in the `ktpass` command.

- `17:37:03,208 INFO [STDOUT] error Message is Client not found in Kerberos database`

If this error message appears in the Kerberos debug information, there is a problem with the keytab file. It may contain an invalid principal name, or the `mks.security.SPN` setting may not match the `-princ` option given in the `ktpass` command.

- `DEBUG(10): Login exception encountered while attempting authentication of user ldaprealmtest1 via policy default-policy. Details of exception Pre-authentication information was invalid (24)`

  If this error message appears in the server log when trying to authenticate using either a `windows_clear` or `windows_private` security policy, the case is wrong in the `mks.security.kerberosRealmName` or `mks.security.kdcAddress` setting in `security.properties`.

- `DEBUG(10): Login exception encountered while attempting authentication of user ldaprealmtest1 via policy default-policy. Details of exception Clock skew too great (37)`

  If this error appears in the server log when trying to authenticate using either a `windows_clear` or `windows_private` security policy, the clock on the server is not synchronized with the clock on the client machines. For the Kerberos authentication domain to work, the server and client clocks must be synchronized (within a reasonable amount of time).

For additional troubleshooting information, visit the following Web page:

https://docs.oracle.com/javase/8/docs/technotes/guides/security/jgss/tutorials/Troubleshooting.html

Troubleshooting Database Repository

Starting Integrity Lifecycle Manager server

To start the Integrity Lifecycle Manager server, follow the procedures outlined in the Installation documentation.

Note

If the Integrity Lifecycle Manager server and database repository are on separate machines, and both machines are running with JVM, the machines must be set to the same locale or the Integrity Lifecycle Manager server may not start.

Database Connections After Routine Backups

When performing a routine backup of your database that requires the database to be disconnected, you should wait until the database is fully online before attempting to run any commands from the Integrity Lifecycle Manager server. If the database is not fully online before running a command, an error message is posted.

Archives for Dropped and Added Migrated Members

When using the database repository option for configuration management, dropping a migrated project member and re-adding the member of the same name does not cause the member archive to branch. Instead, a new archive is created without the RCS directory. If you want to reuse the archive, add the member using the **Add From Archive** command (see the User documentation).

Controlling Bulk Data Storage in Oracle

Where Oracle is used for the database repository, you can control the storage of bulk data by controlling the storage of the `VALUE` column in the `CMARCHBULK` table. Oracle Locator Object (LOBs) data is now supported for the database repository.

Adding Large Files on MS SQL Server Database

Adding large files (for example, files larger than 250 MB) is not supported on the MS SQL Server database and can cause a connection timeout to the database.

MS SQL Server Database Uses Case-sensitive Collation

If you are using MS SQL Server as the backing database for the repository, you must configure a case-sensitive collation when creating the database or the Integrity Lifecycle Manager server service will not start. A collation specifies the bit patterns that represent each character and the rules by which characters are sorted and compared.

Note

Case-sensitive collation is not addressed in the case sensitivity setting for configuration management project names.

Note

If the installer cannot alter an existing Integrity Lifecycle Manager database due to case mismatches, workflow and document and configuration management functionality will have to run in separate, named databases on separate Integrity Lifecycle Manager servers.

Locale Mismatch When Installing With Database Repository Option

When installing the Integrity Lifecycle Manager server with the database repository option, a mismatch can occur between the installer JVM and server JVM locales when setting the default case sensitivity for the SQL Server database. The mismatch does not occur when the case sensitive option is enabled for the database.

To correct the locale mismatch problem, run the following SQL statement:

```
update CMSCHEMAPROP set VALUE='en_US' where NAME='CMInsensitivityLocale'
```

If you encounter any problems installing or using the database repository, contact Customer Support.

Starting Server in Safe Mode

Starting the server in safe mode provides the application administrator with a way to perform configuration tasks while preventing all other users from accessing the system. While in safe mode, from either the GUI or the CLI, the administrator may configure global permissions and administer the MKS Domain.

Note the following about using the Integrity Lifecycle Manager server in safe mode:

- Only one user may log in.
- All ACL permission verification is disabled for the user logged in.
- The administrator does not have access to workflow and document management, and configuration management functions.

To start the Integrity Lifecycle Manager server in safemode from the CLI

To start the Integrity Lifecycle Manager server in safe mode, use the `mksis` command (see “Using mksis to Perform Server Administration” on page 119). The following is the command syntax for starting the Integrity Lifecycle Manager server in safe mode:

```
installdir/bin/mksis console safemode password
```

where

- *installdir* is the Integrity Lifecycle Manager server installation directory
- *password* creates a new password for this safemode session

The user name for logging into the server in safe mode is `safemode`, and the password is the password specified on server start.

You can script the server start in safemode by modifying the following file:

```
installdir/config/
```

The properties are:

```
wrapper.java.additional.8=-Dmks.safemode=
wrapper.java.additional.9=-Dmks.safemode.password=
```

Running Integrity Lifecycle Manager Server Diagnostics

You run diagnostics to monitor and diagnose issues that may arise when performing operations on the Integrity Lifecycle Manager client and server. The client, and the `si diag` and `im diag` CLI commands offer several diagnostics you can run on the workflow and document management, and configuration management components.

Note the following:

- The `si diag` and `im diag` commands are not supported for general customer use. Only use the `si diag` and `im diag` commands under the guidance of a representative from PTC Technical Support.
- To view the **Server Diagnostics** node in the Integrity Lifecycle Manager administration client, you must have the `AdminServer` permission for the `mks` ACL.
- To view the **Workflows and Documents ▸ Server Diagnostics** node in the Integrity Lifecycle Manager administration client, you must have you must have the `AdminServer` permission for the `mks.im` ACL.
- To view the **Configuration Management ▸ Server Diagnostics** node in the Integrity Lifecycle Manager administration client, you must have the `Admin` permission for the `mks.si` ACL.
- Server metrics and certain diagnostics may take a long time to run, impacting server performance. You should run them during non-peak usage.

To run server diagnostics in the GUI

1. In the Integrity Lifecycle Manager administration client, select one of the following nodes, depending on the type of diagnostic that you want to run:

- **Server Diagnostics**
- **Workflows and Documents** ▸ **Server Diagnostics** (see Workflow and Document Diagnostics on page 93.)
- **Configuration Management** ▸ **Server Diagnostics** (see Configuration Management Diagnostics on page 95.)

2. Select one of the diagnostics referenced in Step 1.
3. To specify which machine the diagnostic is run on, select one of the following values in the **Target** field.
   - `client` runs the diagnostic on the Integrity Lifecycle Manager client. This option is only available for diagnostics that work on the client.
   - `server` runs the diagnostic on the Integrity Lifecycle Manager server. This is the default.
   - `proxy` runs the diagnostic on the Integrity Lifecycle Manager FSA proxy server.
   - `cluster` runs the diagnostic on all the nodes in the Integrity Lifecycle Manager server cluster. The diagnostics information is grouped by node name.

   Note the following when running server diagnostics:

   - For the `server`, `proxy`, and `cluster` targets, you can choose to run diagnostics on all or on individual Integrity Lifecycle Manager client sessions connected to the applicable Integrity Lifecycle Manager server(s). The options available to you in the **Target** list depend on your server configuration. For example, you can choose to run diagnostics on **All client sessions connected to the server** and **All client sessions connected to all servers in the cluster**.
   - Not all diagnostics require a target.
4. To run the selected diagnostic, click **Run** at the top of the diagnostic view. The results display in the diagnostic view.
5. To perform a text search on the results, click anywhere in the diagnostic view and press CTRL+F. A search bar displays at the bottom of the diagnostic panel. As you type in the **Find** field, the results become more specific.
6. For some diagnostics, you can save the information to a file by clicking **Save Output**. A standard save dialog box appears, allowing you to save the information as a text file on the server’s file system.

To run server diagnostics in the CLI

From the command line, type one of the following commands:

- Server diagnostics on the configuration management component:

```
si diag --diag=value --target=client|proxy|server
```

- Server diagnostics on the workflow and document management component:

```
im diag --diag=value --target=client|proxy|server
```

where *value* is the diagnostic name, and where:

- `--target=client` runs the diagnostic on the Integrity Lifecycle Manager client.
- `--target=server` runs the diagnostic on the Integrity Lifecycle Manager server.
- `--target=proxy` runs the diagnostic on the Integrity Lifecycle Manager FSA server.

Note

- ○ The default target is `server`.
- ○ Some diagnostics cannot be run on the Integrity Lifecycle Manager client.
- ○ In general, diagnostics that are run using `--target=client` do not require server level permissions. However, diagnostics that are run using `--target=proxy` require the `DebugProxy` and the `AdminServer` permission as noted in the following table. Workflow and Documents diagnostic commands require the `AdminServer` permission. Some Configuration Management diagnostic commands require the `DebugProxy` or `AdminServer` permission, as noted in the following table. Permissions can be set on the `mks`, `mks:im`, or `mks:si` ACL unless otherwise noted.

Acceptable diagnostics for *value* are as follows:

Diagnostic	Description
`backupdatabase`	Backs up the FSA server database to *`installdir`*`/bin/backup/derby.db/`. The `AdminServer` permission is required. `Admin` permission is not required.
`connections`	Displays list of all server connections. List includes user name, host name, port, and time connection established. The `DebugServer` permission is required.
`CollectSupportPackage`	Collects client/server logs and properties in to compressed ZIP file (called support package). When the support package is assembled, you can e-mail to PTC Technical Support for

Diagnostic	Description
	further diagnosis. The AdminServer permission is required. Admin permission is not required. When called as im diag -diag=CollectSupportPackage -param=full, the generated support package includes the previous week's statistics. Each statistic is stored separately in the ZIP file as a separate file, with the statistics under a directory named Stats. Each file under the Stats directory is in the format yyyy.MM.dd-HH.mm.ss. When -param=full is not specified, these statistics are not included. The support package ZIP file also contains a ServerVersionInfo file that includes the following output: Version Information: Product Version: 10.8.0 Build Number: 8356 API Version: 4.16.8356 Hotfixes: <comma-separated list of hotfix ids> For more information, see "Gathering Troubleshooting Information for PTC Technical Support" on page 111.
cpservers	Displays a list of configuration management servers which have change packages tracked by items on this Integrity Lifecycle Manager server. No specific permissions are required to run this diagnostic.
dumpstacks	Collects server stacktrace. Stacktraces are appended to existing files in the following directory: *installdir*/bin.
getNextCheckpointDate	Displays the timestamp of the earliest conflicting transaction. If there are no conflicting transactions, none is displayed. The DebugServer permission is required. Applies only as a configuration management server diagnostic.
isproperties	Displays list of server properties not specific to workflow and document management, or configuration management. The DebugServer permission is required.

Diagnostic	Description
	To configure statistics collection, purging, and license monitoring properties, use the setispolicy diagnostic.
Licenses	Displays each type of license (workflow and document management; configuration management; seat; float) and user that has license currently checked out. The AdminServer permission is required. Admin permission is not required. **Note** Users on multiple Integrity Lifecycle Manager servers show license checked out for each server. If multiple servers connected to same license server, license information may not reflect the actual number of licenses currently checked out.
listCachedBulkdata	Displays records of a specified archive present in the bulkdata cache along with the following information: - HostName - Cache (indicates name of the cache) - Archive path - Revision - Cached Since - File Size (KB) To run this diagnostic, use the following command: `si diag -diag=listCachedBulkdata -target=<value> -param=<archivePath> -param=<revision>` where: - --target=*<value>* specifies proxy or server or proxy:all - --param=*<archivePath>* specifies the path where the archive resides. This parameter is mandatory. - --param=*<revision>* specifies the revision of the archive. This parameter is

Diagnostic	Description
	optional. If this parameter is not specified, the output displays all revisions present in bulk data cache for the specified archive. When the diagnostic is run using `--target=proxy`, the `AdminProxy` permission is required. When the diagnostic is run using `--target=server`, the `AdminServer` permission is required. When the diagnostic is run using `--target=proxy:all`, the `AdminServer` permission is required. **Note** • This diagnostic does not support `--target=client`. • This diagnostic is supported on the configuration management component.
`listopendbconnections`	Displays list of all open database connections. Each open database connection shows the following information: • Where the connection was requested from the JDBC connection pool (via a stacktrace). • The name of the thread that requested the JDBC connection. • The timestamp when the connection was pulled out of the JDBC connection pool The `DebugServer` permission is required.
`listOpenSITransactions`	Displays the current open DB backend transactions that could potentially conflict with a checkpoint operation; along with their stacktrace, transaction time, and thread name. This diagnostic is useful for diagnosing checkpointing issues. The `DebugServer` permission is required. Applies only as a configuration management server diagnostic.
`log`	Displays recent server log output. The `DebugServer` permission is required.

Diagnostic	Description
	To display the most recent 500 lines from the server log, use the appender `MEMORY`. For example, `-diag=log MEMORY`. To display the most recent 50 lines from the server log at or above the `ERROR` level, use the appender `MEMORYERRORS`. For example, `-diag=log MEMORYERRORS`. If there are no log messages with the category `ERROR`, then nothing is returned. ⚠ **Caution** This diagnostic uses the server's memory. **Note** An appender conveys what kind of logging the user is looking for. There are 2 options, `MEMORY` and `MEMORYERRORS`. It is not mandatory to specify an appender. If no appender is specified, by default `MEMORY` would be used.
`metrics`	Displays server metrics, such as the maximum change package entries per change package, and the average time entries per user. The `DebugServer` permission is required.
`migrateRevisionsFromDBToFileVaultAsPerPolicy`	Migrates archive revision data from the database to the configured file vault. This option migrates archive data only when vaulting is configured for the server. Applies only as a configuration management server diagnostic. For more information on file vaulting, see the "File Vaulting for Configuration Management" topic in the *Integrity Lifecycle Manager Help Center*.
`policy`	Displays list of all modified global policy settings. User preferences, default settings, and project settings not displayed. Applies only as a configuration management server diagnostic.
`properties`	Displays list of all properties on server. Includes settings in `si.properties`, `is.properties`, `im.properties`, and

Diagnostic	Description
	`logger.properties`.
`refreshUserGroupCache`	Clears framework-level caches (subject, user name, group, and LDAP). This is useful for updating user and group caches when your external directory service is updated. The `DebugServer` permission is required. **Note** If you specify this option with `im diag`, it has the additional effect of rebuilding the Integrity Lifecycle Manager-level caches.
`running`	Displays list of charts, dashboards, queries, and reports that are currently running. The `DebugServer` permission is required.
`sqllogging`	If SQL logging is enabled on the Integrity Lifecycle Manager (`im/si logging` command), you can set SQL logging levels for specific users. Setting the logging level can help with isolating specific user commands and improving server performance. For example, if logging levels are high and server performance is impacted, reducing logging levels may improve performance. The `DebugServer` permission is required for configuration management logging. To set SQL logging levels for a user, type: `im/si -diag=sqllogging` *`username loggingglevel`* where *username* is the user ID of the user whose commands you want to log. *logginglevel* is one of the following number or text values: `high` (`0`), `medium` (`5`), `low` (`10`), or `off` (`-1`). For example, typing: `im diag -diag=sqllogging jriley high`

Diagnostic	Description
	logs all workflow and document SQL commands for the user `jriley` to the category `SQL-jriley`.

Diagnostic	Description
`serverversioninfo`	Displays build information, including product version number, build number, API version, and a list of all installed hotfixes. This information is also available in the file `ServerVersionInfo` that is available using the `collectSupportPackage` command.
`setispolicy`	Configures the following server policies. The `DebugServer` permission is required. Name: `mksis.statisticsInterval` Int Value: `7200` Default: `60` Min: `30` Max: `10000000` Description: Controls the number of seconds between collecting statistics. To diagnose intermittent server performance issues, lower this number. Name: `mksis.statisticsPurge` Default: `weekly` Description: Purges stored statistics `weekly` or `monthly`. After the specified period, only one statistic per day is stored, while others are deleted. To manually purge statistics, specify `none` and use SQL statements to purge the data. **Note** To prevent overloading the Integrity Lifecycle Manager server with statistics, especially if the interval is set low, one statistic per day is purged on each server start. Name: `mksis.serverLicenseInterval` Int Value: `60` Default: `60` Min: `0`

Diagnostic	Description
	Max: `10000000` Description: Controls how frequently (in seconds) server license usage is polled and recorded in the server statistics. You can disable this policy by setting the value to `0`. If collection is enabled, the minimum allowed value is `60` seconds. Name: `mksis.flexLicenseInterval` Int Value: `0` Default: `0` Min: `0` Max: `10000000` Description: Controls how frequently (in seconds) FlexNet license usage is polled and recorded in the server statistics. If multiple Integrity Servers are pointing at the same FlexNet license server, enable this policy on one Integrity Lifecycle Manager server only. You can disable this policy by setting the value to `0`. If collection is enabled, the minimum allowed value is `300` seconds. Name: `mksis.logging.additional` Description: Controls the logging levels when the Integrity Lifecycle Manager server is down and you cannot change the logging levels using the client. Use the following format: `im/si –diag=setispolicy mksis.logging.additional <Logging Category> :<Value> ,<Logging Category> :<Value> ......`
`showrepdir`	Displays list of projects, archives, and directories contained in database repository. Applies only as a configuration management server diagnostic. Requires the `si.repositoryBrowsingEnabled` property to be `True` in the `si.properties` file.

Workflow and Document Diagnostics

Diagnostic	Description
Change Logging Levels	Changes logging levels for the server. In the available fields, choose the category and logging level. Available logging categories include: ACL, AGENT, API, CACHE, DEBUG, ERROR, GENERAL, IM-NOTIFICATION, LDAP, REALM, SD, SMTP, SQL, TM, and WARNING. By default, DEBUG is selected. To permanently save your changes, select true. By default, false is selected. For more information, see the logging information in the Installation and Configuration documentation.
Change Package Servers	Displays a list of configuration management servers which have change packages tracked by items on this Integrity Lifecycle Manager server.
Client Connections	Displays list of all clients connected to the server. List includes user name, host name, port, and time connection established.
Server Errors	Displays the most recent 50 lines from the server log at or above the ERROR level.
Server Log	Displays the most recent 500 lines from the server log.
Collect Support Package	Collects client/server logs, statistics, and properties into compressed ZIP file (called support package). When support package assembled, you can e-mail to PTC Technical Support for further diagnosis. For more information, see "Gathering Troubleshooting Information for PTC Technical Support" on page 111.
Connected Source Servers	Displays a list of configuration management servers connected to the Integrity Lifecycle Manager server.
Database Connections	Displays list of all open database connections. Each open database connection shows where (via a stacktrace) the connection was requested from the JDBC connection pool, the name of the thread that requested the JDBC connection, and the timestamp when the connection was pulled out of the JDBC connection pool.
JVM Memory Usage	Performs garbage collection and displays memory in use, free memory, and total memory in bytes.
License Usage	Displays each type of license (workflow and document management, configuration management, seat, float) and user that has license currently checked out.

Diagnostic	Description
	Note Users on multiple Integrity Lifecycle Manager servers show license checked out for each server. If multiple Integrity Lifecycle Manager servers connected to same license server, license information may not reflect actual number of licenses currently checked out. If license monitoring is enabled, this diagnostic also displays the appropriate stats categories. For more information on FlexNet license monitoring, For more information, see the licensing information in the Installation and Configuration documentation.
`Metrics`	Displays server metrics, such as the maximum change package entries per change package, and the average time entries per user.
`Running`	Displays a list of charts, dashboards, queries, and reports that are currently running.
`Stacktrace`	Collects server stacktrace.
`Statistics`	Displays server statistics. When you call the `CollectSupportPackage` diagnostic from the command line with the `full` parameter, the previous week's statistics are stored in a support package. Each statistic is stored separately in the ZIP file as a separate file, with the statistics under a directory named `Stats`. Each file under the `Stats` directory is in the format `yyyy.MM.dd-HH.mm.ss`.
`Verbose Garbage Collection`	Enables or displays verbose garbage collection for the server JVM. To diagnose garbage collection issues, the server retains the 10 most recent garbage collection logs if verbose garbage collection is enabled. A new garbage collection log (`gc.out`) is automatically started after the server restarts. Versions of `gc.out` are named incrementally, for example, `gc.out.1` and `gc.out.2`. **Note** Verbose garbage collection slows down server performance. Disable this option as soon as you complete your analysis.

Diagnostic	Description
`Slow Queries`	Displays a list of queries that are running very slow and presents their database plans.
`allowSharedItemsInQuery`	Causes a specific query to return shared items in the query results. By default, shared items are not included in query results. Use the following syntax. To show shared items in the query, set this value to `true`. `im -diag=allowSharedItemsInQuery` `<username>:<query name> <true|false>` **Note** In versions of the GUI client that are earlier than 10.9, this setting may not work as expected. For more information, see “Workflows and Documents: Known Issues” in the *Integrity Lifecycle Manager Release Notes*.

Configuration Management Diagnostics

Diagnostic	Description
`Change Logging Levels`	Changes logging levels. This allows you to enable appropriate logging levels within logging categories depending upon your requirements. In the available fields, choose the category and logging level. Available logging categories include: `ACL`, `AGENT`, `API`, `CACHE`, `DEBUG`, `ERROR`, `GENERAL`, `IM-NOTIFICATION`, `LDAP`, `REALM`, `SD`, `SMTP`, `SQL`, `TM`, and `WARNING`. By default, `DEBUG` is selected. For example, SQL logging category at level 5 produces some X and Y types of SQL statements. This enables the appropriate logging levels within those logging categories depending on the requirements. To permanently save your changes, select `true`. By default, `false` is selected. For more information, see the logging information in the Installation and Configuration documentation.
`Client Connections`	Displays a list of all clients connected to the Integrity Lifecycle Manager server including user name, host name, port, and time connection established.

Diagnostic	Description
	Client connections through a proxy display `via <proxy machine name>` at the end of the entry. Clients connections through the Web display `web` at the end of the entry. User connections are represented in list field by a sessionid, dash, and the logged in username. You can also sort by current Remote Method Invocation (RMI) connection times in cases where you have a large number of users and are trying to troubleshoot a specific user(s).
`Disconnect User`	Allows you to disconnect connected users from an Integrity Lifecycle Manager server. User connections are represented in the list field and are signified by a session ID, dash, and the logged in user name.
`Client Log`	Allows you to retrieve the Integrity Lifecycle Manager client log applicable to the session and save it in a specified local file.
`Server Errors`	Displays the most recent 50 lines from the server log at or above the `ERROR` level.
`Server Log`	Displays the most recent 500 lines from the server log.
`Collect Support Package`	Collects client/server logs, statistics, server migration and hotfix Information, reports, and properties into compressed ZIP file (called support package). When support package assembled, you can e-mail to PTC Technical Support for further diagnosis.
`Database Connections`	Displays list of all open database connections. Each open database connection shows where (via a stacktrace) the connection was requested from the JDBC connection pool, the name of the thread that requested the JDBC connection, and the timestamp when the connection was pulled out of the JDBC connection pool.
`JVM Memory Usage`	Performs garbage collection and displays memory in use, free memory, and total memory in bytes.
`License Usage`	Displays each type of license (workflow and document management, configuration management, seat, float) and user that has license currently checked out.

Diagnostic	Description
	Note Users on multiple Integrity Lifecycle Manager servers show license checked out for each server. If multiple Integrity Lifecycle Manager servers connected to same license server, license information may not reflect actual number of licenses currently checked out. If license monitoring is enabled, this diagnostic also displays the appropriate stats categories. For more information on FlexNet license monitoring, see the licensing information in the Installation and Configuration documentation.
`Response Times`	Displays database and file system response times (in milliseconds) for the Integrity Lifecycle Manager server and for its proxy servers.
`Metrics`	Displays server metrics, such as the maximum change package entries per change package, and the average time entries per user.
`Stacktrace`	Collects server and client stacktraces. Also provides the option for you to view all Integrity Lifecycle Manager client sessions connected to the Integrity Lifecycle Manager server and all Integrity Lifecycle Manager client sessions connected to all servers in the cluster.

Diagnostic	Description
`Statistics`	Displays server statistics. When you run the `CollectSupportPackage` diagnostic from the command line with the `full` parameter, the ZIP file that is created contains the previous week's statistics by default. Each statistic is stored as a separate file in the `Stats` directory in the format `yyyy.MM.dd-HH.mm.ss`. If you are using a clustered server configuration, statistics for each cluster node are stored in separate node name subdirectories under `Stats`.
`Verbose Garbage Collection`	Enables or displays verbose garbage collection for the server JVM. Also provides the option for you to view all Integrity Lifecycle Manager client sessions connected to the Integrity Lifecycle Manager server and all client sessions connected to all servers in the cluster. To diagnose garbage collection issues, the server retains the 10 most recent garbage collection logs if verbose garbage collection is enabled. A new garbage collection log (`gc.out`) is automatically started after the server restarts. Versions of `gc.out` are named incrementally, for example, `gc.out.1` and `gc.out.2`. **Note** Verbose garbage collection slows down server performance. Disable this option as soon as you complete your analysis.

Integrity Server Statistics

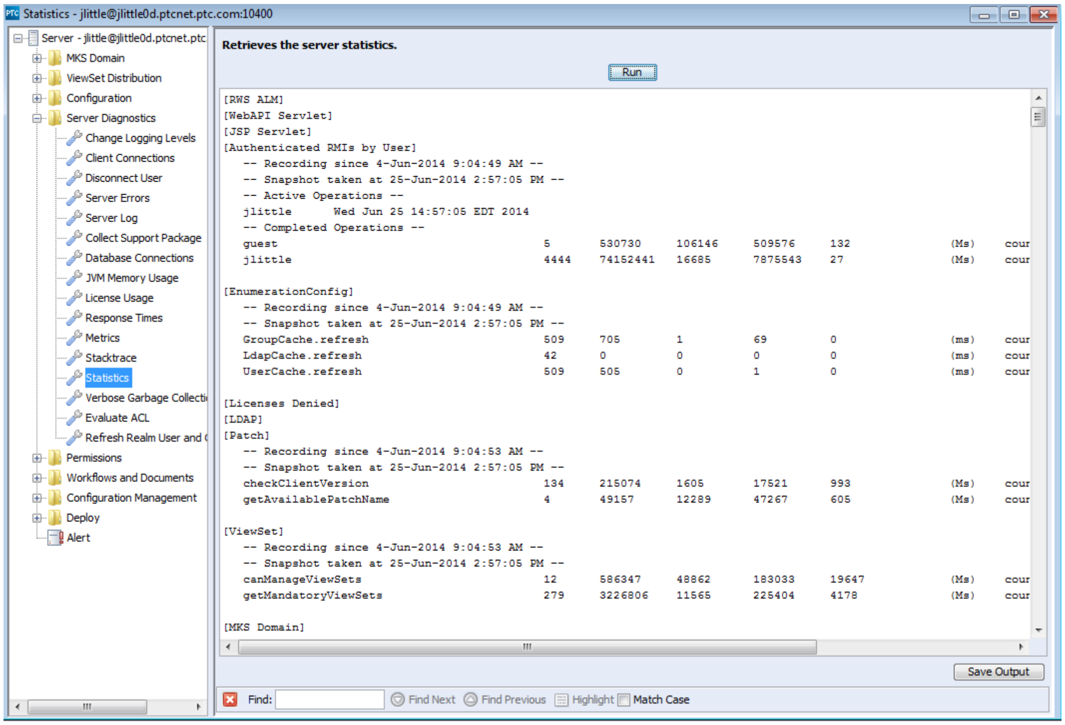
When running, the Integrity Lifecycle Manager server generates and maintains statistics about various operations performed on the server. These statistics can provide insight into the overall performance of operations and identify outliers for further investigation.

You can use the generated statistics to get an overview of how a server is performing over a period of time and gain insight into areas of the product that are not performing optimally and may require tuning.

Because the collected statistics are cumulative over time and are reset when the server restarts, the best time to collect statistics is when the server has been up and running for a significant period of time. Statistics data collected immediately after a server restart generally does not contain enough information to be useful.

Viewing Server Statistics

You can view the Integrity Lifecycle Manager server statistics in the **Server Diagnostics** section of the Integrity Lifecycle Manager administration client. In that section, select **Statistics** and click **Run**. This retrieves and displays the statistics that have been collected from the Integrity Lifecycle Manager server. Click **Save Output** to save the statistics to a text file.



```
[RWS ALM]
[WebAPI Servlet]
[JSP Servlet]
[Authenticated RMIs by User]
   -- Recording since 4-Jun-2014 9:04:49 AM --
   -- Snapshot taken at 25-Jun-2014 2:57:05 PM --
   -- Active Operations --
   jlittle      Wed Jun 25 14:57:05 EDT 2014
   -- Completed Operations --
   guest                                      5       530730     106146     509576     132              (Ms)    cour
   jlittle                                    4444    74152441   16685      7875543    27               (Ms)    cour

[EnumerationConfig]
   -- Recording since 4-Jun-2014 9:04:49 AM --
   -- Snapshot taken at 25-Jun-2014 2:57:05 PM --
   GroupCache.refresh                         509     705        1          69         0                (ms)    cour
   LdapCache.refresh                          42      0          0          0          0                (ms)    cour
   UserCache.refresh                          509     505        0          1          0                (ms)    cour

[Licenses Denied]
[LDAP]
[Patch]
   -- Recording since 4-Jun-2014 9:04:53 AM --
   -- Snapshot taken at 25-Jun-2014 2:57:05 PM --
   checkClientVersion                         134     215074     1605       17521      993              (Ms)    cour
   getAvailablePatchName                      4       49157      12289      47267      605              (Ms)    cour

[ViewSet]
   -- Recording since 4-Jun-2014 9:04:53 AM --
   -- Snapshot taken at 25-Jun-2014 2:57:05 PM --
   canManageViewSets                          12      586347     48862      183033     19647            (Ms)    cour
   getMandatoryViewSets                       279     3226806    11565      225404     4178             (Ms)    cour

[MKS Domain]
```

Note

The **Server Diagnostics** section of the Integrity Lifecycle Manager administration client is only visible to users with the `AdminServer` permission on the `mks` (Server Permissions) ACL.

Interpreting Server Statistics

Server statistics are divided into several categories. Each category is formatted using the same general structure:

```
[Server Info]
   -- Recording since 2014-6-25 13:14:51 --
   -- Snapshot taken at 2014-6-25 13:16:58 --
   -- Active Operations --
   getStatistics Wed Jun 25 13:16:58 EDT 2014
```

```
-- Completed Operations --
getLocalHostname         1    28      28      28      28     (Ms) count/tot/avg/max/min/unit
getUserVisiblePolicies   1    469110  469110  469110  469110 (Ms) count/tot/avg/max/min/unit
hasAdminPermission       4    928     232     305     125    (Ms) count/tot/avg/max/min/unit
```

The first line is the category name. Related statistics (for example, events, operations, and so on) are grouped under the category name.

The line beginning with `Recording since` provides the date and time that the server statistics were last reset. This normally corresponds to the date and time of the last restart of the server.

The line beginning with `Snapshot taken` reflects the time and date that the statistics in this category were last updated (collected). Clicking **Run** to display the server statistics triggers the collection of the statistics. Thus, this is always the time when you clicked **Run**.

The `Active Operations` section in a category lists all of the operations that are currently running on the server at the time the statistics were collected. No metrics are recorded for these operations as they have not yet completed. However the number of operations listed in this section can provide a general idea of the load that the server is experiencing at a particular time.

Finally, the `Completed Operations` section lists all events (for example, actions or invocations) and objects (for example, licenses, items or relationships) tracked in this category. Each line contains a tab separated list with the following information:

- the event or object name
- one or more columns of values
- a unit column
- a column legend indicating what data each column in that line represents

For example, when a line has a column legend of `count/tot/avg/max/min/units`, the columns following the event or object name would be the following (in order):

`count`

For events, this is the number of times the event occurred. For objects, this column identifies how many of that object (as identified by the `units` column) exist.

`tot`

For events, this is the total time spent performing the event. For objects, this column often does not appear or does not apply. However, for some objects (those with `#bytes` or `bytes` in the `units` column), this is the total number of bytes used by all such objects.

avg

For events, this is the average time spent performing the event. For objects, this column often does not appear or does not apply. However, for some objects (those with `#bytes` or `bytes` in the `units` column), this is the average number of bytes used by a single object.

max

For events, this is the maximum amount of time spent performing a single event. For objects, this column often does not appear or does not apply. However, for some objects (those with `#bytes` or `bytes` in the `units` column), this is the maximum number of bytes used by a single object.

min

For events, this is the minimum amount of time spent performing a single event. For objects, this column often does not appear or does not apply. However, for some objects (those with `#bytes` or `bytes` in the `units` column), this is the minimum number of bytes used by a single object.

units

This column identifies, in parentheses, the units in which the other columns are measured. For events, this is usually either `(Ms)` (microseconds) or `(ms)` (milliseconds). For objects, this often identifies the type of object being measured. For example, `(licenses)` indicates that the number of licenses is being measured.

For full details on interpreting server statistics, see article CS179133 on the PTC Integrity eSupport portal.

Running Integrity Lifecycle Manager server as Application

Sometimes it is useful to run the Integrity Lifecycle Manager server as an application instead of as a service. If the Integrity Lifecycle Manager server does not start, different error messages may appear on the command line that can be recorded and sent to PTC Technical Support. Additionally, you can obtain a thread dump in instances where the Integrity Lifecycle Manager server experiences performance issues.

To run the Integrity Lifecycle Manager server as an application

1. Stop the Integrity Lifecycle Manager server service from a command line by changing to *installdir*`/bin`, and typing the following:

   ```
   mksis[.bat] stop
   ```

2. If you are not using the default administrative user, ensure that the administrator belongs to the administrators group.

3. If you are not logged in as this administrator, log out and log in as the administrator.
4. From a command line, change to *installdir*/bin and type the following:

   ```
   mksis[.bat] console
   ```

   The command output is sent to the `startup.log` file. For more information on the command, see "Using mksis to Perform Server Administration" on page 119.
5. To display a thread dump, press CTRL+BREAK. This only displays what is happening at the exact moment. So you may have to press CTRL+BREAK several times to determine how threads change over time.
6. Copy the thread dump information to a text file and send it to PTC Technical Support.

Integrity Lifecycle Manager server Logs

The following Integrity Lifecycle Manager server log files can be useful in diagnosing server errors:

- *IntegrityServerInstallDir*\log\server.log records server errors and events. By default, a new log is automatically started after the previous log reaches 10 MB. Versions of server.log are named incrementally, for example, server.1.log and server.2.log. By default, no more than 50 previous server.log versions are stored. To configure the size of the log file and the number of logs, configure the relevant server property.
- *IntegrityServerInstallDir*\uninstall\.com.zerog.registry.xml records all the activities that occurred during the installation of the Integrity Lifecycle Manager server.
- *IntegrityServerInstallDir*\log\dbmigration.log is created when a database is migrated, for example, during an Integrity Lifecycle Manager upgrade. The output displays database version information, error messages, and SQL statements run against the database as part of the migration process.
- *IntegrityServerInstallDir*\log\dbcreation.log is created when a new database is created during the Integrity Lifecycle Manager server install. The output displays database version information, error messages, and SQL statements run against the database as part of the database creation process.
- *IntegrityServerInstallDir*\hs_err_pid*.err typically appears if a Java error occurs, causing the server to stop running. These files

are generated by the Java Virtual Machine (JVM) and consist of a stack trace at the time the server stops running.

Tip

To receive instant updates about Integrity Lifecycle Manager server errors, you can configure the server to e-mail server log error messages. For more information, see the Logging information in the Installation and Configuration documentation.

Client Logging Levels

To increase the level of logging on the Integrity Lifecycle Manager client, uncomment the desired logging levels in *`IntegrityClientInstallDir`*`\IntegrityClientSite.rc`:

```
# START #
    # Turn up logging.
     #
   IntegrityClient.log.exception.includeCategory.WARNING=10
   #IntegrityClient.log.exception.includeCategory.DEBUG=10
   IntegrityClient.log.exception.includeCategory.ERROR=10
   IntegrityClient.log.exception.includeCategory.GENERAL=10
   IntegrityClient.log.message.includeCategory.WARNING=10
   #IntegrityClient.log.message.includeCategory.DEBUG=10
   IntegrityClient.log.message.includeCategory.ERROR=10
   IntegrityClient.log.message.includeCategory.GENERAL=10
   #IntegrityClient.log.message.includeCategory.CACHE=10
   #IntegrityClient.log.exception.includeCategory.CACHE=10
   #IntegrityClient.log.message.includeCategory.EVENT=10
   #IntegrityClient.log.exception.includeCategory.EVENT=10

   # Changes Formatting
   #IntegrityClient.log.message.defaultFormat={2}({3}) {5}\: {4}\n
   #IntegrityClient.log.exception.defaultFormat={2} {4} {5}\: {6}\n

   # END   #
```

By default, client HTTP-related messages, such as socket termination messages after each command, do not appear in the log. For example, if a low-level HTTP protocol fails, you can turn on HTTP logging to assist in determining what the exact problem might be.

To display all client HTTP-related messages that may appear when using the client Web interfaces, add the following lines to the `IntegrityClientSite.rc` file:

```
IntegrityClient.log.message.includeCategory.HTTP=10
IntegrityClient.log.exception.includeCategory.HTTP=10
```

Logging output displays in the value `logFileName`, or in the *`IntegrityClientInstallDir`*`\bin\IntegrityClient.log` file. You can adjust logging between `0` and `10`. After you make changes to `IntegrityClientSite.rc`, save it, and restart the Integrity Lifecycle Manager client.

To display a timestamp with each log entry, add the following lines to the `IntegrityClientSite.rc` file:

```
IntegrityClient.log.message.defaultFormat={2}({3}) {5}\: {4}\n
    IntegrityClient.log.exception.defaultFormat={2} {4} {5}\: {6}\n
```

si logging

In addition to modifying the `logger.properties` file, you can use the `si logging` command to increase or decrease logging levels for several different categories (as outlined in the table next). You can run this command from a client or server machine. You can use any category included in the `logger.properties` file. Note the following:

- The client and server do not need to be restarted after making changes.
- Logging changes last only until the Integrity Lifecycle Manager server or client is restarted.
- You need the `DebugServer` permission in the `mks:si` ACL to run this command.
- `server.log` file logs information about this command when it runs.

From a command line, type:

```
si logging --category=value --level=value
```

where

- `--category=`*`value`* specifies the category name (see the table next for category names)
- `--debug` is equivalent to `--category=DEBUG`
- `--level=`*`value`* specifies the log level (`0` disables logging, `10` logs all messages)
- `--off` is equivalent to `--level=-1` (no reporting in this category)
- `--on` is equivalent to `--level=10` (logs all messages in this category)
- `--target=`*`value`* specifies the target debugging is enabled for. Valid values are `client`, `server` (default), and `proxy`.

Acceptable values for `--category=`*`value`* are:

Category	Description
`DEBUG`	Logs debug messages. **Note** Command overrides settings in `logger.properties`, but only until Integrity Lifecycle Manager server restarts. Additionally, option does not explicitly log debug exceptions. To log exceptions, open `logger.properties`, uncomment `mksis.logger.exception.includeCategory.DEBUG`, and set value to `10`.
`SQL`	Logs all SQL commands to `server.log`. For example, if you view item, `SELECT` statement logged. When editing item, `INSERT`, `UPDATE`, and `SELECT` statements logged. `-level=5` logs all SQL commands. `-level=10` adds additional information such as Rollback time.
`CACHE`	Logs cache operation information. Levels 0–3 not verbose. Levels 4–15 verbose.
`SMTP`	Logs communication between Integrity Lifecycle Manager server and SMTP server.
`IM-NOTIFICATION`	Logs Integrity Lifecycle Manager e-mail notification.

`ACL`	Logs permission checking to `server.log`. For example, add label command logs following:
	`2009-08-16 13:45:17,140 INFO [mksis.IntegrityServer] ACL(5): Check user administrator for permission CreateProject against acl mks:si. Resolved ACL: mks:si`
	`Decision: GRANTED`

`TRANSACTION`	Logs all transactions performed by specific user, for example, `si logging -category=TRANSACTION-jdoe -on` logs all transactions performed by `jdoe`. To log all cache transactions, type `TRANSACTION-system`
`SICI`	Provides communications between workflow and document management functionality and configuration

Category	Description
	management functionality, and communications between Integrity Lifecycle Manager client and server.
REALM	Logs NT realm information.
API	Logs Integrity API information.
HLL	Logs Integrity HLL server information.
LDAP	Logs LDAP security scheme information.
AGENT	Logs RMIs from Integrity Lifecycle Manager client to server.
SOLUTIONS	Logs all solution data model operations and configuration information.
MKSALM	Logs all server side solution trigger operations.
SDMCACHE	Logs information when a solution command attempts to access an item from the Integrity Lifecycle Manager server.
MKSALMAUDIT	Logs all server-side information for document model auditing.
FINDER	Logs all information related to the operation of the Finder dialog box.

You can also use the `si logging` command to enable debug logging on the Integrity Lifecycle Manager client. To enable debug logging on the client, use:

```
si logging --debug --on --target=client
```

Valid values for `--target` are `client`, `server` (default), and `proxy`.

im logging

In addition to the `si logging` command, you can use the `im logging` command to increase or decrease logging levels for several different categories (as outlined in the preceding table). You can run this command from a client or server machine.

From a command line, type:

```
im logging --category=value --level=value
```

where

- `--category=`*`value`* specifies the category name (see the table following for category names)
- `--debug` is equivalent to `--category=DEBUG`
- `--level=`*`value`* specifies the log level (`0` disables logging, `10` logs all messages)
- `--off` is equivalent to `--level=-1` (no reporting in this category)

- `--on` is equivalent to `--level=10` (logs all messages in this category)
- `--target=`*`value`* specifies the target the debugging is enabled for (valid values are `client`, `server` (default), and `proxy`.

Acceptable values for `--category=`*`value`* are as outlined for the `si logging` command.

Integrity Lifecycle Manager server Logging Levels

To assist you in diagnosing server issues, you can record server messages and exceptions to the `server.log`.

To adjust logging properties

1. Edit the following file:

   ```
   IntegrityServerInstallDir\config\properties\logger.properties
   ```

2. Uncomment the category of logging messages or exceptions to record. Generic categories include: `DEBUG`, `DIAGNOSTIC`, `WARNING`, `GENERAL`, `ERROR`, and `FATAL`.

   Note

   All lines containing the word `DEBUG` are commented out. This suppresses logging of debugging messages, which may help to increase performance. When necessary, these lines can be uncommented out for debugging.

3. Set the level of logging information to record. Ten (`10`) is the highest level of logging, and zero (`0`) disables logging.

   For example, you can adjust the level of logging information for the `ERROR` category:

   ```
   mksis.logger.message.includeCategory.ERROR=10
   mksis.logger.exception.includeCategory.ERROR=10
   ```

   Tip

   Configure a server property to receive e-mail notifications about specific server messages or exceptions that are recorded to `server.log`.

4. Save your changes.
5. Restart the Integrity Lifecycle Manager server.

Miscellaneous Logging Properties

To debug security realms

1. To increase the amount of logging to `server.log` for the security realm, add properties to the *`IntegrityServerInstallDir`*`/config/properties/logger.properties` file.

   For verbose LDAP debugging, use:
   ```
   mksis.logger.message.includeCategory.LDAP=10
   ```
   Successful authentications are logged when you set:
   ```
   mksis.logger.message.includeCategory.AUTHENTICATOR=10
   ```
2. If necessary, uncomment the line for the realm you want to debug.
3. Save your changes.
4. Restart the Integrity Lifecycle Manager server.

To debug solutions

1. To debug installed solutions, add the following properties to the *`IntegrityServerInstallDir`*`/config/properties/logger.properties` file:

   - server-side category that logs trigger operations
     ```
     mksis.logger.message.includeCategory.MKSRQ=10
     ```
   - client and server-side category (can be set for both) that logs solution data model operations and configuration information
     ```
     mksis.logger.exception.includeCategory.IM-NOTIFICATION=5
     ```

     Note

     Due to heavy logging on the server, you should set this category on the client only.

   - server-side category that logs output when a solution command (`rq`) retrieves an item from the server
     ```
     mksis.logger.message.includeCategory.SDMCACHE=10
     ```
2. Save your changes.
3. Restart the Integrity Lifecycle Manager server.

Integrity Lifecycle Manager Error Codes

Errors displayed in the GUI, Web, CLI, API, and logs may contain an error code. The error code is of the form MKS000000, for example MKS004364. Use the error code to obtain more information on the error from the PTC Integrity eSupport portal. Information may include the cause of the error, its resolution, or a workaround. If multiple errors occurred from the action, then multiple error codes are displayed.

Integrity Lifecycle Manager client Logs

- `IntegrityClientInstallDir\bin\IntegrityClient.log` is the main Integrity Lifecycle Manager client log file. It logs information messages, warnings, and errors. By default, a new log is automatically started after the previous log file reaches 10 MB. Versions of `IntegrityClient.log` are named incrementally, for example, `IntegrityClient.1.log` and `IntegrityClient.2.log`. By default, no more than 20 previous `IntegrityClient.log` versions are stored.

  To edit maximum size of the log file, specify a value for the `logFileMaxSize` property in the `IntegrityClientInstallDir\IntegrityClientSite.rc` file.

  To edit maximum number of log files for the client log, specify a value for the `logFileMaxBackupIndex` property in the `IntegrityClientInstallDir\IntegrityClientSite.rc` file.
- `IntegrityClientInstallDir\uninstall\.com.zerog.registry.xml` records all activities that occurred during the Integrity Lifecycle Manager client installation.

Generating Server Stack Traces

A stack trace monitor for generating server stack traces is provided for cases where the Integrity Lifecycle Manager server is unresponsive (hangs). The stack trace monitor does not rely on the `si diag` command. The command writes the stack trace and date/time information to a stacktrace file.

To generate a stack trace, create a file called:

```
installdir/data/runstacktrace
```

where *installdir* is the path to the directory where you installed the Integrity Lifecycle Manager server.

Once you create the `runstacktrace` file, the Integrity Lifecycle Manager server then automatically creates the `installdir/bin/stacktrace` file and writes all stack trace and date/time information to it.

To stop generating stack traces, remove the `runstacktrace` file in the *`installdir`*`/data` directory.

You can also set the interval for the stack trace monitor to poll for the `runstacktrace` file. To set the monitoring interval, set the following property in the *`installdir`*`/data/is.properties` file:

```
mksis.monitorInterval=<number of seconds>
```

Note

By default, the monitoring interval is 30 seconds. Setting a monitoring interval of less than 10 seconds can have adverse effects on performance.

Investigating Integrity Objects Impacting Server Performance

If the Integrity Server is experiencing performance issues, an Integrity object (query, chart, report, dashboard) may be the cause. For example, a report that returns a large amount of data can impact server performance.

While you can notify a user about objects they created that you may suspect are impacting server performance, the user may be busy or unavailable. As an administrator, you can impersonate active users to investigate which object may be the cause of the performance issue. Once you impersonate a user, you have full access to objects created by them. Specifically, you can view, edit, and delete objects created by the user you impersonate.

Impersonating a user is also useful if a user leaves your organization. Before deactivating the user in Integrity, you can impersonate the user to review what objects they created. For example, you could share queries to other users or delete charts that are no longer useful.

Note

To impersonate a user, the Admin permission is required. For more information on impersonation, see the *PTC Integrity Integrations Builder Guide*.

For example, jriley (a developer) frequently runs reports that are suspected to be the cause of server performance issues.

1. To investigate jriley's reports, nsingh (the administrator) creates an impersonate ACL to impersonate jriley:
   ```
   aa  addaclentry  --acl=""mks:impersonate:user:jriley""
   ```

```
u=nsingh:Impersonate
```

2. nsingh views jriley’s reports:
   ```
   im  reports -g  --impersonateuser=jriley  --user=nsingh
   --password=password
   ```
3. nsingh views the report he suspects is causing the performance issue:
   ```
   im  viewreport -g  --impersonateuser=jriley  --user=nsingh
   --password=password  jrileyQuery
   ```
4. nsingh edits the report to reduce the amount of data the report returns:
   ```
   im  editreport -g  --impersonateuser=jriley  --user=nsingh
   --password=password  jrileyQuery
   ```

Gathering Troubleshooting Information for PTC Technical Support

If you are unable to diagnose a problem with the Integrity Lifecycle Manager server using the server diagnostics, you can collect client and server logs, server statistics, and properties into a compressed ZIP file (called a support package). When your support package is assembled, you can e-mail it to PTC Technical Support for further diagnosis.

Note the following:

- If certain files do not exist on the Integrity Lifecycle Manager client or server, within the ZIP file each client and server’s `manifest.txt` file indicates which files are missing. Additionally, each `manifest.txt` file includes the date and time the files were collected, the target component (client, server, or proxy), the full path to each file, and the date and time each file was last modified.
- If the support package exceeds 5 MB, you should remove some of the older `server.log` (or for previous server versions, `weblogic.log`) files from the support package before e-mailing it to PTC Technical Support.
- By default, the support package does not contain server statistics. You can include server statistics by running the `CollectSupportPackage` diagnostic from the command line with the `full` parameter. Each server statistic is stored as a separate file in the `Stats` directory. Each file is in the format `yyyy.MM.dd-HH.mm.ss`.
- The five most recent `license.log` files are captured in the support package.

For security reasons, the following sensitive information is not included in the support package:

- in `si.properties`:
  - `im.user=xxxx`

    - `im.password=xxxx`
    - `si.anonymousUser=xxxx`
    - `si.anonymousPassword=xxxx`
- in `im.properties`:
    - `mksis.im.prodPassword=xxxx`
    - `mksis.im.prodUser=xxxx`
- in `is.properties`:
    - `mksis.proxy.*.adminPassword=xxxx`
    - `mks.dbUser=xxxx`
    - `mks.dbPassword=xxxx`
    - `mksis.privatekey.password=xxxx`
- in `security.properties`:
    - `ldap.credential=xxxx`
    - `ldap.principal=xxxx`
    - `ldap.credential=xxxx`
    - `ldap.principal=xxxx`
- clear text passwords

To create a support package in the GUI

1. From the Integrity Lifecycle Manager administration client, connect to the Integrity Lifecycle Manager server that you want to retrieve server files from.
2. From the tree pane, select the **Workflows and Documents** or **Configuration Management** node, and then select **Server Diagnostics** ▸ **Collect Support Package**.
3. In the Target ZIP file for Support Package field, type a name for the ZIP file, for example, `support.zip`, or click **Browse** and locate an existing ZIP file to add the information to.
4. Click **Run**. You are notified that the support package has been created.
5. E-mail the support package to PTC Technical Support. For tips on what information to include when you contact PTC Technical Support, see "Environment Information" on page 113.

Note

Support packages generated from the Integrity Lifecycle Manager administration client do not include a Stats subdirectory with server statistics.

To create a support package in the CLI when the Integrity Lifecycle Manager server is not running

From a command line, go to *`installdir`*`/bin` and type one of the following commands:

Note

This command collects server files only.

On Windows:

```
collectSupportPackage.exe zipFilename [full]
```

On UNIX:

```
collectSupportPackage zipFilename [full]
```

where *zipFilename* specifies the name of the ZIP file, for example, `support.zip`.

When the optional `full` parameter is specified, the generated support package includes the previous week's statistics. Each statistic is stored separately in the ZIP file as a separate file, with the statistics under a directory named Stats. Each file under the Stats directory is in the format yyyy.MM.dd-HH.mm.ss. When `full` is not specified, these statistics are not included.

Gathering Important Information

Environment Information

When reporting an issue to PTC Technical Support, the following information is commonly requested:

- customer number
- operating system, RAM, and CPU speed of the client machine
- operating system, RAM, and CPU speed of the server machine
- if using a Web interface, the Web browser version

- version of the Integrity Lifecycle Manager client and server, including service packs installed
- database type and version
- exact error message (a screen shot is helpful)
- steps to reproduce the problem
- properties and log files (for more information, see "Gathering Troubleshooting Information for PTC Technical Support" on page 111)

Common Support Questions

To identify the problem category, PTC Technical Support may ask the following questions:

Are your platforms on the supported platforms list?
To view this list, browse to:

http://www.ptc.com/support/integrity.htm

Does the problem happen for all users?
If yes, then it is likely a server problem.

Can you reproduce the problem on a different machine with the same configuration?
If yes, then it is likely a client problem.

Can another user only reproduce the problem on the same machine?
If yes, then it is likely a machine problem.

Integrity Lifecycle Manager Customer Number

To determine your customer number, do one of the following:

- Browse to `http://`*hostname*`:`*port*. Your customer number displays near the bottom of the Integrity Lifecycle Manager server home page.
- Look for a sticker on the top of the product box.
- Look for the `mksis.customerInfo.customerNumber` property in:

  `IntegrityServerInstallDir``/config/properties/is.properties`

Integrity Lifecycle Manager Server Version

To determine your Integrity Lifecycle Manager server version, browse to `http://`*hostname*`:`*port*. The version and build number appear at the bottom of the Integrity Lifecycle Manager server home page.

Integrity Lifecycle Manager Client Version

To determine your Integrity Lifecycle Manager client version, do one of the following:

- From a command line, type one of the following:
    - `si about` (for configuration management)
    - `im about` (for workflow and document management)
    - `integrity about` (Integrity Lifecycle Manager administration client)
    - `aa about` (Authorization Administration)
- From the GUI or Web interfaces, select **Help ▸ About**.

Integrity Lifecycle Manager Client and Server Directory Listing

If PTC Technical Support asks for a directory listing of the Integrity Lifecycle Manager server and the client directories, change to the directory that the server is installed in or the directory that the client is installed in.

On Windows, use the following command to record the directory listing to a file that can be sent to PTC Technical Support:

```
dir /b /s > listing.txt
```

On UNIX, type:

```
ls –R > listing.txt
```

On Windows, the default *IntegrityServerInstallDir* is `c:\Program Files\Integrity\ILMServer11` and the default *IntegrityClientInstallDir* is `c:\Program Files\Integrity\ILMClient11`.

Is Integrity Lifecycle Manager Server Running?

1. On the Integrity Lifecycle Manager server, open:

   ```
   IntegrityServerInstallDir\log\server.log
   ```

2. Look for lines of the following form:

   ```
   INFO [IntegrityServer] GENERAL(0): Listening on clear
       port portnumber
   INFO [IntegrityServer] ApplicationState(0): Integrity Server
       (Build:<buildnumber>, Service Pack: <servicepacknumber>) started
   ```

   where *portnumber* is your Integrity Lifecycle Manager server’s port number.

   No lines should follow those lines that indicate a shutdown occurred.

   If you are using SSL, the following line is also in the log:

   ```
   INFO [IntegrityServer] GENERAL(0): Listening on secure
        port secureportnumber
   ```

where *secureportnumber* is your Integrity Lifecycle Manager server's port used for secure SSL connections.

What Time Zone Are You In?

For data integrity purposes, Integrity Lifecycle Manager server records what time zone it is in. If the time zone of the server differs from that of the database, the server does not start. View the `server.log` file to determine which time zone the server is expecting for each.

Following is an example from the log of an error for two differing time zones:

```
17:59:38,331 WARN  [ServiceController] Problem starting service
    mks:name=ServerTimeZone
17:59:38,284 ERROR [Security] Starting failed mks:name=ServerTimeZone
java.lang.Exception: Timezones and/or daylight savings settings don't match.
    Database: America/New_York, server: America/Buenos_Aires
```

Is FlexNet Server Running?

Note

If FlexNet is unavailable, Integrity Lifecycle Manager server shuts down and displays an error message.

- On Windows
  1. To start the FlexNet configuration utility, start:
     ```
     FlexNetInstallDir\bin\lmtools.exe
     ```
  2. On the **Service/License File** tab, click **Configuration using Services**.
  3. If multiple services are listed, select the service associated with the Integrity Lifecycle Manager server. By default, this is `FlexNet Service 1`.
  4. Click **Server Status**.
  5. Click **Perform Status Enquiry**. If the FlexNet server is running, a message notifies you that the server is running, and it displays the location of the license file and how many licenses are available.
- On UNIX
  1. From a command line, type:
     ```
     ps –ef | grep lmgrd
     ```
     This searches a complete listing of all processes running and displays the `lmgrd` process (the FlexNet Manager daemon).
  2. To use the `lmstat` command line utility (in `flexnet/bin`) to perform a status inquiry, type:

```
lmstat -a -c[license file]
```

For more information on this utility, browse to http://www.flexerasoftware.com.

Directing License Server Output to a Log

You can direct output from the FlexNet license server manager to a local log file. For detailed information on managing Report Log and Debug Log files for the licensing server, consult the FlexNet Publisher documentation provided in the `flexnet.zip` file with your Integrity Lifecycle Manager product components.

Differencing Integrity Lifecycle Manager Server Properties Files

If you change properties files containing properties not in the database and make backup copies, differencing changed properties files against the backup copies can be useful in diagnosing server problems.

Note

Create backup copies of the properties files located in *`IntegrityServerInstallDir`*`\config\properties`.

To difference properties files

1. Start the Visual Differencing tool.
2. Select **File** ▸ **Compare**.
3. Browse for the two files that you want to compare. and click **OK**. The differences between the two files are highlighted.

Creating IDE Integrations Logs

Logging can be set for IDE integrations, such as Sybase PowerBuilder.

1. Add the following lines to the *`userprofile_dir`*`\IntegrityClient.rc` file:

```
integrations.debugLogFile=c\:\\installdir\\log\\mksapi.log
integrations.debugLogLevel=5
```

Note

You must use double slashes to escape slashes and colons.

The `debugLogLevel` property can be set to any value between `0` and `10`, where `0` does not display any information and `10` displays the most information. To reduce the amount of extraneous information, use `debugLogLevel=5`.

2. Restart the client.

Sybase PowerBuilder 8.0

In PowerBuilder 8.0, source control logging is set up on a per-workspace basis in the connection profile.

IBM Eclipse 2.0/Websphere Studio 5.0

Websphere does not log version control activity specifically, but version control errors appear in Websphere's general debug log located in *`workspace dir`*`\.metadata\.log`.

To display the debug log, system properties, and the list of plug-ins, select **Help ▸ About**, and click **Configuration Details**.

To log source control activity in PowerBuilder 8.0

1. Right click a workspace, and choose **Properties**.
2. Click **Source Control**.
3. To enable trace logging, select **Log All Activity**.

Note

The default log file name is `pbscc80.log` and is saved in the workspace directory, but you can change the name and location.

If `Overwrite Log File` is selected for PowerBuilder 8.0, the log file is overwritten for each session. The default setting is `Append To Log File`.

Configuring the Location of Dr. Watson Logs

If the client or server quits on a Windows platform, useful information is typically logged to the machine's Dr. Watson log. On Windows 2000, you can configure the location of the produced log file and dump by running `C:\winnt\system32\drwtsn32.exe`.

Using mksis to Perform Server Administration

The `mksis.bat` (`mksis` in UNIX) file provides some administrative functionality for the Integrity Lifecycle Manager server. The file is located in the following directory:

```
installdir/bin/mksis[.bat]
```

where *installdir* is the Integrity Lifecycle Manager server installation directory.

The following is the syntax for the command:

```
mksis subcommand
```

where *subcommand* is one of the following:

- `console` starts the Integrity Lifecycle Manager server in console mode as an application only without the service as specified in:

  ```
  installdir/config/
  ```

- `install` installs the Integrity Lifecycle Manager server as an Windows NT service
- `remove` removes the Integrity Lifecycle Manager server service
- `restart` restarts the Integrity Lifecycle Manager server service
- `safemode` restarts the Integrity Lifecycle Manager server in safe mode (see "Starting Server in Safe Mode" on page 81)
- `start` starts the Integrity Lifecycle Manager server service (Windows) or daemon (UNIX)
- `stop` stops the Integrity Lifecycle Manager server service (Windows) or daemon (UNIX)

Using isutil to Manage the Database Repository

To assist with managing the database repository, the Integrity Lifecycle Manager server includes a series of `isutil` commands. The `isutil` commands allow you to perform a variety of maintenance and configuration tasks related to the database repository while the Integrity Lifecycle Manager server is not running.

If you experience cache corruption issues or are restoring your repository from a backup, a cold restart of the Integrity Lifecycle Manager server may be required. The `isutil` command allows you to flush and rebuild all or specific caches when you restart the server, resulting in a potentially lengthy downtime before users can access the server. To determine if you need to perform a cold restart, contact Customer Support.

The `isutil` utility is installed with the Integrity Lifecycle Manager server in the *`installdir`*`/bin` directory.

⚠ **Caution**

> Certain `isutil` commands can permanently alter the database and cannot be undone. Use these commands only with guidance from PTC Technical Support or Integrity Lifecycle Manager Professional Services.
>
> Command usage is:

```
isutil -c command [-?] [-d] [command-arguments]
```

where *command* is one of the following:

- `aclcreate` creates or resets the ACL table in the Integrity Lifecycle Manager server database.
- `acldestroy` deletes the ACL table in the Integrity Lifecycle Manager server database
- `aclexists` tests whether the ACL table exists in the Integrity Lifecycle Manager server database.
- `aclmigrate` migrates the ACL entries table to the latest version (database schema). No action is taken if the tables are already at the current schema level.
- `blobperf` tests the performance of BLOB access in the Integrity Lifecycle Manager server database.
- `cmactiveon` enables the maintenance of ArchiveShared and active-project properties.
- `cmalterarchivebasename` changes the archive base name.
- `cmalterprojectbasename` changes the project base name.
- `cmdbcreate` creates and initializes the configuration management database repository tables.
- `cmdbmigrate` migrates the configuration management database repository tables to the latest version (database schema). No action is taken if the tables are already at the current schema level.

⚠ **Caution**

The cmdbmigrate command cannot be reversed because there is no reverse migration tool. Because the new database schema does not operate with older versions of Integrity Lifecycle Manager, ensure that you have a backup of your database before running the command.

- `cmlist` lists the names of the requested object type from the database.
- `dbinstall` sets up the database similar to the Integrity Lifecycle Manager server installer.
- `diag` runs an Integrity Lifecycle Manager server diagnostic. For a list of diagnostics, see "To run server diagnostics in the CLI" on page 83.
- `generateserversalt` creates a new server salt for MKS Domain passwords. For more information, see "MKS Domain Security Realm" on page 32.
- `imdbcreate` creates and initializes the workflow and document database tables.
- `imdbdestroy` deletes the workflow and document database tables.
- `imdbgetversion` retrieves the current workflow and document database schema version.
- `imdbmigrate` migrates the workflow and document database tables to the current Schema.
- `imdbrebuildtable` rebuilds the specified database table. You can specify the location for the table to be stored, another location for its LOB data, and another location for its indexes (for example, `isutil -c imdbrebuildtable <tablename> [<tablelocation> [<loblocation> [<indexlocation>]]]`).
- `metrics` runs database metrics. For a list of metrics, see "To run server diagnostics in the CLI" on page 83.

- `migrateserverconfig` migrates your existing Integrity Lifecycle Manager server configuration to a new Integrity Lifecycle Manager server installation.

> **Note**
>
> If you configured Oracle SQL*Net encryption, the LAX file updates are not migrated. You must reconfigure these files manually. For more information, see the "Configuring Oracle SQL*Net Encryption" topic in the *Integrity Lifecycle Manager Help Center*.

- `sidbcreate` creates and initializes the configuration management tables.

An example of `isutil` command usage is as follows:

```
C:\Program Files\Integrity\ILMServer11\bin\isutil -c aclcreate
```

4

Uninstalling Integrity Server

Uninstalling Integrity Lifecycle Manager server

Before attempting to uninstall the Integrity Lifecycle Manager server, make sure the Integrity Lifecycle Manager server service has been stopped and removed. If you are using the Derby embedded database, the *installdir*`/data/derby.db` directory is not removed when you uninstall the Integrity Lifecycle Manager server. Even though the directory is retained, making backups of this directory before uninstalling the Integrity Lifecycle Manager server is a recommended practice.

For more information on stopping the Integrity Lifecycle Manager server and removing the Integrity Lifecycle Manager server service, see "Running Integrity Server" on page 60.

To uninstall the Integrity Lifecycle Manager server on Windows

1. Stop the Integrity Lifecycle Manager server, and remove the Integrity Lifecycle Manager server service.
2. Do one of the following:
   - Launch the uninstall program file:
     ```
     installdir/uninstall/IntegrityServerUninstall.exe
     ```
   - Use **Add or Remove Programs** in the Windows Control Panel.

   The uninstall window opens.

   Note

   If you installed the server using a silent installation, the uninstallation is also silent. No dialog box appears and no further action is required.

3. Click **Uninstall**. All installed components are removed, with the exception of any files or folders created after the installation.
4. To exit the uninstaller, click **Done**.

To uninstall the Integrity Lifecycle Manager server on UNIX

1. Stop the Integrity Lifecycle Manager server, and remove the Integrity Lifecycle Manager server service.
2. Make sure the environment variable `$DISPLAY` is set, if necessary.

> **Note**
>
> If you installed the server using a silent install, the uninstall is also silent. You do not need to set this variable.

3. Launch the uninstall program file:

   ```
   installdir/uninstall/IntegrityServerUninstall
   ```

> **Note**
>
> If you installed the server using a silent install, the uninstall is also silent. No dialog box displays and no further action is required.

4. Click **Uninstall**. The uninstaller program removes all installed components, except any files or directories created after the installation.
5. To exit the uninstaller, click **Done**.

II

Upgrading Integrity Lifecycle Manager

5

Upgrading to Integrity Lifecycle Manager 11.1

This document contains information that you must know before upgrading to Integrity Lifecycle Manager 11.1. The information is relevant for upgrades from Integrity 10.5 and later.

For the most up-to-date upgrading information made available to you since the publication of this document, see article CS256463 on the PTC Integrity eSupport portal.

For the most current product platform support information, see Supported Platforms.

For the most current Knowledge Base articles, go to http://www.ptc.com/support/integrity.htm.

For detailed information about configuring the supported databases for the Integrity Lifecycle Manager server, see the *Integrity Lifecycle Manager Help Center*.

For detailed information about new features, general notes, fixed issues, and known issues in Integrity Lifecycle Manager 11.1, see the *Integrity Lifecycle Manager Release Notes*.

Prerequisites

Before upgrading, PTC recommends reviewing this entire guide. Depending on your current version of Integrity Lifecycle Manager, your upgrade path may require a full install using an executable or you may be able to upgrade using a service pack install.

If you are currently using Integrity version earlier than 11.0, then you must run the installer to upgrade to Integrity Lifecycle Manager 11.1. If you are currently using 11.0, then you can either run the installer or apply the 11.1 service pack to your existing 11.0 server.

Before you begin upgrading to Integrity Lifecycle Manager 11.1, you must read the General Considerations on page 131 section first. Then, read the subsequent sections relevant to your current release.

For example, if you are currently using Integrity 10.7, read the General Considerations on page 131 section, then read the sections Considerations for Upgrading From Integrity 10.7 to Integrity 10.8 on page 139 through Considerations for Upgrading From Integrity Lifecycle Manager 11.0 to Integrity Lifecycle Manager 11.1 on page 144. You do not have to read the sections that contain information about releases older than your current release.

General Considerations

Have a backup plan

Develop a plan to use in case an upgrade fails. By retaining the previous version of the server, you reduce downtime if the upgrade fails and allows you to restore the database. In case an upgrade fails, you can contact PTC Technical Support for assistance.

Backup and verify your database

Perform a full database backup and keep the backup on site until it is determined that the upgrade is successful. Verify the database backup to ensure it can be restored if required.

Consider integrated components

Consider the dependencies between integrated components when upgrading. In case of split servers, you must keep the IM and SI server data in sync to accommodate these dependencies. When you take a backup your IM server, you must also ensure to back up your SI server at the same time. The backed up data can be very useful in case of failed upgrades when you have to restore data back to your system.

Do not reuse backed up configuration files

If you upgrade using the full install executable, note that old configuration files cannot be reused with the new release. Copying and pasting files from the old installation into the new server directory causes problems if configuration properties have been renamed or new variables added. A utility is provided for migrating configuration files.

In addition, many properties are now contained in the database. Once properties are successfully migrated to the database with the `isutil` utility, they can be further modified in the Integrity Lifecycle Manager administration client GUI or from the CLI using the `integrity setproperty`, `si setproperty`, `im setproperty`, and `aa setproperty` commands.

Always stop the Integrity Lifecycle Manager server service before installing the upgrade

Stop the Integrity Lifecycle Manager server service, but do not uninstall the Integrity Lifecycle Manager server until after the new server is installed. Retaining the old installation ensures that a rollback is possible if you run into difficulties with the upgrade.

Custom integrations compatibility

If you are upgrading to Integrity Lifecycle Manager 11.1 from Integrity 10.6 or earlier, and you have a custom integration, additional steps for updating your integration can be necessary. You must perform these extra steps if all of the following criteria apply:

- Your environment uses a private copy of Integrity Lifecycle Manager C API
- Your environment uses the Integrity Lifecycle Manager server as the integration point
- SSL is used for secure communication with the server integration point

If your environment meets all of these criteria, you must update your copy of the Integrity Lifecycle Manager C API to the version provided with Integrity Lifecycle Manager 11.1. If you are using a Microsoft Excel or Project integration, you must also update to the currently supported version of the integration software. For supported versions, see the latest Integrations document on the PTC website. If you do not upgrade the Integrity Lifecycle Manager C API, the integration will be unable to connect to the secure port of the Integrity Lifecycle Manager server.

Considerations for Upgrading from Integrity 10.5 to Integrity 10.6

Before upgrading from Integrity 10.5, you must consider the following:

Extended localization support

Integrity 10.6 extended localization support on the client and server from English and Japanese to three additional languages: German, Chinese

Simplified, and Chinese Traditional. However, PTC does not currently support or recommend changing the server language. For example, you cannot change the server language from English to Chinese Simplified.

Compatibility of Item Presentation Templates in Integrity 10.6 and later (966764)

If you open an Integrity 10.5 and earlier item presentation template (IPT) in an Integrity 10.6 and later release, Integrity adds a header, populated with the `Item Created Information` and `Item Modified Information` read-only fields. If there is a logo in the IPT, it is placed in the header, based on the logo alignment property. Integrity also applies the default text style to fields in the header.

If you open an Integrity 10.6 and later IPT in Integrity 10.5 and earlier, Integrity ignores the read-only fields in the header and moves any custom fields to the first tab in the IPT. As of Integrity 10.6, Integrity no longer includes a logo property for IPTs because the header layout can be customized just like the layout of any tab. This allows you to add more than one image in the header.

Large amounts of Test Results data in Oracle databases can increase Integrity 10.6 upgrade time (275480, 950629)

If you are using an Oracle database that contains a large amount of Test Results data, the database migration step of the Integrity server upgrade can take a long time. During the upgrade, a message warning of this appears and provides instructions to look for status updates in the migration log (`dbinstall.log`) found in the server installation log directory.

Revision description audit tags in text files (976757)

In Integrity 10.6 and later, revision annotation `--- Added comment ---` tags are replaced with `- Added comment -` tags. As a result, users can see additional differences when differencing files from a Sandbox that contains the `$Log$` keyword. These differences are resolved by updating the working file in the Sandbox. New tags use the format `- Added comment -`. Users do not see additional differences.

Enhanced project visibility for project administrators (961444)

To provide enhanced project security, project administrators can view and edit only the projects which they are assigned. For example, visible projects are listed in the Integrity Lifecycle Manager administration client GUI >**Projects** node and also when using the `im projects` command.

To reflect this enhancement from the CLI, the `im dynamicgroups` and `im editdynamicgroup` commands contain the following added options and updated option behavior.

- `im dynamicgroups --fields=`*membership* displays only the projects to which the project administrator is assigned.

  To indicate that a project does not have any groups and users as members of the dynamic group, specify the new `nomembers` keyword. For example, specify `--membership=/Project=nomembers`. Previously, a blank space indicated that a project did not have any groups and users as members of the dynamic group. For example, the following was previously acceptable: `--membership=/Project=`.

⚠ **Caution**

If you are a project administrator, the `im dynamicgroups --fields= membership` command displays a subset of the projects in your Integrity Lifecycle Manager configuration. If a super administrator uses that list of projects when specifying `im editdynamicgroup --membership`, the membership for the projects that the project administrator does not have permission to view are removed.

- `im editdynamicgroup --membership` processes only the projects to which the project administrator is assigned.

  To indicate that a project does not have any groups and users as members of the dynamic group, specify the new `nomembers` keyword. For example, specify `--membership=/Project=nomembers`. Previously, a blank space indicated that a project did not have any groups and users as members of the dynamic group. For example, the following was previously acceptable: `--membership=/Project=`.

  To inherit the membership from the parent project to the dynamic group, specify the `inherit` keyword. For example, specify `--membership=/Project=inherit`. Previously, a membership list was specified.

  To allow a project administrator to set membership for a specific project to which he or she is assigned, use the `--projectMembership=`*project*`= inherit|nomembers|`*per-project-membership* option,

  where:

  ○ *per-project-membership* is in the form *user-list*|*group-list*|*user-list:group-list*

  where:

  ○ *user-list* is in the form `u=`*username*`[`*,username*`]`

- ○ *group-list* is in the form `g=`*groupname*`[`,*groupname*`]`
    - ◆ `inherit` specifies that the membership for the parent project is to be inherited by the dynamic group.
    - ◆ `nomembers` specifies that the project does not have any groups and users as members of the dynamic group

Note

Specifying users but no groups removes any existing groups. Similarly, specifying groups but no users removes any existing users.

To correctly work with dynamic groups, PTC recommends using the options provided with these commands. For more information, see the 10.6 and later version of the CLI man pages.

If a project administrator is using a 10.5 and earlier Integrity Lifecycle Manager administration client, the GUI and CLI commands return results based on 10.5 and earlier behavior.

Configuration Options For Non-Build Subprojects When Creating a Development Path (972187, 972193)

When creating a development path from a project that contains non-build subprojects (normal or variant subprojects), the options are configurable from the Integrity Lifecycle Manager client. However, an administrator can override and enforce a specific option on the Integrity Lifecycle Manager server.

If one of these options is enforced on a 10.6 and later Integrity server and a 10.5 and earlier Integrity client creates a development path, the default behavior is enforced. All subprojects that are configured differently from the parent project retain their existing configuration.

Improperly using `im dynamicgroups` or `im editdynamicgroup` can result in lost membership data (96020, 148955, 976751)

The `--fields=membership` and `--membership` options were designed so that a script could update project membership for a dynamic group by retrieving the membership for all projects with `im dynamicgroups --fields=membership`, making changes to the list, and then returning the list to `im editdynamicgroup --membership`.

In Integrity 10.6 and later, the `im dynamicgroups --fields=membership` and `im editdynamicgroup --membership` commands have been changed so that they only return or edit the memberships for projects that the project administrator has rights to administer. For example, if a user is a project administrator, the list of members returned from `im` `dynamicgroups --fields=membership` only includes the projects that the project administrator is allowed to administer. Similarly, if a project administrator attempts to edit membership using `im editdynamicgroup --membership`, Integrity Lifecycle Manager only updates those projects that the project administrator can administer.

With these changes in Integrity 10.6 and later, scripts should not call `im dynamicgroups --fields=membership` as a project administrator and then use the returned list to call `im editdynamicgroup --membership` as a different project administrator. If a project is missing from the list passed to `im editdynamicgroup --membership`, Integrity Lifecycle Manager sets that project's membership to inherit its parent project's membership. This means that using these two commands with different project administrators causes the memberships for some projects to inherit from their parent project, losing their membership data if the two project administrators administer a different set of projects.

Creating, viewing, editing, copying, and deleting configuration management policies from the CLI (949198)

From the CLI, administrators can create, view, edit, copy, and delete configuration management policies using the following commands:

- `si copypolicysection`
- `si deletepolicysection`
- `si setpolicysection`
- `si viewpolicysection`
- `si viewpolicysections`

Note

The `si setpolicysection` command replaces the `si createpolicysection` command (a previously unsupported command that was removed in Integrity 10.6). To create and edit policies, use the `si setpolicysection` command and modify any existing scripts that refer to the `si createpolicysection` command.

These commands are also published commands supported by PTC for use with the Integrity Lifecycle Manager Java API. For more information, see the *Integrity Lifecycle Manager Integrations Builder Guide*.

Using scripts or the Integrity Lifecycle Manager Java API, administrators can automate the setup of configuration management policies. For more information on CLI commands, see the CLI man pages.

Note

To manage your configuration management policies, PTC recommends using the Integrity Lifecycle Manager administration client.

Deprecated options for the `im editissue` command (976630)

When using the `im editissue` command in Integrity 10.6 and later, the following options are deprecated and should no longer be used:

- `--addRelationships=`*value*
- `--removeRelationships=`*value*

Instead, use the newer options:

- `--addFieldValues=`*value*
- `--removeFieldValues=`*value*

In addition, the `--addRelationships=`*value* option is deprecated in the `im createissue` and `im copyissue` commands. Instead, use the newer `--addFieldValues=`*value* option.

Although the deprecated options continue to work, PTC does not recommend using them for new scripts because they will be removed in a future release. Existing scripts using the deprecated options should use the new options.

New method added to `ScriptDynamicGroupBean` (917243)

To check the user membership of a dynamic group based on the project, the `isUserMemberOf` method was added to `ScriptDynamicGroupBean` in Integrity 10.6. This updated method can be used in an event trigger to restrict the following:

- creating an item
- editing a project on an item (cannot change to specific projects)
- recording time entries
- editing test results
- label operations on items

For more information, see the "Event Trigger Java Documentation" documentation link on the Integrity Lifecycle Manager server Homepage.

Referencing checkpoints in unregistered configuration management projects (982628)

In Integrity 10.6 and later, project paths are referenced using the repository location. This allows you to reference checkpoints in unregistered (dropped) configuration management projects. If a checkpoint is currently referenced and the corresponding project is later dropped, the checkpoint remains accessible as read-only. For example, you can continue to open the referenced checkpoint from an SI project field or create a build Sandbox from the checkpoint for auditing purposes.

Before upgrading, note the following:

- Project paths are referenced using the repository location, which can affect existing scripts and triggers. PTC recommends reviewing any existing scripts and triggers.
- Integrity 10.5 and earlier clients cannot view metrics links for SI Project fields that reference build projects created with Integrity 10.6 and later.
- If you are using different versions of the Integrity Lifecycle Manager client and Integrity Lifecycle Manager server and Integrity Lifecycle Manager server dedicated to configuration management, the behavior is dependent on the version of the Integrity Lifecycle Manager server dedicated to configuration management. For an Integrity server 10.5 and earlier that is dedicated to configuration management, a flat path is stored for build projects.

Considerations for Upgrading from Integrity 10.6 to Integrity 10.7

Integrity 10.7 is the earliest supported release that you can directly upgrade to Integrity Lifecycle Manager 11.1. Integrity Lifecycle Manager servers before release 10.7 cannot be upgraded automatically. If your version of Integrity is earlier than 10.7, contact PTC Technical Support for assistance.

Before upgrading from Integrity 10.6, you must consider the following:

Migrate relationship data to new database table prior to upgrade

Integrity 10.7 introduced a new database storage model and table for relationship data.

Prior to upgrading to Integrity 10.9 and later, the item and document relationship data must be migrated to the new database format using a 10.7 or 10.8 Integrity server.

Note

If the migration has not occurred prior to attempting an upgrade to Integrity 10.9 (or later), the upgrade does not succeed but the database is still usable to run the original version of Integrity Lifecycle Manager. If you are performing a silent install, the following error is logged in `dbinstall.log`:

```
ERROR(0): Database migration aborted. The migration
to the IIDeltaMap table must be performed prior to
upgrading to Integrity 10.9 (and later). For migration
information, see the Integrity 10.8 version of the PTC
Integrity Upgrading Guide.
```

The new relationship table is more compact and grows at a significantly slower rate than the relationship table that it replaces. For more information on migrating relationship data, consult the 10.8 version of the *Integrity Lifecycle Manager Installation and Upgrading Guide*.

Considerations for Upgrading from Integrity 10.7 to Integrity 10.8

Before upgrading from Integrity 10.7, you must consider the following:

Capability for administrators to localize configuration and add translation

Integrity 10.8 introduces a capability for the administrator to localize their configuration and add translation strings for the display name and description attributes of the administrative objects **Fields**, **Test Result Fields**, **Types**, and **States**. During an upgrade of an existing server to Integrity 10.8 and later, the display name and description of the administrative objects (standard set and custom created) are stored in the server locale. The server locale cannot be changed during an upgrade. For example, if an administrator installs an Integrity 10.7 server and earlier, using English as the installer locale, then during an upgrade to Integrity 10.8 and later, the display name and description attributes of the administrative objects **Fields**, **Test Result Fields**, **Types**, and **States** are stored in the English locale.

For detailed information, see the “Configuring Localization” topic in the *Integrity Lifecycle Manager Help Center*.

Integrity 10.8 and later does not support PTC System Monitor 3.0

Integrity 10.8 and later is only compatible with PTC System Monitor (PSM) 4.0 and 5.0. It is not supported with PSM 3.0 and earlier PSM releases. This is due to an incompatibility with the version of Java that is used with both products. This incompatibility does not allow the Integrity server to start. If you are currently running PSM 3.0 and earlier, you must upgrade to 4.0 or 5.0 before upgrading to Integrity 10.8 and later.

Sandbox project.pj files are now stored in a client-side database

As of Integrity 10.8 and later, project information is stored in a client-side database, and consequently there are no `.pj` files in Sandboxes. Project files still display as virtual project files (with the `.pj` file extension) in Integrity Lifecycle Manager interfaces. Instead of project files, project information is stored in a client-side database in the `.mks` directory of the system on which the Integrity Lifecycle Manager client is installed. The location of the `.mks` directory is specified by the `MKS_IC_INSTANCE_DIR` environment variable. By default, on all platforms, the `.mks` directory can be located in the home directory of the user.

Note

An Integrity Lifecycle Manager user's `.mks` directory must have sufficient space available to fit three copies of the client-side database. The amount of space needed depends on how many Sandboxes the user has, and a minimum of 50 MB available space is recommended. This space is required to store two separate backups of the database.

Trigger bean updates to support deactivating and activating development paths (1010427, 1071406)

In Integrity 10.8 and later, the following updates have been made:

- The following new trigger bean classes are available: `ScriptActivateVariantArgumentsBean` and `ScriptDeactivateVariantArgumentsBean`. These classes make the variant name being activated available to script authors.
- For the `ScriptProjectBean` class, the following new methods are available: `deactivateVariant` and `activateVariant`. These methods deactivate and activate variant projects.
- For the `ScriptProjectBean` class, the `getVariants` method has been updated to return both active and inactive variants. If you only want active variants to be returned, you must update your scripts to use `getActiveVariants` instead.
- For the `ScriptProjectBean` class, the following new methods have been added: `getActiveVariants` and `getInactiveVariants`.

For more information, see the Javadocs.

Shared Sandboxes not supported as of Integrity 10.8

Shared Sandboxes are not supported for Integrity 10.8 and later. After the Integrity Lifecycle Manager client upgrade, only the owner of the Sandbox continues to have access to the Sandbox through the Integrity Lifecycle Manager client. Other users who were sharing the Sandbox can no longer access that Sandbox.

Sandboxes used for the Staging and Deploy functionality no longer supported as of Integrity 10.8

In Integrity 10.8 and later, the Sandboxes used for the Staging and Deploy functionality are no longer supported. The migration of such Sandboxes from earlier versions of Integrity to Integrity 10.8 and later is also no longer supported.

Staging and Deploy functionality is no longer supported in Integrity 10.8 and later

If the Staging and Deploy functionality is enabled (`mksagent.startup.sd=true`) in the `agent.properties` file, the Integrity Lifecycle Manager Agent fails to start. A `FATAL` category log message in the `agent.log` and `FATAL.log` files is also logged. The log message indicates that the Staging and Deploy functionality is no longer supported. Applying the Integrity 10.8 service pack to Integrity 10.7 disables the Staging and Deploy functionality by updating an existing `mksagent.startup.sd=true` property to `mksagent.startup.sd=false` in the `agent.properties` file. This automatic disabling of Staging and Deploy functionality on the Integrity Lifecycle Manager Agent is required to support remote automated patching.

Error when using Oracle 12c with Integrity 10.8

If your Integrity Lifecycle Manager server is running on an Oracle 12c database, the `ORA-01792: maximum number of columns in a table or view is 1000` error may occur. This error is related to a known Oracle 12c issue (Doc ID 1951689.1, Bug 17376322 “Select Statement Throws ORA-01792 Error”). This defect is addressed by the Oracle patch 19509982.

Considerations for Upgrading from Integrity 10.8 to Integrity 10.9

Before upgrading from Integrity 10.8, you must consider the following:

Backup files generated by Integrity 10.8 during multiple-row editing are not compatible with Integrity 10.9 (or later)

During multiple-row editing of a document, unsaved changes are stored in a backup file so that these changes can be recovered if an unexpected shutdown of the Integrity Lifecycle Manager client occurs. The backup files generated by Integrity 10.8, which introduced a beta version of multiple-row editing, cannot be opened by Integrity 10.9 (or later). Before upgrading an Integrity Lifecycle Manager client, ensure that all documents are saved or closed successfully. Otherwise, after the client is upgraded, you are unable to recover changes from a 10.8 backup file. Backwards compatibility is included in all releases subsequent to 10.9. For example, an Integrity Lifecycle Manager 11.1 client will be able to open backup files from Integrity 10.9. Future releases will also support recovering pending imports, which are new in Integrity 10.9.

Considerations for Upgrading from Integrity 10.9 to Integrity Lifecycle Manager 11.0

Before upgrading from Integrity 10.9, you must consider the following:

New installation directories

For Integrity Lifecycle Manager server 11.0 product versions, the default installation directory is:

installdir/Integrity/ILMServer11

In addition, the installer prevents you from installing the server in an existing installation directory.

For the Integrity Lifecycle Manager client, the default installation directory is:

installdir/Integrity/ILMClient11

PTC recommends reviewing and updating any scripts you use that reference the installation directories.

Upgrading FlexNet

To control the use of Integrity Lifecycle Manager components, it is mandatory that you install FlexNet Publisher License Server 11.13.1.3 that is distributed with Integrity Lifecycle Manager 11.0. This version of the FlexNet Publisher License Server installer includes an embedded Java Runtime Environment (JRE) installation and is also compatible with older versions of Integrity servers.

Note

You must manually copy over the `lmutil.exe` and `MKS.exe` (`lmutil` and `MKS` on Unix) files as part of the FlexNet Licensing Server installation. This must be done for new installs and upgrades. The Integrity Lifecycle Manager server will not start without completing this task. Refer to the `readme.html` file, included under the `/flexnet/doc` directory, within the Integrity Lifecycle Manager 11 server installer ZIP file.

Please note that if your FlexNet Licensing Server is on a Solaris platform and if you plan to upgrade the FlexNet Publisher License Server version to 11.13.1.3, then Integrity server versions 10.9 and older will not connect with this version of FlexNet. Therefore, you can perform any one of the following:

- You can either upgrade the older versions of your Integrity servers to Integrity Lifecycle Manager version 11.0 OR
- You can install the FlexNet Publisher License Server on a platform other than Solaris. For this option, you will need to generate a new license.

For further assistance, you can contact PTC Technical Support.

Plan to recompute active paths

As part of upgrading to Integrity Lifecycle Manager 11.0 and above, administrators must address an issue that affects the software configuration management functionality for all releases up to and including Integrity Lifecycle Manager 11.0. There was a limitation in active paths management, where it did not properly take into account the development paths that were created from subprojects. Consequently, features that reference those paths (such as the shared member indicator) did not properly account for members in development paths created at a subproject level. For those features to work correctly, the active paths must be recomputed for the entire repository. Until that operation is performed, features will continue to use the old (existing) active paths algorithm.

After upgrading to Integrity Lifecycle Manager 11.0, an administrator must recompute the active paths using the `isutil -c cmactiveon` command while the server is down. The command first clears the existing reference counts used by the active paths algorithm, and then recomputes the reference counts for the entire repository. For information on using the command, see the "Using isutil to Manage the Database Repository" topic in the *Integrity Lifecycle Manager Help Center*.

Note

Recomputing active paths can take several hours for large repositories. Plan for the server downtime by first running the command on a test server that contains a copy of the source repository. The time used for the test server can then be used to plan the operation on the production server.

Considerations for upgrading from Integrity Lifecycle Manager 11.0 to Integrity Lifecycle Manager 11.1

If you are using Integrity Lifecycle Manager 11.0, then you can use either the service pack or the full installer to upgrade to version 11.1. If you are using Integrity version older than 11.0, then you must use the full installer to upgrade to Integrity Lifecycle Manager 11.1.

Compatibility Support

The next several topics provide information about compatibility:

- Integrity Lifecycle Manager client and Server Compatibility on page 144
- Configuration Management Repository Compatibility on page 146
- Dedicated Integrity Lifecycle Manager server Compatibility on page 146
- Integrity Lifecycle Manager server and Integrity Lifecycle Manager Agent Compatibility on page 147
- Integrity Lifecycle Manager API Compatibility on page 147
- Implementer and Integrity Lifecycle Manager server Compatibility on page 147

Integrity Lifecycle Manager Client and Server Compatibility

Integrity Lifecycle Manager server 11.1 supports connections from Integrity client 10.5 through 11.1.

If you are currently using an FSA server and upgrading from an earlier release to Integrity Lifecycle Manager 11.1, you do not have to perform the upgrade all at once. Integrity Lifecycle Manager server 11.1 supports connections from a proxy Integrity server 10.5 and an Integrity Lifecycle Manager server through 11.1.

Note the following:

- Upgrade the components of the system in the following order:
  1. Workflows and Documents and Test Management-enabled servers
  2. Configuration Management-enabled servers
  3. FSA servers
  4. Clients
- By default, the `servicepack.policy` file specifies the minimum Integrity Lifecycle Manager client version and service pack that can connect to Integrity Lifecycle Manager 11.1 server. However, you can configure these values as new service packs are released. For more information on configuring the `servicepack.policy` file, refer to the *Integrity Lifecycle Manager Help Center*.
- Integrity Lifecycle Manager supports the installation of multiple Integrity Lifecycle Manager clients on a single machine. For example, a 11.1 client and a 10.8 client. This is useful for accessing functionality available in specific releases and connecting to different versions of the Integrity Lifecycle Manager server. For more information, see the *Integrity Lifecycle Manager Help Center*.
- The Web interface version does not depend on the client version.
- ViewSets edited with a new Integrity Lifecycle Manager client may be unusable on older clients, and will have an adverse impact on your users if those ViewSets are configured to be mandatory. For example, a ViewSet edited with a 11.1 Integrity Lifecycle Manager administration client and published to a 11.1 server may only be usable for 11.1 clients. Older clients will continue to be able to use existing ViewSets that have not been edited by an 11.1 Integrity Lifecycle Manager client. Particular care should be taken when working with mandatory ViewSets in a mixed version environment. A mandatory ViewSet edited with a new Integrity Lifecycle Manager client may cause the older client to become unusable, due to new functionality introduced in newer releases.
- Viewsets of older clients may not display the options related to new functionality introduced in newer releases. To use new functionality in an older client with Integrity Lifecycle Manager 11.1 server, delete the existing viewset and reimport the viewset or create a new viewset.

- If some users who are working within a project will be using an older Integrity Lifecycle Manager client, ensure that you do not deactivate development paths in those projects. Older clients do not interact correctly with deactivated development paths.
- Ensure the Integrity Lifecycle Manager administration client and Integrity Lifecycle Manager server versions match. Administrative operations are not supported when using an Integrity Lifecycle Manager administration client that is a different version than the Integrity Lifecycle Manager server version.
- Integrity 10.9 (and later) includes the **Ignore Keywords** policy that prevents keywords from being expanded and unexpanded in text files worked on by users of configuration management functionality. For Integrity 10.8 (and earlier) clients, this policy works for **Member ▸ Check In** and **Member ▸ Rename** operations. For all other operations that have keyword settings, those keyword settings apply, even when the policy is set.

Configuration Management Repository Compatibility

Updates of earlier versions of the configuration management database repository to later versions are handled automatically by the Integrity Lifecycle Manager server installer, in the same manner that configuration management repositories are automatically updated.

Dedicated Integrity Lifecycle Manager Server Compatibility

You can have multiple Integrity Lifecycle Manager servers in your environment, each dedicated to a specific functionality. A common practice is to have one server configured for Workflows and Documents functionality and one or more servers configured for Configuration Management functionality.

The Workflows and Documents server is the central server and runs the current Integrity Lifecycle Manager version. Any number of Configuration Management servers can be connected to the central Workflows and Documents server. Connected Configuration Management servers can run any mix of Integrity Lifecycle Manager versions from 10.5 through the current version.

Integrity Lifecycle Manager Server and Integrity Lifecycle Manager Agent Compatibility

Integrity Lifecycle Manager server 11.1 supports connections from Integrity Agent 10.5 and Integrity 10.5 through 11.1. If you are upgrading from an earlier release to Integrity Lifecycle Manager Agent 11.1, you do not have to perform the upgrading all at once.

Integrity Lifecycle Manager API Compatibility

Integrity Lifecycle Manager provides an Application Program Interface (API) for integrating third-party products with Integrity Lifecycle Manager. The API provides a framework to invoke Integrity Lifecycle Manager commands and receive responses. The Integrity Lifecycle Manager API is versioned to ensure future changes to API commands and output do not affect existing integrations.

The following table identifies the supported compatibility configurations for API language bindings with Integrity Lifecycle Manager integration points:

API Language	API Version	Released in Integrity Lifecycle Manager Version
Integrity Lifecycle Manager Java/C API	4.16	11.1
	4.16	11.0
	4.16	10.9
	4.16	10.8
	4.15	10.7
	4.14	10.6
	4.13	10.5
	4.12	10.4
	4.11	10.0
	4.10	2009
Integrity Lifecycle Manager Web Services API	10.2	10.2
	10	10.0
	9.7	2009 SP7
	9.0	2009 SP2

Implementer and Integrity Lifecycle Manager Server Compatibility

Implementer integrates with Integrity Lifecycle Manager to provide functionality for Workflows and Documents, Configuration Management, or both.

Supported Databases

Supported Databases

You can find information about supported databases in Integrity Lifecycle Manager Product Platforms.

⚠ **Caution**

Ensure database upgrades are performed before upgrading the Integrity Lifecycle Manager server.

Database Migration

Database schema migrations are automatically handled during the installation process when you upgrade using the full install executable. During the installation, you must select your database type and database connection information. You are then prompted to migrate the database schema you used with the existing Integrity Lifecycle Manager for use in this release. For more information, see the *Integrity Lifecycle Manager Help Center*.

Note

- The installer does not permit you to create new tables if it detects any existing Integrity Lifecycle Manager tables. If there are existing data, your choices are to migrate the data or exit the installation.
- For Oracle databases, ORA-01555 errors can occur if you do not have sufficiently large rollback segments. Consult your Oracle product documentation for more information.
- The database migration eliminates duplicate ACLs in the following way:
  - If the ACL is a complete duplicate (same ACL name, same principal, same; permission name, and same grant/deny value), then the duplicate ACL is deleted.
  - If the ACL is a partial duplicate, such as conflicting values of grant/deny, then the ACL with the deny value wins.

If the migration fails, you must resolve the error, restore the database, and then attempt the migration again.

Note

If the migration fails due to a `Database Transaction Log Size Exceeded` error, you can correct the problem and then rerun the migration without first restoring the database.

Debugging and Diagnosing Database Issues

During the installation, the following file is created for debugging and diagnostic purposes when a database is upgraded:

`installdir``/log/dbinstall.log`

where *installdir* is the path to the directory where you installed the Integrity Lifecycle Manager server.

Integrations Support

Supported Integrations

For current information on supported integrations for Integrity Lifecycle Manager, see the Supported Integrations document.

Implementer Integration

When upgrading your Integrity Lifecycle Manager server, specific issues with Implementer integrations can occur. If you are upgrading an Implementer integration, contact PTC Technical Support for assistance, and consult the documentation for Implementer.

Getting Ready to Upgrade

When to Upgrade

PTC provides an alert system that publishes Integrity Lifecycle Manager Alerts for all new HotFixes as well as other pertinent product information (such as deprecated support for operating systems, databases, and releases). You can select the criteria of interest to you and sign up for e-mail notifications to alert you to the relevant updates. You can also review and search all existing alerts. For more information, go to the PTC Integrity eSupport portal at:

http://www.ptc.com/support/integrity.htm

To avoid disruption in service to your users, perform the upgrade in off hours. If that is not possible, remember to inform your users that you are performing an upgrade and that the Integrity Lifecycle Manager is to be temporarily unavailable.

Backing Up

Backing Up Your Database

You cannot rollback without an existing database backup. It is essential that you back up your existing database before starting the upgrade.

Performing regular backups of your database should already be part of your normal operations.

Note

Always stop the Integrity Lifecycle Manager server before performing any database maintenance.

Backing Up Integrity Lifecycle Manager Server Directory

You must back up the Integrity Lifecycle Manager server installation directory before running the installation. A backup of this directory allows you to restore the Integrity Lifecycle Manager server directory if the need arises.

Note

To ensure the database is not in use during a backup, stop the Integrity Lifecycle Manager server before creating any backups.

Performing a Trial Run

Because Integrity Lifecycle Manager is a mission-critical application for your enterprise, it is recommended that you perform a test of the upgrade in the form of a trial run before installing live on the production server.

Upgrading for UNIX Users

The upgrading instructions in the next sections apply to both Windows and UNIX users, except as follows for UNIX systems:

- Make sure the environment variable *$DISPLAY* is set, if necessary.
- References to “services” do not apply to UNIX.
- Install the Integrity Lifecycle Manager server in a new location without uninstalling the existing (previous) server.
- Ensure that the new server replacing the existing server has been installed, configured, tested, and performing as needed. Then uninstall the existing (previous) server by running the following file:

  ```
  prev_installdir/uninstall/IntegrityServerUninstall
  ```

  For more information on stopping the server, see the *Integrity Lifecycle Manager Help Center*.

  PTC recommends that you stop the Integrity Lifecycle Manager server before installing. This ensures that the previous Integrity Lifecycle Manager server is not running while upgrading your database.

Upgrading to Integrity Lifecycle Manager 11.1

If you are upgrading the Integrity Lifecycle Manager from versions older than 11.0, you must upgrade using the full install executable. If you are upgrading to Integrity Lifecycle Manager from 11.0, then you can upgrade using the full install executable or from a service pack file.

For more information, see the Prerequisites on page 131 section.

Single Server

The following is a high-level description for upgrading a single server:

1. Stop the existing Integrity Lifecycle Manager.
2. Install 11.1 to a different directory than the one in which the existing Integrity Lifecycle Manager server resides.
3. Migrate the server configuration files, such as properties, policies, reports, and triggers. Make manual adjustments to configuration files as required.
4. Upgrade Integrity Lifecycle Manager clients.
5. Uninstall the existing (previous) Integrity Lifecycle Manager server.

Multiple Servers

The following is a high-level description for upgrading multiple servers:

1. Plan your upgrade sequence, determining how to minimize downtime for each server. The recommended upgrade sequence follows:
   a. Workflows and Documents and Test Management-enabled servers
   b. Configuration Management-enabled servers
   c. FSA servers
   d. Clients
2. Implement your upgrade sequence. For each server:
   a. Stop the existing Integrity Lifecycle Manager server.
   b. Install Integrity Lifecycle Manager server 11.1 to a different directory than the one in which the existing Integrity Lifecycle Manager resides.
   c. Migrate server configuration files, such as properties, policies, reports, and triggers. Make manual adjustments to configuration files as required.
3. Upgrade Integrity Lifecycle Manager clients.
4. Uninstall all existing (previous) Integrity Lifecycle Manager servers.

Admin Staging Servers

As a best practice to avoid losing any in-progress changes while upgrading, PTC recommends a three-tiered admin staging configuration for upgrading staging and production servers.

Consider the following admin staging configuration:

Development (staging server) > *Test* (staging server) > *Production* (production server)

This configuration reduces the risk of losing any in progress changes on the Development and Test servers when the upgrade occurs.

1. Migrate all of the changes you want to retain from Development to Test and from Test to Production.
2. Ensure you have reliable backups of all of the servers in your admin staging configuration.
3. Ensure you have a rollback plan.
4. On the Development server:
   a. From the command line, run `mksis stop` to stop the server.
   b. To uninstall the Windows service for your server, run `mksis remove`. This removes the service only; it does not uninstall the program files.

c. Install the new server using the **Upgrade of an existing server** option. Specify the Development server's database as the database to upgrade, and point to the existing Integrity Lifecycle Manager server install directory as the server to upgrade. As part of this process, the Windows service is installed.

d. From the new *`Installdir`*`/bin`, run `isutil -c migrateServerConfig` *prev_installdir*.

   where *Installdir* is the new Integrity Lifecycle Manager server 11.1 directory and *prev_installdir* is the existing (previous) Integrity Lifecycle Manager server directory.

   This creates a very large amount of output, including files that were not migrated because of a conflict. This output appears in the `serverMigration.log` file, `migrateserverconfig.log` file, and the computer’s display.

   Note

   The documentation files pointed to in `documentationlist.properties` and `installationlist.properties` must be manually updated from the installation DVD and their locations manually updated in those files. For more information, see the *Integrity Lifecycle Manager Help Center*.

   Also, if you are upgrading using the full install executable, note that custom CA Root certificates stored in the `cacerts` keystore are not migrated during the upgrade. You must plan to import such certificates manually after the upgrade. If you do not migrate the certificates, the Integrity Lifecycle Manager server can fail to start due to an inability to establish the proper certificate trust.

e. Manually migrate the customized contents of any of the following *prev_Installdir* directories: `/data/public_html`, `/data/triggers`, `/data/gateway`.

5. On the Test server:

   a. From the command line, run `mksis stop` to stop your server.

   b. To uninstall the Windows service for your server, run `mksis remove`. This removes the service only; it does not uninstall the program files.

   c. Install the new server using the **Upgrade of an existing server** option. Specify the Test server's database as the database to upgrade, and point to the existing Integrity Lifecycle Manager server install directory as the server to upgrade. As part of this process, the Windows service is installed.

d. From `Installdir/bin`, run `isutil -c migrateServerConfig` *prev_installdir*.

   where *Installdir* is the new Integrity Lifecycle Manager server 11.1 directory and *prev_installdir* is the existing (previous) Integrity Lifecycle Manager server directory.

   This creates a very large amount of output, including files that were not migrated because of a conflict. This output appears in the `serverMigration.log` file, `migrateserverconfig.log` file, and the computer’s display.

> **Note**
>
> The documentation files pointed to in `documentationlist.properties` and `installationlist.properties` must be manually updated from the installation DVD and their locations manually updated in those files. For more information, see the *Integrity Lifecycle Manager Help Center*.
>
> Also, if you are upgrading using the full install executable, note that custom CA Root certificates stored in the `cacerts` keystore are not migrated during the upgrade. You must plan to import such certificates manually after the upgrade. If you do not migrate the certificates, the Integrity Lifecycle Manager server can fail to start due to an inability to establish the proper certificate trust.

e. Manually migrate the customized contents of any of the following *prev_Installdir* directories: `/data/public_html`, `/data/triggers`, `/data/gateway`.

6. On the Production server:

   a. From the command line, run `mksis stop` to stop your server.

   b. To uninstall the Windows service for your server, run `mksis remove`. This removes the service only; it does not uninstall the program files.

   c. Install the new server using the **Upgrade of an existing server** option. Specify this server's database as the database to upgrade, and point to the existing Integrity Lifecycle Manager server install directory as the server to upgrade. As part of this process, the Windows service is installed.

d. From the *`Installdir`*`/bin`, run `isutil -c migrateServerConfig` *prev_installdir*.

   where *Installdir* is the new Integrity Lifecycle Manager server 11.1 directory and *prev_installdir* is the existing (previous) Integrity Lifecycle Manager server directory.

   This creates a very large amount of output, including files that were not migrated because of a conflict. This output appears in the `serverMigration.log` file, `migrateserverconfig.log` file, and the computer’s display.

   Note

   The documentation files pointed to in `documentationlist.properties` and `installationlist.properties` must be manually updated from the installation DVD and their locations manually updated in those files. For more information, see the *Integrity Lifecycle Manager Help Center*.

   Also, if you are upgrading using the full install executable, note that custom CA Root certificates stored in the `cacerts` keystore are not migrated during the upgrade. You must plan to import such certificates manually after the upgrade. If you do not migrate the certificates, the Integrity Lifecycle Manager server can fail to start due to an inability to establish the proper certificate trust.

e. Manually migrate the customized contents of any of the following *prev_Installdir* directories: `/data/public_html`, `/data/triggers`, `/data/gateway`.

7. Restart the Production server.
8. Restart the Test server.
9. Restart the Development server.
10. To confirm that the Test server is functioning properly, start the Admin Migration Wizard on the Test server.
11. To confirm that the Development server is functioning properly, start the Admin Migration Wizard on the Development server.

Installing Integrity Lifecycle Manager 11.1 Server

This section contains instructions for installing the Integrity Lifecycle Manager server from the full install executable. If you are upgrading the Integrity Lifecycle Manager server using a service pack install, see the Install Notes documentation in the zip file.

Determining the Correct license.dat File

As you prepare to install Integrity Lifecycle Manager 11.1, you must determine the correct `license.dat` file to use.

If you are upgrading an existing server, you can continue to use the same `license.dat` file. On the existing server, check the *`prev_installdir`*`/config/properties/is.properties` file, and search for the `mksis.licensePath` property, which lists the path to the `license.dat` file. *prev_installdir* is the existing Integrity Lifecycle Manager server directory.

If you are moving the FlexNet server to new hardware, PTC recommends using the PTC Licensing Tool (https://www.ptc.com/apps/licenseManager/auth/ssl/index.jsp) to get a license transfer. This is not necessary if the Integrity Lifecycle Manager server is moving to new hardware, but the FlexNet server is not moving.

You can also use the PTC Licensing Tool to get a fresh copy of the license file.

Step 1: Stop Existing Integrity Lifecycle Manager Server

Before attempting to install Integrity Lifecycle Manager server 11.1, stop the existing Integrity Lifecycle Manager server. Always stop the server using the `mksis stop` command. Stopping the server using this method ensures that the service is also stopped; this is a requirement before installing a new Integrity Lifecycle Manager server.

⚠ **Caution**

Never perform a hard stop of the Integrity Lifecycle Manager server. Archives can become corrupted if a check-in operation is being performed when a hard stop is used to stop the Integrity Lifecycle Manager server.

For more information on stopping the existing Integrity Lifecycle Manager server, see the *Integrity Lifecycle Manager Help Center* for that release.

Note

This release installs the service named `Integrity Lifecycle Manager 11`. You need to uninstall the service from the previous Integrity Lifecycle Manager server release before installing Integrity Lifecycle Manager server 11.1. For information on installing and uninstalling the service, see the *Integrity Lifecycle Manager Help Center*.

Step 2: Install Integrity Lifecycle Manager 11.1 Server

As part of the installation and upgrade, you must also perform the following tasks:

1. Specify a suitable installation directory.
2. Select the language.
3. Specify the installation type.
    - If you are installing a new server, click **New server**.
    - If you are upgrading an existing server, click **Upgrade of an existing server**.
4. Specify the Server Type. The server type can be either a Full Server or an FSA (Proxy) Server.
5. Specify the appropriate server host name and port number for Integrity Lifecycle Manager server 11.1.
6. Upgrade your existing database.
7. Migrate the server configuration files and admin objects.
8. Identify changes to ACL Permissions in this release, and upgrade your ACLs accordingly.

For detailed information on configuring and starting the Integrity Lifecycle Manager server, or for information on administering the Integrity Lifecycle Manager server, see the *Integrity Lifecycle Manager Help Center*.

To upgrade an existing server

Note

If you select **Upgrade of an existing server**, the Item Presentation Templates (IPTs) from the existing Integrity server will be migrated to the newly installed Integrity server. If you select the **New Server**, the IPTs must manually be migrated from the existing Integrity server to the newly installed Integrity server. If you are using the Requirements Management 07 (RM 07) solution, then the built-in Integrity Lifecycle Manager fields will not be mapped to the existing IPTs in the solution and these IPTs will have to be upgraded manually.

1. Run the Integrity Lifecycle Manager installation.
2. When the installer prompts you for the type of installation, click **Upgrade of an existing server**, and then click **Next**.
3. Type the path or browse to the location of the existing server install directory, and then click **Next**.

   The installer copies any existing IPTs and stores them in the Integrity Lifecycle Manager server database.
4. Continue the server installation.

Installation Directory

If you are migrating server files, you must install the Integrity Lifecycle Manager server to a different directory than the previous installation.

Upgrade Your Database

During the installation, you must select your database type and database connection information. You are then prompted to migrate the database schema you used with the existing Integrity Lifecycle Manager server for use in Integrity Lifecycle Manager 11.1. For more information, see the *Integrity Lifecycle Manager Help Center*.

Note

- The server installer does not permit you to create new tables if it detects any existing Integrity Lifecycle Manager tables. If there are existing data, your choices are to migrate the data or exit the installation.
- For Oracle databases, ORA-01555 errors can occur if you do not have sufficiently large rollback segments. Consult your Oracle product documentation for more information.

If the migration fails, you must resolve the error, restore the database, and then attempt the migration again.

Note

If the migration fails due to a Database Transaction Log Size Exceeded error, you can correct the problem and then rerun the migration without first restoring the database.

Upgrade ACL Permissions

The Access Control Lists (ACLs) use a set of database tables to store security, configuration, and administrative information for the Integrity Lifecycle Manager.

Permissions added since the last release are set to deny. For more information, see the *Integrity Lifecycle Manager Help Center*.

Step 3: Migrate Server Configuration Files

In this release, certain properties are now contained in the database. If you have existing properties you want to retain, Integrity Lifecycle Manager provides the means to migrate server properties and configuration files automatically. However, the source files you are migrating must be located in the original directory structure.

If you intend to manually transfer server configuration file information instead of using the utility provided, see the section entitled “Manually Transferring Server Configuration Files”.

To migrate server settings and configuration files, use the `migrateServerConfig` command for the `isutil` utility.

Once the properties are successfully migrated to the database, they can be further modified in the Integrity Lifecycle Manager administration client GUI or from the CLI using the `integrity setproperty`, `im setproperty`, and `si setproperty` commands; and the `im diag` and `si diag` commands.

Migrating Using the `isutil` Utility

The `migrateServerConfig` command for the `isutil` utility migrates:

- properties

> **Note**
>
> The `migrateServerConfig` command converts security settings as part of the properties file migration.

- policies
- trigger scripts

  Script files that do not exist in the destination location are copied to the destination location and retain their original file names. All other script file names have an appropriate file extension appended and are copied to the destination location.

  - A file extension of `.10` is appended for Integrity versions 10.0-10.4.
  - A file extension of `.11` is appended for Integrity 10.5.
  - A file extension of `.13` is appended for Integrity 10.6.

  No merging of script contents is performed by the utility. You must manually modify script file contents.

  Similarly, the `global.events` file is copied and renamed.

  - It is renamed to `global.events.10` on Integrity versions 10.0-10.4.
  - It is renamed to `global.events.11` on Integrity 10.5.
  - It is renamed to `global.events.13` on Integrity 10.6.

  > **Note**
  >
  > If the scripts are located outside of the standard directory, the utility does not copy them. You must manually move them to the desired location.

> **Note**
>
> As of Integrity 10.6, Japanese versions of the trigger scripts are no longer installed with Integrity Lifecycle Manager server.

- reports and report resources

> **Note**
>
> In the event of an upgrade conflict (a report file was edited by the user and was also updated by PTC since the last release), the existing reports in the destination location (*installdir*/data/reports)are retained with their original filenames, and the new files have a number appended to their filenames. For Integrity 10.6 and later, the included report files support localization (see the documentation for localizing reports in the *Integrity Lifecycle Manager Help Center*). If you are using included reports from an Integrity release 10.5 and earlier, you can continue to use those reports if there is no localization requirement. To make use of the localization features of reports, manually update the reports as needed.
>
> After you run this utility to upgrade to Integrity 10.6 and later, the *installdir*/data/reports/ja folder will be present on the upgraded server. However, Integrity Lifecycle Manager no longer supports language-based folders, so you can delete this folder. Manually copy your pre-10.6 report recipes (English and non-English) to *installdir*/data/reports/recipes.

- Java properties from *prev_installdir*/config/mksservice.conf:
  - Memory (only if they are larger in the previous release than the latest release): -Xms, and -Xmx, -Xss
  - Garbage collection: (if they do not currently exist): -XX:+PrintGCTimeStamps, -XX:+PrintTenuringDistribution, and -XX:+PrintGCDetails
- change package reviewer rules
- sitenotes.html (if the file does not exist in the target directory)
- information about shared Visual Studio solutions and projects in *prev_installdir*/data/vsi/vsibinding.properties

Note

As of Integrity 10.6, this utility no longer migrates files that are not part of the Integrity Lifecycle Manager version to which you are updating. If you wish to use existing files on your system that were dropped or deprecated, you must manually copy them from the source server (the one you are migrating from) to destination server (the one you are migrating to).

The `isutil` utility is installed with Integrity Lifecycle Manager 11.1 in the following location:

```
installdir/bin/isutil.exe
```

where *installdir* is the installation directory for Integrity Lifecycle Manager server 11.1.

The syntax for the command is:

```
isutil –c migrateServerConfig "prev_installdir"
```

where *prev_installdir* is the installation directory for the source Integrity Lifecycle Manager server you are migrating from.

Caution

There is no undo mechanism in place to undo a migration. To restore Integrity Lifecycle Manager server 11.1 files, see the section entitled "Migration Backup Files".

For the purposes of this procedure, only the `migrateServerConfig` command is documented and intended to be used. For more information on the available commands for the `isutil` utility, see the Using `isutil` to Manage the Database Repository topic in the *Integrity Lifecycle Manager Help Center*.

Upon completion, the command outputs a message to the console describing the success or failure of the migration. If the command fails, details are printed to the console only, requiring you to analyze and save the output. If there are script files, you need to manually merge them at this time.

Note

Passwords in migrated files are never displayed in messages to the console or the log file. Regardless of the actual length of the passwords, they are replaced with the "XXXX" string.

Migration Backup Files

The migrateServerConfig command for the isutil utility creates a backup of every file in the Integrity Lifecycle Manager server 11.1 installation directory that it changes. Files are backed up into the following location:

```
installdir/backup
```

where *installdir* is the Integrity Lifecycle Manager server 11.1 installation directory.

The backup directory contents mimic the target server directory structure. The backup root directories are versioned such that rerunning the migration never overwrites existing backup files. For example, running the migration a second time creates an additional folder backup1, and running a third time, backup2.

Migration Log

The migrateServerConfig command for the isutil utility generates the following log:

```
installdir/log/serverMigration.log
```

where *installdir* is the Integrity server installation directory.

The serverMigration.log file contains:

- The date and time the migration occurred.
- The absolute path to the source directory used, for example, the installation directory for Integrity server.
- The name of each file changed by the utility, and the name of the backup file created to represent the original (see the section entitled “Migration Backup Files”).
- A description of how the file was changed, stating:
  - if the file was deleted
  - if the file is an exact copy from the old installation (including the name of the source file copied, and its new name)
  - what settings were changed, moved or removed (including the setting name, its original value, and its new value)

Migrating Agent Properties

The Integrity Lifecycle Manager Agent AgentUtils.exe utility migrates Integrity Lifecycle Manager Agent properties. This utility is run on each agent and must be pointed at the previous installed version of the Integrity Lifecycle Manager Agent.

The utility also migrates the following Java properties from *prev_Installdir*/config/mksservice.conf:

- Memory (only if they are larger in the previous release than in the latest release): `-Xms`, `-Xmx`, `- Xss`
- Garbage collection: (if they do not currently exist): `-XX:+PrintGCTimeStamps`, `-XX:+PrintTenuringDistribution`, and `-XX:+PrintGCDetails`
- The Integrity Lifecycle Manager Agent 11.1 target directory must be located in a different location than the original Integrity installation location.

The utility is installed with Integrity Lifecycle Manager 11.1 in the following location:

```
installdir/bin/AgentUtils.exe
```

where *installdir* is the directory Integrity Lifecycle Manager Agent is migrated to.

The syntax for the command is:

```
agentutil –c migrateAgentConfig “prev_installdir”
```

where *Integrity Agent installdir* is the current Integrity Lifecycle Manager Agent target directory.

On completion, the command outputs a message to the console describing the success or failure of the migration. If the command fails, details are printed to the console only, requiring you to analyze and save the output.

Note

Passwords in migrated files are never displayed in messages to the console or the log file. Regardless of the actual length of the passwords, they are replaced with the “XXXX” string.

Step 4: Start Integrity Lifecycle Manager 11.1 Server

To start the new Integrity Lifecycle Manager server, use the `mksis start` command.

For complete details on running the Integrity Lifecycle Manager server, see the server installation documentation in the *Integrity Lifecycle Manager Help Center*.

Ensure that users can connect to the Integrity Lifecycle Manager server by checking the server log file (and verifying an entry for `GENERAL(0): Listening on port *:`*nnnn*) or by connecting through an Integrity Lifecycle Manager client.

Step 5: Uninstall Existing Integrity Lifecycle Manager Server

Note

Before uninstalling the existing Integrity Lifecycle Manager server, ensure that the Integrity Lifecycle Manager server replacing it has been installed, configured, tested, and performing as required.

For more information on uninstalling the existing Integrity Lifecycle Manager server (the previous installation you are upgrading from), refer to the documentation that was originally issued for the release version you installed.

When uninstalling Integrity Lifecycle Manager server on Windows, you should always use the following file:

`installdir``/uninstall/IntegrityServerUninstall.exe`

When uninstalling Integrity Lifecycle Manager on Linux or Solaris, you should always use the following file:

`installdir``/uninstall/IntegrityServerUninstall`

After the server is stopped, the `IntegrityServerUninstall.exe` file removes the Integrity Lifecycle Manager service and launches the application that performs the server uninstall.

On Windows, if the server is running as a service, make note of the user the service is running as before uninstalling. In most cases this is “Local System”. If you use the operating system’s uninstall feature or the uninstall shortcut in the Integrity Lifecycle Manager server program group, you must manually remove the service. This step is not necessary if the Integrity Lifecycle Manager server is installed on Linux or Solaris operating system.

6

Command Line Interface Changes

This release includes changes that affect the command line interface (CLI).

Note

As part of any upgrade, PTC recommends that you review and update any scripts you use with the CLI.

This document summarizes the key changes for the command line interfaces that have changes in this release of Integrity Lifecycle Manager.

Command Line Information

For detailed information on all commands and options, see the CLI man pages.

im Command Options

im Command	Changes
`im editfield`	New command option: `--[no|confirm]changeStoreToHistoryFrequencyFromNever`

integrity Command Options

There are no changes to `integrity` commands for this release.

si Command Options

si Command	Changes
`si addprojectmergeline`	New command
`si deleteprojectmergeline`	New command
`si exportsandbox`	New command
`si memberpermissions`	New command
`si projectpermissions`	New command
`si viewdroppedmembers`	New command

`si importsandbox`	New command options:
	`--importFiles=[none|members|all]`
	`--[no|confirm]overwriteExistingFile`
	`--targetSandboxDir`
	Deprecated command options:
	`-S`
	`--[no]failOnAmbiguousProject`
	`--sandbox`

`si viewsandbox`	New value for `--fields`: `deferredOperationInformation`

tm Command Options

There are no changes to `tm` commands for this release.

ACL Permissions

There are no changes to the Access Control Lists (ACLs) for this release.