InterSystems™
IRIS Data Platform

Encryption Guide

Version 2023.1
2024-04-15

InterSystems Corporation   1 Memorial Drive   Cambridge MA 02142   www.intersystems.com

Encryption Guide
InterSystems IRIS Data Platform Version 2023.1 2024-04-15

For Support questions about any InterSystems products, contact:

InterSystems Worldwide Response Center (WRC)
Tel: +1-617-621-0700
Tel: +44 (0) 844 854 2917
Email: support@InterSystems.com

Table of Contents

1

About Managed Key Encryption

InterSystems IRIS® data platform includes support for managed key encryption, a suite of technologies that protects data at rest. These technologies are:

- *Block-level database encryption*, also known simply as *database encryption* — A set of administrative tools to allow creation and management of databases in which all the data is encrypted. Such databases are managed through the Management Portal.
- *Data-element encryption for applications*, also known simply as *data-element encryption* — A programmatic interface that allows applications to include code for encrypting and decrypting individual data elements (such as particular class properties) as they are stored to and retrieved from disk.
- *Encryption key management*, also known simply as *key management* — A set of tools for creating and managing the keys that are used to encrypt either databases or data elements.

Keys for encrypting either databases or data elements are known as *data-encryption keys* and may also be known simply as *keys* (when the context is clear). Each instance can simultaneously have up to 256 data-encryption keys activated for database encryption and up to four data-encryption keys activated for data-element encryption; *activating* a key makes it available for encryption and decryption operations.

Encryption keys can be stored in two ways:

- On standard machines in key files
- On dedicated hardware on which keys are accessible via the Key Management Interoperability Protocol (KMIP)

Note: You can simultaneously use a key in a key file for database encryption and data-element encryption.

InterSystems IRIS uses AES (the Advanced Encryption Standard) to perform its encryption and decryption when an instance writes to or reads from disk. For databases, InterSystems IRIS writes and reads in fixed-length blocks, and the entire database is encrypted, except for the single label block; this encrypted content includes the data itself, indexes, bitmaps, pointers, allocation maps, and incremental backup maps. For data elements, only the specified data is encrypted, and a unique identifier for the encryption key is included with the encrypted data on disk.

Encryption and decryption have been optimized, and their effects are both deterministic and small for any InterSystems IRIS platform. For information about how InterSystems IRIS database encryption affects facilities related to but separate from databases, see Encryption and Database-Related Facilities.

2

Key Management Tasks

A *key*, short for *data-encryption key*, is a 128–, 196–, or 256–bit bit string that is used with a cryptographic algorithm to reversibly encrypt or decrypt data. Each key has a unique identifier (known as a *GUID*), which InterSystems IRIS® data platform displays as part of key management activities.

Key management is the set of activities associated with creating keys, activating keys, deactivating keys, assigning default keys for various activities, and deleting keys. It also includes management activities associated with key storage. You can store keys in either of two ways:

- In key files — A *key file* is a file that holds encrypted copies of up to 256 keys. A key file administrator provides a password in order to decrypt the key for use.
- On a KMIP server — A *KMIP server* is a key management server that can send and receive communications using the key management interoperability protocol (KMIP). KMIP servers are available from various third-party vendors; those vendors provide instructions for configuring and using the KMIP server generally.

Note: If you wish to configure encryption for journal files or the IRISTEMP and IRISLOCALDATA databases, this is part of InterSystems IRIS startup configuration. See Configure Encryption Startup Settings for details.

2.1 Manage Keys in Key Files

A *key file* is a file that holds encrypted copies of one or more data-encryption keys. Key file management is the set of activities associated with key files themselves, such as adding administrators to or removing administrators from key files. Within a particular key file, all administrators have access to all keys. All keys are stored in an encrypted form, along with administrator information; each data-encryption key is individually encrypted using a *master key*. For each administrator in the key file, there is a unique, encrypted copy of the master key, which is encrypted using a *key-encryption key* — where each key-encryption key is derived from an individual key administrator's password. Encryption tasks require an activated data-encryption key, and InterSystems IRIS requires an administrator username and password to decrypt that key so that it can then be activated.

Working with key files involves the following tasks:

- Create a Key File
- Add a Key to a Key File, or Delete a Key from a Key File
- Add an Administrator to a Key File, or Delete an Administrator from a Key File
- Activate a Database Encryption Key from a Key File, or Deactivate a Database Encryption Key

- Activate a Data-Element Encryption Key from a Key File, or Deactivate a Data-Element Encryption Key
- Manage Keys and Key Files with Multiple-Instance Technologies
- Specify the Default Encryption Key or Journaling Encryption Key for an Instance

Note: If an instance uses multiple keys at startup time (such as with journal files, the audit database, and other databases), then those keys must all be in a single key file. This allows them all to be available when the instance starts.

2.1.1 Create a Key File

When you create an encryption key file, it contains one key. To create an encryption key file and its initial key:

1. From the Management Portal home page, go to the **Create Encryption Key File** page (**System Administration** > **Encryption** > **Create New Encryption Key File**).
2. On the **Create Encryption Key File** page, specify the following values:
   - **Key File** — The name of the file where the encryption key is stored; this can be an absolute or relative path name.

     If you enter an absolute file name, the key file is placed in the specified directory on the specified drive; if you enter a relative file name, the key file is placed in the manager's directory for the InterSystems IRIS instance (which is below the InterSystems IRIS installation directory — that is, in <install-dir>/mgr/). Also, no file suffix is appended to the file name, so that the file MyKey is saved simply with that file name. You can also use the **Browse** button to the right of this field to choose the directory where InterSystems IRIS will create the key file. (If you provide the name of an existing file, InterSystems IRIS will not overwrite it and the save will fail.)

     WARNING! Any key stored in <install-dir>/Mgr/Temp is deleted when InterSystems IRIS next reboots — *never* store a key in <install-dir>/Mgr/Temp.

   - **Administrator Name** — The name of an administrator who can activate the key. There must be at least one administrator.

     Because the database encryption functionality exists independent of InterSystems IRIS security, this name need not match any user names that are part of InterSystems IRIS security. By default, the initial administrator name value is the current username. The administrator name cannot include Unicode characters.

   - **Password** — A password for this user.

     Because the database encryption functionality exists independent of InterSystems security, this password need not match the password that a user has for InterSystems IRIS security. Note that this password is not stored anywhere on disk; it is the responsibility of the administrator to ensure that this information is not lost.

     InterSystems suggests that this password follow the administrator password strength guidelines. If someone can successfully guess a valid password, the password policy is too weak. Also, this password cannot include Unicode characters.

     Important: The key administrator's password is not stored anywhere on disk. It is the responsibility of the key administrator to ensure that this information is not lost.

   - **Confirm Password** — The password for this user entered again to confirm its value.
   - **Cipher Security Level** — The length of the key, where choices are 128–bit, 192–bit, and 256–bit.
   - **Key Description** — Text that describes the key that is initially created and stored in the key file. This text appears in the **Description** column of the **Encryption Keys Defined in Key File** table.
3. Click **Save** at the top of the page to save the key file to disk.

4. Having just created a key, follow the instructions in Protection from Accidental Loss of Access to Encrypted Data to create and store a backup copy of the newly updated key file.

This creates a key file with a single database encryption key in it and with a single administrator. The page displays ID for the key, which is a string such as `9158980E-AE52-4E12-82FD-AA5A2909D029`. The key ID is a unique identifier for the key which InterSystems IRIS displays here and on other pages. It provides a means for you to keep track of the key, regardless of its location. This is important because, once you save the key file, you can move it anywhere you choose; this means that InterSystems IRIS cannot track it by its location.

The key is encrypted using the master encryption key, and there is a single copy of master encryption key, which is encrypted using the administrator's key-encryption key (KEK). You can add additional keys to the key file according to the instructions in Add a Key to a Key File. You can add administrators to the key file according to the instructions in Add an Administrator to a Key File.

WARNING! InterSystems strongly recommends that you create and store a backup copy of the key file. Each time you create a database encryption key, it is a unique key that cannot be re-created. Using the same administrator and password for a new key still results in the creation of a different and unique key. If an unactivated key is lost and cannot be recovered, the encrypted database that it protected will be unreadable and its data will be *permanently lost*.

2.1.2 Add a Key to a Key File

When using key files, there are two different ways to create a key:

- Create a key file. This causes InterSystems IRIS to create a key and place it in the file. To create a key file, see Create a Key File.
- Add a key to an existing key file, as described in this section.

To add a key to an existing key file:

1. From the Management Portal home page, go to the **Manage Encryption Key File** page (**System Administration** > **Encryption** > **Manage Encryption Key File**).
2. On the **Manage Encryption Key File** page, in the **Key File** field, enter the name of the key file to which you want to add and store the key; click **OK**. This displays information about that key file; at the bottom of the shaded area, the **Encryption Keys Defined in Key File** table displays a list of the one to 256 keys in the key file. If there are three or fewer keys in the file, you can create a new key and add it to the file.
3. Click the **Add** button below the **Encryption Keys Defined in Key File** table to add a key to the key file. This displays the **Add a New Encryption Key** screen.
4. In the **Add a New Encryption Key** screen, enter values in the following fields:
   - **Existing Administrator Name** — The name of an administrator associated with the key file. (Administrators associated with the file appear in the **Administrators Defined in Key File** table on the **Manage Encryption Key File** page.)
   - **Existing Administrator Password** — This administrator's password.
   - **Description** — Text to describe the key. This text appears in the **Description** column of the **Encryption Keys Defined in Key File** table.
5. Click **OK** to save the key to the key file. This displays information about it in the **Encryption Keys Defined in Key File** table, including its ID, which is a string such as `9158980E-AE52-4E12-82FD-AA5A2909D029`. (The key ID is a unique identifier for the key which InterSystems IRIS displays here and on other pages. It provides a means for you to keep track of the key, regardless of its location. This is important because, once you save the key file, you can move it anywhere you choose; this means that InterSystems IRIS cannot track it by its location.)

6. Having just added a new key to the key file, follow the instructions in Protection from Accidental Loss of Access to Encrypted Data to create and store a backup copy of the newly updated key file.

WARNING! InterSystems strongly recommends that you create and store a backup copy of the key file. Each time you create a database encryption key, it is a unique key that cannot be re-created. Using the same administrator and password for a new key still results in the creation of a different and unique key. If an unactivated key is lost and cannot be recovered, the encrypted database that it protected will be unreadable and its data will be *permanently lost*.

2.1.3 Delete a Key from a Key File

To delete a key from a key file:

1. From the Management Portal home page, go to the **Manage Encryption Key File** page (**System Administration** > **Encryption** > **Manage Encryption Key File**).
2. On the **Manage Encryption Key File** page, in the **Key File** field, enter the name of the key file from which you want to delete the key; click **OK**. This displays information about that key file; at the bottom of the shaded area, the **Encryption Keys Defined in Key File** table displays a list of the one to 256 keys in the key file. If there are two or more keys in the file, you can delete a key from the file.
3. In the table of keys, click **Delete** in the row for a key to delete that key. Clicking **Delete** displays a confirmation page for the action.

   If the key's **Delete** button is not available, this is because the key is the default encryption key or the journal encryption key for the file. To delete the key, first specify that another key is the default encryption key or the journal encryption key for the file by clicking **Set Default** or **Set Journal** for the other key.
4. Click **OK** on the confirmation dialog to delete the key from the file.

WARNING! Before deleting the only existing copy of a key, it is critical that you are absolutely sure that there is no existing encrypted content that uses it. If there is no copy of the key that is required to decrypt data, the encrypted data that it protected will be unreadable and *permanently lost*.

2.1.4 Add an Administrator to a Key File

To add an administrator to an existing key file:

1. From the Management Portal home page, go to the **Manage Encryption Key File** page (**System Administration** > **Encryption** > **Manage Encryption Key File**).
2. In the **Key File** field, enter the path and filename of the key file to open and click **OK**; you can also use the **Browse** button to look for the key. Once the Portal opens the key file, it displays a table with the administrators listed in the file; administrator names appear in all capital letters, regardless of how they were defined.
3. In the table of administrators, click **Add** to add a new administrator. This displays a page with the following fields:
   - **Existing Administrator Name** — The name of an administrator already in the file.
   - **Existing Administrator Password** — The password associated with the already existing administrator in the file.
   - **New Administrator Name** — The name of the new administrator to be added to the file. Because the encryption functionality is independent of InterSystems IRIS security, the administrator name need not match any user names that are part of InterSystems IRIS security. This user name cannot include Unicode characters.
   - **New Administrator Password** — The password for the new administrator. InterSystems suggests that this password follow the administrator password strength guidelines; also, this password cannot include Unicode characters.

Because the encryption functionality is independent of InterSystems IRIS security, the password need not match the password that a user has for InterSystems IRIS security.

- **Confirm New Administrator Password** — Confirmation of the password for the new administrator.

Complete these fields and click **OK**. You have now added a new administrator to the key file.

Once you have added the new administrator to the key file, you may wish to copy the key file, making sure that each copy is in a secure location. Further, InterSystems strongly recommends that you create multiple administrators for each key, one of which has the name and password written down and stored in a secure location, such as in a fireproof safe. However, if copies of the key file are made and later on, as an administrative function, a new administrator is added, only the copy of the key file with the new administrator will be up to date.

Note: When you add a new administrator to a key file, that administrator's password is permanently associated with the entry for the administrator name created in the file. Once assigned, passwords cannot be changed. If you wish to assign a new password, delete the entry in the key file for that administrator name and then create a new entry with the same name and a new password.

2.1.5 Delete an Administrator from a Key File

To delete an administrator from a key file:

1. From the Management Portal home page, go to the **Manage Encryption Key File** page (**System Administration** > **Encryption** > **Manage Encryption Key File**).
2. In the **Key File** field, enter the path and filename of the key and click **OK**. This displays a table with the administrators listed in the file (as well as a table of encryption keys in the file).
3. In the table of administrators, click **Delete** next to an administrator to remove that administrator for the key. Clicking **Delete** displays a confirmation page for the action. (If there is only one administrator in the file, there is no **Delete** button, as it is not permitted to delete this administrator.)
4. Click **OK** to delete the administrator from the file.

2.1.6 Activate a Database Encryption Key from a Key File

InterSystems IRIS supports up to 256 simultaneously activated keys for database encryption. To activate a key from a key file for database encryption:

1. From the Management Portal home page, go to the **Database Encryption** page (**System Administration** > **Encryption** > **Database Encryption**). If there are already any activated keys, the page displays a table listing these.
2. On this page, click **Activate Key**, which displays the fields for activating a key.
3. Enter values for the following fields:
    - **Key File** — The name of the file where the encryption key is stored. If you enter an absolute file name, InterSystems IRIS looks for the key file in the specified directory on the specified drive; if you enter a relative file name, InterSystems IRIS looks for the key file starting in the manager's directory for the InterSystems IRIS instance (which is below the InterSystems IRIS installation directory — that is, in `<install-dir>/mgr/`). You can also use the **Browse** button to display a dialog for opening the key file.
    - **Administrator Name** — The name of an administrator for this key, specified either when the key was created or edited.
    - **Password** — The password specified for the named administrator.

4. Click the **Activate** button.

InterSystems IRIS then attempts to activate all the keys in the specified file. If there are not enough slots to activate all the keys in the file, then InterSystems IRIS opens as many keys as it can.

After key activation, the **Database Encryption** page displays the table of activated keys. For each key that InterSystems IRIS activates, the page adds the key to table of activated keys and displays the key's identifier. For each activated key, you can also perform various actions:

- **Set Default** — Click to specify that InterSystems IRIS uses this key when creating new encrypted databases. For more details, see Specify the Default Encryption Key or Journaling Encryption Key for an Instance.
- **Set Journal** — Click to specify that InterSystems IRIS uses this key to encrypt journal files. For more details, see Specify the Default Encryption Key or Journaling Encryption Key for an Instance.
- **Deactivate** — Click to deactivate this key. For more details, see Deactivate a Database Encryption Key

Note: The table of keys does not display any file or path information. This is because, once a key file is created, any sufficiently privileged operating system user can move it; hence, InterSystems IRIS may not have accurate information about the operating system location and can only rely on the accuracy of the GUID for the activated key in memory. When activating a second or subsequent key, note the identifier(s) for the currently activated key(s) first, so that you can identify the new one.

2.1.7 Activate a Data-Element Encryption Key from a Key File

InterSystems IRIS supports up to four activated keys at one time for data-element encryption. To activate a key for data-element encryption:

1. From the Management Portal home page, go to the **Data Element Encryption** page (**System Administration** > **Encryption** > **Data Element Encryption**). If there are already any activated keys, the page displays a table listing these.
2. On the **Data Element Encryption** page, select **Activate Key**, which displays the fields for activating a key. If key activation is not available, this is because there are already four activated data element keys.
3. Enter values for the following fields:
    - **Key File** — The name of the file where the encryption key is stored. If you enter an absolute file name, InterSystems IRIS looks for the key file in the specified directory on the specified drive; if you enter a relative file name, InterSystems IRIS looks for the key file starting in the manager's directory for the InterSystems IRIS instance (which is below the InterSystems IRIS installation directory — that is, in `<install-dir>/mgr/`).
    - **Administrator Name** — The name of an administrator for this key, specified either when the key was created or edited.
    - **Password** — The password specified for the named administrator.
4. Click the **Activate** button.

InterSystems IRIS then attempts to activate all the keys in the specified file. If there are not enough slots to activate all the keys in the file, then InterSystems IRIS opens as many keys as it can.

After key activation, the **Data Element Encryption** page displays the table of activated keys. For each key that InterSystems IRIS activates, the page adds the key to table of activated keys and displays the key's identifier.

Note: The table of keys does not display any file or path information. This is because, once the key file is activated, any sufficiently privileged operating system user can move the key; hence, InterSystems IRIS may not have accurate information about the operating system location and can only rely on the accuracy of the GUID for the activated key in memory. When activating a second or subsequent key, note the identifier(s) for the currently activated key(s) first, so that you can identify the new one.

2.1.8 Manage Keys and Key Files with Multiple-Instance Technologies

If you are using encrypted databases or journal files within an InterSystems IRIS cluster, the InterSystems IRIS instances on all nodes in the cluster must share a single database encryption key.

Before enabling journal file encryption for any instance that belongs to an InterSystems IRIS mirror, see Activating Journal Encryption in a Mirror for important information. (There are no mirroring-related requirements in regard to database encryption.)

There are two ways to enable sharing of a single key:

- All of the instances share a single key file, which is located on one cluster node or mirror member.

  In this case, if you change the single copy of the key file, then these changes are visible to all nodes or members. However, if the host holding the key file becomes unavailable to the other nodes or members, any attempt to read the key from the key file fails; this can prevent InterSystems IRIS instances from restarting properly.

- Each cluster node or mirror member has its own copy of the key file.

  Here, if you change the key file, then you propagate copies of the key file (containing the same key) to all the other nodes or members. This increases the burden of administering the key file (which is typically small), but ensures that each instance of InterSystems IRIS always has a key available at startup.

Important: Whether there are single or multiple key files, the database encryption key itself is the same for all instances.

2.1.8.1 Using a Single Key File

To use a single key file:

1. Create a database encryption key on one node or member. For more information on this procedure, see Create a Key File.
2. Secure this key according to the instructions in Protection from Accidental Loss of Access to Encrypted Data.

   CAUTION: Failure to take these precautions can result in a situation in which the encrypted databases or journal files are unreadable and *permanently lost*.

3. Configure each instance of InterSystems IRIS for unattended startup and provide InterSystems IRIS with the path to the key file. For more information on this procedure, see Startup with Unattended Key Activation.

Since all the InterSystems IRIS instances use the same key, they are able to read data encrypted by each other. Any changes to the key file are visible to all instances.

2.1.8.2 Using Multiple Key Files

To use multiple copies of a key file:

1. Create a database encryption key on one node or member. For more information on this procedure, see Create a Key File.

2. Secure this key according to the instructions in Protection from Accidental Loss of Access to Encrypted Data.

   CAUTION: Failure to take these precautions can result in a situation in which the encrypted databases or journal files are unreadable and *permanently lost*.

3. Make a copy of the key file for each of the remaining nodes or members.
4. On each node or member:
   a. Get a copy of the key file and put it in a secure and stable location on that machine.
   b. Configure each instance of InterSystems IRIS for unattended startup. For more information on this procedure, see Startup with Unattended Key Activation.

Since each copy of the key file contains the same key, all the InterSystems IRIS instances are able to read data encrypted by each other. Since each InterSystems IRIS instance has a key file on its machine, the key file should always be available for an InterSystems IRIS restart. If there are any changes to the key file (such as adding or removing administrators), you must propagate new copies of the key file to each machine and reconfigure each instance of InterSystems IRIS for unattended startup using the new copy of the key file (even if that file is in the same location as the old file).

2.2 Managing Keys with the Key Management Interoperability Protocol (KMIP)

InterSystems supports the use of a KMIP server to manage database encryption keys. Using KMIP includes the following tasks:

- Create, edit, or delete a KMIP Server Configuration
- List the KMIP Server Configurations
- List Details about a KMIP Server Configuration
- Create a Key on the KMIP Server, or Delete a Key on the KMIP Server
- List the Keys on the KMIP Server
- Activate a Database Encryption Key from a KMIP Server, or Deactivate a Database Encryption Key
- Activate a Data-Element Encryption Key from a KMIP Server, or Deactivate a Data-Element Encryption Key
- Copy a Key from a KMIP Server to a Key File
- Specify the Default Encryption Key or Journaling Encryption Key for an Instance

Note:
- InterSystems IRIS supports KMIP protocol versions 1.0–2.1.
- KMIP activities are not supported on macOS instances of InterSystems IRIS.

2.2.1 Create a KMIP Server Configuration

When establishing a connection between InterSystems IRIS and a KMIP server, you create a *KMIP server configuration*, which defines properties of the KMIP server and represents it within the InterSystems IRIS instance. To create a KMIP server configuration:

1. Set up the KMIP server according to its vendor's instructions.

   CAUTION: When configuring a KMIP server, follow all proper backup procedures according to your vendor's instructions. If you do not have backup copies of your keys, you may lose data *permanently*.

   Once you have set up the server, you can then set up the KMIP server configuration in InterSystems IRIS:

2. To set up a KMIP server configuration, you must have:

   - The certificate authority (CA) certificate for the KMIP server, which must a trusted CA. You should receive this certificate from the vendor that provides the KMIP server or should obtain it according to instructions from that vendor.
   - A public-key certificate and private key for each instance of InterSystems IRIS that will communicate with the KMIP server. The certificate must be issued by a trusted CA. You should receive this certificate and private key from the vendor that provides the KMIP server or should obtain them according to instructions from that vendor.
   - The following information about the KMIP server:
     - Its fully-qualified DNS name or IP address
     - The port number on which it accepts connections
     - The version of the KMIP protocol that it supports
     - Any TLS settings that it requires for its clients

3. On the InterSystems IRIS instance that will communicate with the KMIP server, create an TLS configuration that will represent the instance to the KMIP server:

   a. In the Portal, go to the **SSL/TLS Configurations** page (**Home** > **System Administration** > **Security** > **SSL/TLS Configurations**).

   b. On the **SSL/TLS Configurations** page, click the **Create New Configuration** button, which displays the **New SSL/TLS Configuration** page.

   c. On the **New SSL/TLS Configuration** page, set up the TLS configuration. For the fields listed below, specify or select values as follows

      - **Enabled** — Select this check box.
      - **Type** — Select **Client**.

      The values for other fields (**Server certificate verification**, the **This client's credentials** fields, and the **Cryptographic settings** fields) depend on the requirements of the KMIP server. The values for the **This client's credentials** fields depend on the client certificate, client private key, and CA certificate that you have received from the vendor that provides the KMIP server.

      For more information on this creating an TLS configuration, see Create or Edit a TLS Configuration.

4. Create the configuration to the KMIP server:

   a. Start the Terminal and log in as a sufficiently privileged user.

   b. At the terminal prompt, go to the %SYS namespace:

   ```
   >set $namespace="%SYS"
   ```

   c. Run **^SECURITY**

   ```
   %SYS>do ^SECURITY
   ```

d. In **^SECURITY**, select option **14**, **KMIP server setup**.

e. In the **KMIP server setup** choices, select option **1**, **Create KMIP server**.

f. At the **Create KMIP server** prompts, specify values for the following:

   - **KMIP server to create?** — The name of the KMIP server configuration.
   - **Description?** — A text description.
   - **Server host DNS name?** — The fully-qualified DNS name or IP address of the KMIP server.
   - **TCP port number?** — The port number on which the KMIP server accepts connections.
   - **OASIS KMIP protocol version?** — The number associated with your KMIP server's supported version of the protocol. This is part of the information that you have received from the vendor that provides the KMIP server.
   - **SSL/TLS Configuration name?** — The name of the TLS configuration that you created in the previous step.

     Note: This case of the value that you enter here must match that of the TLS configuration name as defined.

   - **Non-blocking I/O?** — Whether or not connections to the KMIP server enable non-blocking I/O. InterSystems recommends **Yes**, which enables non-blocking I/O.

     If non-blocking I/O is enabled, control returns to the application after the timeout specified at the **I/O timeout, in seconds?** prompt (below). If non-blocking I/O is disabled, control returns to the application after an operating-system timeout (which may not occur).

   - **Auto-reconnect?** — Whether or not InterSystems IRIS reconnects with the KMIP server if the connection drops. InterSystems recommends that you select **No**; there is then no attempt to automatically reconnect if the connection drops.
   - **I/O timeout, in seconds?** — The amount of time, in seconds, before a timeout occurs in the connection to the KMIP server. This is only relevant if the configuration has enabled non-blocking I/O.
   - **Log KMIP messages?** — Whether or not InterSystems IRIS logs messages that it sends to the KMIP server. If messages are logged, InterSystems IRIS stores them in the <install-dir>/mgr/kmipcmd.log file.
   - **Debug SSL/TLS?** — Whether or not InterSystems IRIS logs TLS debugging information. If information is logged, InterSystems IRIS stores it in the <install-dir>/mgr/kmipssl.log file.

g. After the prompts for KMIP server properties, confirm that you wish to create the KMIP server at the **Confirm creation of KMIP server** prompt.

Note: InterSystems supports the use of multiple KMIP servers and the use of a single KMIP server that has multiple configurations. The most recently activated configuration is the default.

2.2.2 Edit a KMIP Server Configuration

To modify the values of the properties of an existing KMIP server configuration:

1. For the relevant instance, start the Terminal and log in as a sufficiently privileged user.
2. At the terminal prompt, go to the %SYS namespace:

   ```
   >set $namespace="%SYS"
   ```

3. Run **^SECURITY**:

   ```
   %SYS>do ^SECURITY
   ```

4. In **^SECURITY**, select option **14**, **KMIP server setup**.
5. In the **KMIP server setup** choices, select option **2**, **Edit KMIP server**.
6. At the **Edit KMIP server** prompt, enter the name of the configuration to edit.
7. **^SECURITY** then presents prompts for the same properties as when creating a KMIP server configuration; it uses the existing values for the configuration's properties as its defaults. Modify these values as required.
8. After the prompts for KMIP server properties, confirm any edits to the properties of the KMIP server at the **Confirm changes to KMIP server <servername>** prompt.

2.2.3 Delete a KMIP Server Configuration

To delete a KMIP server configuration:

1. For the relevant instance, start the Terminal and log in as a sufficiently privileged user.
2. At the terminal prompt, go to the %SYS namespace:

```
>set $namespace="%SYS"
```

3. Run **^SECURITY**

```
%SYS>do ^SECURITY
```

4. In **^SECURITY**, select option **14**, **KMIP server setup**.
5. In the **KMIP server setup** choices, select option **5**, **Delete KMIP server**.
6. At the **KMIP server to delete?** prompt, enter the name of the configuration to delete.
7. Confirm the deletion when prompted.

2.2.4 List the KMIP Server Configurations

To list an InterSystems IRIS instance's KMIP server configurations:

1. For the relevant instance, start the Terminal and log in as a sufficiently privileged user.
2. At the terminal prompt, go to the %SYS namespace:

```
>set $namespace="%SYS"
```

3. Run **^SECURITY**

```
%SYS>do ^SECURITY
```

4. In **^SECURITY**, select option **14**, **KMIP server setup**.
5. In the **KMIP server setup** choices, select option **3**, **List KMIP servers**.

^SECURITY then displays a list of any existing configurations to KMIP servers by name, whether or not they are currently in use.

2.2.5 List Details about a KMIP Server Configuration

To view details about a particular KMIP server configuration:

1. For the relevant instance, start the Terminal and log in as a sufficiently privileged user.

2. At the terminal prompt, go to the %SYS namespace:

   ```
   >set $namespace="%SYS"
   ```

3. Run **^SECURITY**

   ```
   %SYS>do ^SECURITY
   ```

4. In **^SECURITY**, select option **14**, **KMIP server setup**
5. In the **KMIP server setup** choices, select option **4**, **Detailed list KMIP server**.
6. Enter the name of a KMIP server configuration at the *Display which KMIP configuration?* prompt.

^SECURITY then displays a list of the specified configuration's properties, along with each one's value.

2.2.6 Create a Key on the KMIP Server

To create a data-encryption key on a KMIP server:

1. For the relevant instance, start the Terminal and log in as a sufficiently privileged user.
2. At the terminal prompt, go to the %SYS namespace:

   ```
   >set $namespace="%SYS"
   ```

3. Run **^EncryptionKey**:

   ```
   %SYS>do ^EncryptionKey
   ```

4. In **^EncryptionKey**, select option **5**, **Manage KMIP server**.
5. When prompted, enter the name of the configuration for the KMIP server on which you wish to create a key.
6. At the next prompt, where you select the action you wish to take, select option **2**, for **Create new key on KMIP server**.
7. At the next prompt, select a key length.

The **^EncryptionKey** routine then creates the key and displays its key ID. Newly created keys are not activated by default; to activate the key, see Activate a Database Encryption Key from a KMIP Server.

Important: InterSystems recommends that you record the key ID, so that you have this information available for future reference.

2.2.7 Delete a Key on the KMIP Server

To delete an encryption key on a KMIP server:

1. For the relevant instance, start the Terminal and log in as a sufficiently privileged user.
2. At the terminal prompt, go to the %SYS namespace:

   ```
   >set $namespace="%SYS"
   ```

3. Run **^EncryptionKey**:

   ```
   %SYS>do ^EncryptionKey
   ```

4. In **^EncryptionKey**, select option **5**, **Manage KMIP server**.
5. When prompted, enter the name of the configuration for the KMIP server on which you wish to delete the key.

6. At the next prompt, where you select the action you wish to take, select option **3**, for **Destroy existing key on KMIP server**.
7. The routine then lists the keys on the KMIP server and prompts for the key to delete. Specify a key at the **Select key** prompt.

   WARNING! Before deleting the only existing copy of a key, it is critical that you are absolutely sure that there is no existing encrypted content that uses it. If there is no copy the key required to decrypt data, the encrypted data that it protected will be unreadable and will be *permanently lost*.

8. When prompted, confirm that you wish to delete the key.

The routine then deletes the key from the KMIP server.

2.2.8 List the Keys on the KMIP Server

To list the encryption keys on a KMIP server:

1. For the relevant instance, start the Terminal and log in as a sufficiently privileged user.
2. At the terminal prompt, go to the %SYS namespace:

   ```
   >set $namespace="%SYS"
   ```

3. Run **^EncryptionKey**:

   ```
   %SYS>do ^EncryptionKey
   ```

4. In **^EncryptionKey**, select option **5**, **Manage KMIP server**.
5. When prompted, enter the name of the configuration of the KMIP server for which you wish to list the key(s).
6. At the next prompt, select option **1**, for **List keys on KMIP server**.

The routine then displays a list of all the keys on the KMIP server.

2.2.9 Activate a Database Encryption Key from a KMIP Server

To activate a database encryption key from a KMIP server:

1. For the relevant instance, start the Terminal and log in as a sufficiently privileged user.
2. At the terminal prompt, go to the %SYS namespace:

   ```
   >set $namespace="%SYS"
   ```

3. Run **^EncryptionKey**:

   ```
   %SYS>do ^EncryptionKey
   ```

4. In **^EncryptionKey**, select option **3**, **Database encryption**.
5. In the **Database encryption** choices, select option **1**, **Activate database encryption keys**.
6. In the **Activate database encryption keys** choices, select option *2*, *Use KMIP server*.

   Note: If this prompt does not appear, it is because the instance does not have any KMIP server configurations; see Create a KMIP Server Configuration for instructions on this process.

7. When prompted, enter the name of the configuration of the KMIP server from which you wish to activate the key.

8. The routine then lists the keys on the KMIP server and prompts for which key to activate. Specify a key at the **Select key** prompt.

The routine then activates the key, displaying its ID.

For each key that InterSystems IRIS activates, the **Database Encryption** page (**System Administration** > **Encryption** > **Database Encryption**) adds the key to table of activated keys and displays the key's identifier.

Note: The table of keys does not display any file or path information. When activating a second or subsequent key, note the identifier(s) for the currently activated key(s) first, so that you can identify the new one.

2.2.10 Activate a Data-Element Encryption Key from a KMIP Server

InterSystems IRIS supports up to four activated keys at one time for data-element encryption. To activate a key for data-element encryption from a KMIP server:

1. For the relevant instance, start the Terminal and log in as a sufficiently privileged user.
2. At the terminal prompt, go to the %SYS namespace:

   ```
   >set $namespace="%SYS"
   ```

3. Run **^EncryptionKey**:

   ```
   %SYS>do ^EncryptionKey
   ```

4. In **^EncryptionKey**, select option **4**, **Data element encryption for applications**.
5. In the **Data element encryption for applications** choices, select option **1**, **Activate data element encryption key**.
6. In the **Activate data element encryption key** choices, select option **2**, **Use KMIP server**.

   Note: If this prompt does not appear, it is because the instance does not have any KMIP server configurations; see Create a KMIP Server Configuration for instructions on this process.

7. At the KMIP server prompt, enter the name of the configuration of the KMIP server from which you wish to activate the key.
8. The routine then lists the keys on the KMIP server and prompts for which key to activate. Specify a key at the **Select key** prompt.

The routine then activates the key, displaying its ID.

For each key that InterSystems IRIS activates, the **Data Element Encryption** page (**System Administration** > **Encryption** > **Data Element Encryption**) adds the key to table of activated keys and displays the key's identifier.

Note: The table of keys does not display any file or path information. When activating a second or subsequent key, note the identifier(s) for the currently activated key(s) first, so that you can identify the new one.

2.2.11 Copy a Key from a KMIP Server to a Key File

You can copy a database encryption key from a KMIP server to a key file. This allows you to make keys available both for backup and for recovery from a network or KMIP service outage. You can:

- Create a database encryption key file with a copy of a key from a KMIP server
- Add a copy of a database encryption key from a KMIP server to an existing encryption key file

Important: Always store encryption key files on removable devices that are kept in securely locked storage.

2.2.11.1 Create a Key File with a Copy of a Key from a KMIP Server

To create a key file and copy a key from a KMIP server to it:

1. For the relevant instance, start the Terminal and log in as a sufficiently privileged user.
2. At the terminal prompt, go to the %SYS namespace:

   ```
   >set $namespace="%SYS"
   ```

3. Run **^EncryptionKey**:

   ```
   %SYS>do ^EncryptionKey
   ```

4. In **^EncryptionKey**, select option **1**, **Create new encryption key file**.
5. At the prompts that follow, specify:
   - The name of the key file (which is relative the <install-dir>/mgr/ directory)
   - A description of the key file
   - The name of an administrator for the key — this is a new administrator and can have a new name
   - The password (then confirmed) for that administrator — this is a new password, which can have any valid value
   - Available cipher security levels — the length of the key used to encrypt keys stored in the file
6. At the next prompt, select option **2**, **Copy key from KMIP server**. **^EncryptionKey** then prompts for the key to copy to the file.
7. At the **Select key** prompt, specify the number of the key to copy.

^EncryptionKey then creates the file with the administrator username and password that you specified, and places the selected key in that file.

2.2.11.2 Adding a Copy of a Key from a KMIP Server to an Existing Key File

To add a key from a KMIP server to an existing key file:

1. For the relevant instance, start the Terminal and log in as a sufficiently privileged user.
2. At the terminal prompt, go to the %SYS namespace:

   ```
   >set $namespace="%SYS"
   ```

3. Run **^EncryptionKey**:

   ```
   %SYS>do ^EncryptionKey
   ```

4. In **^EncryptionKey**, select option **2**, **Manage existing encryption key file**.
5. At the **Encryption key file** prompt, enter the path and name of the key file to which you are adding a key. The path is relative to the <install-dir>/mgr/ directory.
6. At the next prompt, select option **5**, **Add encryption key**. At the prompts that follow:
   a. Under **Existing administrator**, enter the **Username** and **Password** of an administrator for the key file.
   b. Enter a description of the key that you are adding to the key file.
7. At the next prompt, select option **2**, **Copy key from KMIP server**. At the prompts that follow:

   a. At the **KMIP server** prompt, enter the name of the KMIP server from which you are copying the key.
   b. At the **Select key** prompt, specify the number of the key to copy.

^EncryptionKey then adds the selected key to selected key file.

2.3 Storage-Independent Key Management Tasks

Some tasks are the same for keys in files and keys on a KMIP server:

- Deactivate a Database Encryption Key
- Deactivate a Data-Element Encryption Key
- Specify the Default Database Encryption Key or Journaling Encryption Key for an Instance

2.3.1 Deactivate a Database Encryption Key

To deactivate a database encryption key:

1. From the Management Portal home page, go to the **Database Encryption** page (**System Administration** > **Encryption** > **Database Encryption**). If a key is currently activated, its identifier appears in the table of keys.
2. You cannot deactivate a key if it is the default key either for new encrypted databases or for encrypting journal files. If you wish to deactivate a key that is InterSystems IRIS is using for either of these activities, then you must select a different key to be used for them. Do this by clicking **Set Default** or **Set Journal** for another key. Once the key is not in use for either of these activities, its **Deactivate** button will be available.
3. To deactivate the key, click **Deactivate** in its row.

   Note: If it is not possible to deactivate the key for some other reason, the Portal displays an error message. InterSystems IRIS does not allow you deactivate a key under the following circumstances:

   - The IRISTEMP and IRISLOCALDATA databases are encrypted.
   - There is a currently-mounted encrypted database (other than IRISTEMP and IRISLOCALDATA) that is encrypted with this key.
   - The key is currently in use to encrypt journal files. (Note that if you change the journal file encryption key, until you switch the journal file, InterSystems IRIS continues to use the old key for encryption.)

   See below for information about how to address the underlying condition.

4. Click **OK** on the confirmation dialog to deactivate the key.

To deactivate the key, each underlying condition requires a different action:

- For any encrypted database except IRISTEMP and IRISLOCALDATA, dismount the database on the **Databases** page (**System Operation** > **Databases**). You can then deactivate the key.
- For IRISTEMP and IRISLOCALDATA, specify that these databases are not to be encrypted and then restart InterSystems IRIS. To do this, select **Configure Startup Settings** on the **Database Encryption** page; either you can choose not to activate a database encryption key at startup (in which case InterSystems IRIS turns off encryption for IRISTEMP and IRISLOCALDATA) or you can choose interactive or unattended database encryption key activation at startup (in which cases the choice whether or not to encrypt IRISTEMP and IRISLOCALDATA becomes available — choose **No**).

- For encrypted journal files, ensure that no encrypted journal file is required for recovery. This is described in Encrypted Journal Files.

2.3.2 Deactivate a Data-Element Encryption Key

To deactivate a data-element encryption key:

1. From the Management Portal home page, go to the **Data Element Encryption** page (**System Administration** > **Encryption** > **Data Element Encryption**) page. If there are any activated keys, the page displays a table listing them.
2. In the table of activated keys, for the key you wish to deactivate, click **Deactivate**. This displays a confirmation dialog.
3. In the confirmation dialog, click **OK**.

When the **Data Element Encryption** page appears again, the row in the table for the deactivated key should no longer be present.

2.3.3 Specify the Default Database Encryption Key or Journal Encryption Key for an Instance

Each instance has a default database encryption key and a default journal encryption key. The instance sets the initial value for each of these when an administrator first activates a database encryption key; the key that is initially the default depends on the key(s) that are in the activated key file. These values are preserved across InterSystems IRIS shutdowns and restarts.

To specify a new key for either of these purposes:

1. From the Management Portal home page, go to the **Database Encryption** page (**System Administration** > **Encryption** > **Database Encryption**. This displays a table of currently activated encryption keys for the instance.
2. In the table of encryption keys:
    - To specify a new default encryption key, click **Set Default** for that key. The **Set Default** button for the current default key is unavailable.
    - To specify a new journal encryption key, click **Set Journal** for that key. The **Set Journal** button for the current journal encryption key is unavailable.
3. When prompted to confirm your action, click **OK.**

InterSystems IRIS then sets the selected key as the default or journal encryption key. If a key is either the default or journal encryption key, then it cannot be deleted (since it is required for operations on the InterSystems IRIS instance). Hence, specifying either of these for a key makes the key’s **Delete** button unavailable.

3

Using Encrypted Databases

To protect entire databases that contain sensitive information, InterSystems IRIS® data platform supports block-level database encryption (or, for short, database encryption). Database encryption is technology that allows you to create and manage databases that, as entire entities, are encrypted; it employs the InterSystems IRIS key management tools to support its activities.

When you create a database, you can choose to have it be encrypted; this option is available if there is a currently activated key. Once you have created an encrypted database, you can use it in the same way as you would use an unencrypted database. The encryption technology is transparent and has a small and deterministic performance effect.

This topic describes how to create and manage encrypted databases. The database encryption functionality also supports the ability to encrypt the audit log and journal files. Both these features require access to the database encryption key at startup time, as described in Configure Encryption Startup Settings.

3.1 Create an Encrypted Database

When creating a new database, you can specify that it is encrypted. However, before you can create an encrypted database, InterSystems IRIS must have an activated database encryption key. Hence, the procedure is:

1. Activate a database encryption key.
2. From the Management Portal home page, go to the **Local Databases** page (**System Administration** > **Configuration** > **System Configuration** > **Local Databases**).
3. On the **Local Databases** page, select **Create New Database**. This displays the **Create Database** wizard.
4. On the second page of the wizard, in the **Encrypt Database?** box, select **Yes**. This causes InterSystems IRIS to create an encrypted database. On all the other pages of the wizard, choose database characteristics as you would when creating any database. (For more information on creating databases, see Create Local Databases.)

Note: InterSystems IRIS also provides encryption management tools to encrypt unencrypted databases or decrypt encrypted databases, if this is necessary.

3.2 Establish Access to an Encrypted Database

To perform various operations, such as adding a database to a mirror, the database must be mounted. However, for an encrypted database to be mounted, its key must be activated. Hence, access to the database requires activating the key and mounting the database, and the procedure for this is:

1. Activate the key.
2. From the Management Portal home page, go to the **Databases** page (**System Operation** > **Databases**).
3. On this page, for the database that you wish to mount, select the **Mount** button in the far right column of its row in the table of databases. After selecting **OK** on the confirmation screen, the database is mounted. If the key is not activated, InterSystems IRIS cannot mount the database and displays an error message.

You can now access the data within the database.

3.3 Close the Connection to an Encrypted Database

To close the connection to an encrypted database, the procedure is:

1. From the Management Portal home page, go to the **Databases** page (**System Operation** > **Databases**).
2. On this page, select the **Dismount** button on the right in the table of databases. After selecting **OK** on the confirmation screen, the database is dismounted.
3. Deactivate the key.

Because the activated key is used for each read and write to the database, you cannot deactivate the key without first dismounting the database. If you attempt to deactivate the key prior to dismounting the database, InterSystems IRIS displays an error message.

3.4 Move an Encrypted Database Between Instances

If your organization has multiple InterSystems IRIS instances, you may need to use an encrypted database on one instance that was created on another instance using a different key. To move the data from one instance to another, back up and then re-encrypt the database using the available encryption management tools. For more information, see Modify Database Encryption Using ^EncryptionKey.

3.5 Configure Encryption Startup Settings

This topic describes how to set up each of the three database encryption startup options:

- Startup without Key Activation (the default) — The instance does not have a database encryption key available at startup time.
- Startup with Interactive Key Activation — The instance gathers database encryption key information at startup time interactively.

- Startup with Unattended Key Activation — The instance gathers database encryption key information at startup time without human intervention. This is also known as *unattended startup*.

InterSystems IRIS has several features that require having a key available at startup time (either interactively or through unattended startup):

- Encrypting the InterSystems IRIS audit log.
- Encrypting the IRISTEMP and IRISLOCALDATA databases. (Either both are encrypted or neither.)
- Encrypting InterSystems IRIS journal files.
- Having an encrypted database mounted at startup time.

3.5.1 Startup without Key Activation

This is the default behavior for an instance of InterSystems IRIS prior to having any keys activated. If you have set up key activation at startup and choose to turn it off, the procedure is:

1. From the Management Portal home page, go to the **Database Encryption** page (**System Administration** > **Encryption** > **Database Encryption**).
2. Select **Configure Startup Settings**. This displays the area with options for configuring InterSystems IRIS startup and other options for encrypted databases.
3. In this area, from the **Startup Options** list, select **None**.
4. Click **Save**. InterSystems IRIS may prevent you from performing this action if:
   - Any encrypted databases are required at startup. See Encrypted Databases Required at Startup for more details.
   - There are any encrypted journal files with open transactions. See Encrypted Journal Files for more details.
   - The audit log is encrypted. (The error message for this refers to an encrypted database because InterSystems IRIS stores the audit log in a database called IRISAUDIT.) See Encrypted Audit Log for more details.

   Address the issue that is preventing the change and then perform this procedure again. Once any issues are corrected, you will be able to successfully change to having startup without key activation.

3.5.1.1 Encrypted Databases Required at Startup

If the instance has encrypted databases that are required at startup and you attempt to configure startup not to involve key activation, the Management Portal displays an error message stating this and indicating that the key activation option cannot be changed. (If the error message refers to the IRISAUDIT database, this is because the audit log is encrypted.)

To configure InterSystems IRIS to start without activating an encryption key, any encrypted databases can only be mounted after startup. To configure a database to be mounted after startup:

1. Confirm that the database is mounted or mount it:
   a. From the Management Portal home page, go to the **Databases** page (**System Operation** > **Databases**).
   b. Find the database's row in the table of databases. If it is mounted, there is a **Dismount** choice in its row; if it is not mounted, there is no **Dismount** choice and there *is* a **Mount** choice.
   c. If it is not mounted, select **Mount**
   d. On the confirmation screen, select **OK**. (The database needs to be writable, so do not select the **Read Only** check box.)

2. Edit the database's properties so that it is not mounted at startup:

   a. Go to the **Local Databases** page (**System Administration** > **Configuration** > **System Configuration** > **Local Databases**).
   b. Find the database's row in the table of databases.
   c. Select the database by clicking on its name. This displays the page for editing the database.
   d. On this **Edit** page, clear the **Mount Required at Startup** check box.
   e. Click **Save**.

The database will no longer be mounted at startup. This means that it will no longer require key activation at startup (though it may be required for other reasons.)

3.5.1.2 Encrypted Journal Files

If the instance uses journaling and you attempt to configure startup not to involve key activation, you may be unable to turn off key activation at startup. These conditions are:

- The instance is configured to encrypt its journal files.
- There are open transactions in the journal file (which is fairly likely on a busy system).

If this occurs, you need to suspend the use of encrypted journal files before you change the startup key activation settings. To do this, the procedure is:

1. On the **Database Encryption** page (**System Administration** > **Encryption** > **Database Encryption**), change the **Encrypt Journal Files** setting to **No**. Leave the **Key Activation at Startup** setting as it is.
2. Switch journal files. To do this, click **Switch Journal** on the **Journals** page (**System Operation** > **Journals**).

Once all open transactions within the encrypted journal files have either been committed or rolled back, you can then change the InterSystems IRIS startup configuration.

CAUTION: Even after there are no open transactions, you may need the encrypted journal files to restore a database. For this reason, it is very important that you maintain copies of the key file containing the key used to encrypt these files.

For more information on journal files generally, see Journaling.

3.5.1.3 Encrypted Audit Log

If the instance has an encrypted audit log and you attempt to configure startup not to involve key activation, InterSystems IRIS displays an error message that an encrypted database is required at startup, such as:

```
ERROR #1217: Can not disable database encryption key activation at startup.
Encrypted databases are required at startup:
C:\InterSystems\IRIS\Mgr\IRISAudit\
```

The error message refers to encrypted databases because the audit log is stored in the InterSystems IRIS database IRISAUDIT.

The audit log cannot be encrypted if InterSystems IRIS starts without activating an encryption key. To configure startup not to involve key activation, you must change the InterSystems IRIS setting to specify that the instance uses an unencrypted audit log. The procedure is:

1. Back up the instance's audit data.
2. Go to the **Database Encryption** page (**System Administration** > **Encryption** > **Database Encryption**).
3. Select **Configure Startup Settings**, which displays the area with options for configuring InterSystems IRIS startup and other options for encrypted databases.

4. Under **Optionally Encrypted Data**, in the **Encrypt Audit Log** list, click **No**.

Changing this setting causes InterSystems IRIS to erase any existing audit data, to start using unencrypted auditing immediately, and to write an AuditChange event to the audit log.

CAUTION: If you have not backed up audit data, changing the encryption setting for the audit log results in the loss of that existing audit data.

3.5.2 Startup with Interactive Key Activation

This is the default behavior for an instance of InterSystems IRIS if a key has been activated. With interactive key activation, the InterSystems IRIS instance prompts for the location of a key and its associated information during its startup.

Important: On Windows, interactive key activation is incompatible with configuring InterSystems IRIS as a service that starts automatically as part of system startup.

To configure InterSystems IRIS for interactive key activation:

1. From the Management Portal home page, go to the **Database Encryption** page (**System Administration** > **Encryption** > **Database Encryption**).
2. Select **Configure Startup Settings**. This displays the **Startup Options** area, which includes the **Key Activation at Startup** list.
3. In the **Key Activation at Startup** list, select **Interactive**. If the previous value for the field was **None**, then this displays the page's **Optionally Encrypted Data** area.
4. The fields in this area are:
   - **Encrypt IRISTEMP and IRISLOCALDATA Databases** — Allows you to specify whether or not the IRISTEMP and IRISLOCALDATA databases are encrypted. To encrypt them, select **Yes**; to have them be unencrypted, select **No**.
   - **Encrypt Journal Files** — Allows you to specify whether or not the instance encrypts its own journal files. To encrypt journal files, select **Yes**; to have them be unencrypted, select **No**. This choice depends on startup options because InterSystems IRIS startup creates a new journal file; if you choose encryption, startup requires a key.

     Note: This change takes effect the next time that InterSystems IRIS switches journal files. To begin journal file encryption without a restart, switch journal files after completing this page.

   - **Encrypt Audit Log** — Allows you to specify whether or not InterSystems IRIS encrypts the audit log. To encrypt the audit log, select **Yes**; to have it be unencrypted, select **No.** This choice depends on startup options because the InterSystems IRIS startup procedure records various events in the audit log; if you choose encryption, startup requires a key.

     CAUTION: This change takes effect immediately and deletes any existing audit data. Back up the audit database prior to changing this setting; otherwise, audit data will be lost.

5. Click **Save** to save the selected settings.

Important: If InterSystems IRIS is configured to

- Encrypt IRISTEMP and IRISLOCALDATA, journal files, or the audit log
- Require an encrypted database at startup

then failure to activate the required encryption key causes an InterSystems IRIS startup failure. If this occurs, use InterSystems IRIS emergency startup mode to configure InterSystems IRIS not to require any encrypted facilities at startup.

3.5.3 Startup with Unattended Key Activation

Startup with unattended key activation, also known as *unattended startup*, activates a key and potentially mounts encrypted databases at startup time without any human intervention. Successful unattended startup requires that the instance have access to:

- The encrypted database
- The database encryption key, either through:
  - The KMIP server that holds the key
  - The database encryption key file that holds the key and the username and password used for unattended database encryption key activation

This section includes the following topics:

- Configuring Unattended Startup Using a Key on a KMIP Server
- Configuring Unattended Startup Using a Key in a Key File
- Temporarily Addressing Issues with Unattended Startup

CAUTION: By making all these items available, the security of the data in InterSystems IRIS becomes entirely dependent on the physical security of the machine(s) holding these elements. If your site cannot ensure this physical security, your data will then be subject to the same level of risk as if it were not encrypted; to avoid this situation, either use interactive startup (which prevents the simultaneous exposure of these elements) or ensure the physical security of the relevant machine(s).

3.5.3.1 Configuring Unattended Startup Using a Key on a KMIP Server

To configure an InterSystems IRIS instance for unattended startup using a key on a KMIP server:

1. For the relevant instance, start the Terminal and log in as a sufficiently privileged user.
2. At the terminal prompt, go to the %SYS namespace:

   ```
   >set $namespace="%SYS"
   ```

3. Run **^EncryptionKey**:

   ```
   %SYS>do ^EncryptionKey
   ```

4. In **^EncryptionKey**, select option **3**, **Database encryption**.
5. At the next prompt, select option **4**, **Configure startup options**.
6. At the next prompt, select option **4**, **Unattended key activation with a KMIP server**.
7. At the **KMIP server instance name** prompt, enter the name of a KMIP server configuration.

8. At the prompts that follow, specify what items to encrypt (all of which require an activated key at startup time):

   - **Encrypt journal files** (no by default) — Allows you to specify whether or not the instance encrypts its own journal files. To encrypt journal files, enter **yes**; to have them be unencrypted, enter or select **no** (the default). This choice depends on startup options because InterSystems IRIS startup creates a new journal file; if you choose encryption, startup requires a key.

     This change takes effect the next time that InterSystems IRIS switches journal files. By default, this occurs the next time that InterSystems IRIS restarts. To begin journal file encryption without a restart, switch journal files after completing this page.

   - **Encrypt IRISTEMP and IRISLOCALDATA databases** (no by default) — Allows you to specify whether or not the IRISTEMP and IRISLOCALDATA databases are encrypted. To encrypt them, enter **yes**; to have them be unencrypted, enter or select **no** (the default).

   - **Encrypt audit database** (no by default) — Allows you to specify whether or not InterSystems IRIS encrypts the audit log. To encrypt the audit log, select **yes**; to have it be unencrypted, select **no** (the default). This choice depends on startup options because the InterSystems IRIS startup procedure records various events in the audit log; if you choose encryption, startup requires a key.

     CAUTION: This change takes effect immediately and deletes any existing audit data. Back up the audit database prior to changing this setting; otherwise, audit data will be lost.

9. The routine then displays the current list of KMIP keys to activate at startup, and then prompts for the next action:

   - To add a key to the list of the startup keys, select option **1**, **Add key to list**.
   - To remove a key from the list of the startup keys, select option **2**, **Delete key from list**.
   - To save the list of startup keys, select option **3**, **Save list**.

10. When the list contains the desired list of KMIP keys to activate at startup, select option **3**, which saves the list.

3.5.3.2 Configuring Unattended Startup Using a Key in a Key File

CAUTION: When you configure InterSystems IRIS for unattended startup, the instance adds another administrator to the database encryption key file; that administrator has a system-generated name and password. Once InterSystems IRIS has modified the key file to add this username and password, InterSystems strongly recommends that you place any copies of the key file only on hardware that can be physically locked in place, such as a lockable CD-ROM or DVD drive in a rack. Further, you should lock and monitor the data center facility where this hardware is stored. Do *not* store the database encryption key on the same drive as any databases that it is used to encrypt.

To configure an InterSystems IRIS instance for unattended startup with a key in a key file:

1. You need to have a key currently activated. To activate a key, see Activating a Key.
2. From the Management Portal home page, go to the **Database Encryption** page (**System Administration** > **Encryption** > **Database Encryption**).
3. Select **Configure Startup Settings**. This displays the **Startup Options** list.
4. In **Startup Options**, select **Unattended (NOT RECOMMENDED)**. This changes the fields that the page displays.
5. The **Startup Options** area expands to display three fields. Complete these:

   - **Key File** — The path of the database encryption key file. This can be an absolute or relative path; if you specify a relative path, it is relative to the InterSystems IRIS installation directory. Click **Browse** to search for the database encryption key file on the file system.

- **Administrator Name** — An administrator for this key file.
- **Password** — The administrator’s password.

6. Complete the fields in the **Optionally Encrypted Data** area:
   - **Encrypt IRISTEMP and IRISLOCALDATA Databases** — Allows you to specify whether or not the IRISTEMP and IRISLOCALDATA databases are encrypted. To encrypt them, select **Yes**; to have them be unencrypted, select **No**.
   - **Encrypt Journal Files** — Allows you to specify whether or not the instance encrypts its own journal files. To encrypt journal files, select **Yes**; to have them be unencrypted, select **No.** This choice depends on startup options because InterSystems IRIS startup creates a new journal file; if you choose encryption, startup requires a key.

     Note: This change takes effect the next time that InterSystems IRIS switches journal files. By default, this occurs the next time that InterSystems IRIS restarts. To begin journal file encryption without a restart, switch journal files after completing this page.

   - **Encrypt Audit Log** — Allows you to specify whether or not InterSystems IRIS encrypts the audit log. To encrypt the audit log, select **Yes**; to have it be unencrypted, select **No.** This choice depends on startup options because the InterSystems IRIS startup procedure records various events in the audit log; if you choose encryption, startup requires a key.

     CAUTION: This change takes effect immediately and deletes any existing audit data. Back up the audit database prior to changing this setting; otherwise, audit data will be lost.

7. Click **Save** to save the selected settings.

3.5.3.3 Temporarily Addressing Issues with Unattended Startup

If InterSystems IRIS is configured to

- Encrypt IRISTEMP and IRISLOCALDATA, journal files, or the audit log
- Require an encrypted database at startup

then failure to activate the encryption key causes an InterSystems IRIS startup failure. If this occurs, use InterSystems IRIS emergency startup mode to configure InterSystems IRIS not to require any encrypted facilities at startup.

3.6 Encrypt the Databases that Ship with InterSystems IRIS

Each instance of InterSystems IRIS ships with a number of databases. The ability to encrypt and the value of encryption depends on the database:

- IRISLOCALDATA: Can be encrypted in conjunction with the IRISTEMP database. Encrypting IRISLOCALDATA requires that a key be available at startup, since the database is required at startup time.
- IRISAUDIT: Can be encrypted. Encrypting IRISAUDIT requires that a key be available at startup, since the database is required at startup time.
- IRISLIB: Must not be encrypted. (Note that all content in IRISLIB is publicly available.)
- IRISSYS: Must not be encrypted. If an instance has an encrypted form of this database, InterSystems IRIS cannot start.

- IRISTEMP: Can be encrypted in conjunction with the IRISLOCALDATA database. Encrypting IRISTEMP requires that a key be available at startup, since the database is required at startup time.
- USER: Can be encrypted.

3.7 Modify Database Encryption Using ^EncryptionKey

There are occasions when you may need to perform encryption management operations that are not available through the Management Portal. Using the **^EncryptionKey** utility, you can perform the following actions:

- Convert an Unencrypted Database to Be Encrypted
- Convert an Encrypted Database to Be Unencrypted
- Convert an Encrypted Database to Use a New Key

The following is true about the tools used by the **^EncryptionKey** utility:

The **^EncryptionKey** utility uses a set of encryption management tools:

- When built-in hardware instructions are available for encryption-related activities, these activities are considerably faster than when using software-based encryption. The encryption management tools use hardware instructions when they are available.
- The encryption management tools can use keys stored on a KMIP server.
- The encryption management tools can run in FIPS mode.

Note: The encryption management tools do not operate on journal files.

3.7.1 Convert an Unencrypted Database to be Encrypted

To convert an unencrypted database to an encrypted database:

1. Back up the data in the database to be encrypted.

   InterSystems IRIS encrypts data in place. This means that it uses on-disk space for its operations (not copying the database elsewhere and restoring it to its current disk location after successful completion). If the utility is interrupted before completion, the database will be partly encrypted and partly unencrypted, rendering it unusable.

   CAUTION: It is critical that you back up the database before converting it. Failure to do so can result in data being lost.

2. Activate the key with which you wish to encrypt the database, either from a key file or a KMIP server.
3. Start the Terminal.
4. In the %SYS namespace, run the **^EncryptionKey** utility.
5. In **^EncryptionKey**, select option **3**, **Database encryption**.
6. In the database encryption submenu, select option **7**, **Modify encrypted status of existing database**.
7. In the **Database directories** submenu, select the database that you wish to modify; databases are listed by their directories. When you select a database, the routine announces if the database is encrypted or not.

8. If the database is unencrypted, the routine allows you to encrypt it; at the **Encrypt database?** prompt, enter `yes` or `y`. This is not case sensitive.
9. At the **Select key for encryption prompt**, select the key that the routine will use to encrypt the database. If the database is currently mounted, the routine then displays this information.
10. If the database is currently mounted, the routine states this. At the **Dismount database** prompt, enter `yes` or `y`. This is not case sensitive.

    Important: Because dismounting and then remounting a database interrupts its operations, take the appropriate precautions to ensure that this does not cause problems.

The routine then encrypts the database. As part of this process, if the database was mounted, the routine displays messages that it has dismounted and mounted the database. When the database is mounted again, encryption is complete.

3.7.2 Convert an Encrypted Database to be Unencrypted

To convert an encrypted database to an unencrypted database:

1. Back up the data in the database to be unencrypted.

   InterSystems IRIS unencrypts data in place. This means that it uses on-disk space for its operations (not copying the database elsewhere and restoring it to its current disk location after successful completion). If the utility is interrupted before completion, the database will be partly encrypted and partly unencrypted, rendering it unusable.

   CAUTION: It is critical that you back up the database before converting it. Failure to do so can result in data being lost.

2. Activate the key with which you wish to encrypt the database, either from a key file or a KMIP server.
3. Start the Terminal.
4. In the `%SYS` namespace, run the **^EncryptionKey** utility.
5. In **^EncryptionKey**, select option **3**, **Database encryption**.
6. In the database encryption submenu, select option **7**, **Modify encrypted status of existing database**.
7. In the **Database directories** submenu, select the database that you wish to modify; databases are listed by their directories. When you select a database, the routine announces if the database is encrypted or not. If the database is encrypted and its encryption key has not been activated, the routine announces this as well.
8. If the database is encrypted, the routine allows you to decrypt it; at the **Decrypt database?** prompt, enter `yes` or `y`. This is not case sensitive.
9. After reporting the encryption key for the database, the routine prompts if you wish to encrypt the database with a different key. Press Enter to simply convert it to a decrypted database and use a new key to encrypt it.
10. If the database is currently mounted, the routine states this. At the **Dismount database** prompt, enter `yes` or `y`. This is not case sensitive.

    Important: Because dismounting and then remounting a database interrupts its operations, take the appropriate precautions to ensure that this does not cause problems.

The routine then decrypts the database. As part of this process, if the database was mounted, the routine displays messages that it has dismounted and mounted the database. When the database is mounted again, decryption is complete.

3.7.3 Convert an Encrypted Database to Use a New Key

To convert an encrypted database to use a new key:

1. Back up the data in the database to be re-encrypted.

   InterSystems IRIS encrypts data in place. This means that it uses on-disk space for its operations (not copying the database elsewhere and restoring it to its current disk location after successful completion). If the utility is interrupted before completion, the database will be partly encrypted and partly unencrypted, rendering it unusable.

   CAUTION: It is critical that you back up the database before converting it. Failure to do so can result in data being lost.

2. Activate the keys with which the database is encrypted and with which you wish to re-encrypt the database, either from a key file or a KMIP server.
3. Start the Terminal.
4. In the `%SYS` namespace, run the **^EncryptionKey** utility.
5. In **^EncryptionKey**, select option **3**, **Database encryption**.
6. In the database encryption submenu, select option **7**, **Modify encrypted status of existing database**.
7. In the **Database directories** submenu, select the database that you wish to modify; databases are listed by their directories. When you select a database, the routine announces if the database is encrypted or not.
8. If the database is encrypted, the routine allows you to decrypt it; at the **Decrypt database?** prompt, enter `yes` or `y`. This is not case sensitive.
9. At the next prompt, which is the **Re-encrypt database?** prompt, enter `yes` or `y`. This is not case sensitive.
10. At the **Select key for encryption prompt**, select the key that the routine will use to encrypt the database.
11. If the database is currently mounted, the routine states this. At the **Dismount database** prompt, enter `yes` or `y`. This is not case sensitive.

    Important: Because dismounting and then remounting a database interrupts its operations, take the appropriate precautions to ensure that this does cause problems.

The routine then re-encrypts the database. As part of this process, if the database was mounted, the routine displays messages that it has dismounted and mounted the database. When the database is mounted again, encryption is complete.

4

Using Data-Element Encryption

Data-element encryption provides a means of encrypting application data at a finer level of granularity than the database as a whole; it is for sensitive data elements whose exposure must be prevented. For example, customer records can exclusively encrypt the credit card field; patient records can exclusively encrypt fields that display test results (such as for HIV testing); or a record that includes a social security number can exclusively encrypt that field.

Data-element encryption is available programmatically (via an API), rather than through the Management Portal. Because it is accessible through an API, you can use it in your application code. You have the option of using data-element encryption with database encryption (though there is no requirement to use both).

For an application to use data-element encryption, the required keys must be available when the application is running. To make a key available, activate it; for details, either see Programmatically Manage Keys or, if using the Portal, see Activating a Key for Data-Element Encryption. When a key is activated, InterSystems IRIS® data platform displays its unique identifier in the table of activated keys; the application then uses this identifier to refer to the key so that it can be loaded from memory for encryption operations. Since there can be up to four keys simultaneously activated, data-element encryption provides the infrastructure for tasks that require multiple keys.

When encrypting data for data-element encryption, InterSystems IRIS stores the encryption key’s unique identifier with the resulting ciphertext. The unique identifier enables the system to identify the key at decryption time using only the ciphertext itself.

This topic describes:

- Programmatically Manage Keys
- Data-Element Encryption Calls
- Support for Re-Encrypting Data in Real Time

4.1 Programmatically Manage Keys

Since data-element encryption is available through an API, there are also a set of calls for managing keys:

- **$SYSTEM.Encryption.CreateEncryptionKey**
- **$SYSTEM.Encryption.ActivateEncryptionKey**
- **$SYSTEM.Encryption.DeactivateEncryptionKey**
- **$SYSTEM.Encryption.ListEncryptionKeys**

These are all methods of the %SYSTEM.Encryption class.

4.2 Data-Element Encryption Calls

The system methods available for data-element encryption are all methods of the %SYSTEM.Encryption class and are:

- $SYSTEM.Encryption.AESCBCManagedKeyEncrypt
- $SYSTEM.Encryption.AESCBCManagedKeyDecrypt
- $SYSTEM.Encryption.AESCBCManagedKeyEncryptStream
- $SYSTEM.Encryption.AESCBCManagedKeyDecryptStream

These method names all begin with “AESCBCManagedKey” because the methods use AES (the Advanced Encryption Standard) in cipher block chaining (CBC) mode and are part of the suite of tools for managed key encryption.

Important: The AESCBC methods that do not include “ManagedKey” in their names are older methods and cannot be used for these purposes.

4.2.1 $SYSTEM.Encryption.AESCBCManagedKeyEncrypt

The signature of this method as it is usually called is:

```
$SYSTEM.Encryption.AESCBCManagedKeyEncrypt
        (
        plaintext As %String,
        keyID As %String,
        )
    As %String
```

where:

- *plaintext* — The unencrypted text to be encrypted.
- *keyID* — The GUID of the data-encryption key to be used to encrypt plaintext.
- The method returns the encrypted ciphertext.

If the method fails, it throws either the <FUNCTION> or <ILLEGAL VALUE> error. Place calls to this method in a **Try-Catch** loop; for more information on Try-Catch, see The TRY-CATCH Mechanism.

For more details, see the **$SYSTEM.Encryption.AESCBCManagedKeyEncrypt** class reference content.

4.2.2 $SYSTEM.Encryption.AESCBCManagedKeyDecrypt

The signature of this method as it is usually called is:

```
$SYSTEM.Encryption.AESCBCManagedKeyDecrypt
        (
        ciphertext As %String
        )
    As %String
```

where:

- *ciphertext* — The encrypted text to be decrypted.
- The method returns the decrypted plaintext.

If the method fails, it throws either the <FUNCTION> or <ILLEGAL VALUE> error. Place calls to this method in a **Try-Catch** loop; for more information on Try-Catch, see The TRY-CATCH Mechanism.

You do not need to include the key ID with this call, as the key ID is associated with the ciphertext to be decrypted.

For more details, see the **$SYSTEM.Encryption.AESCBCManagedKeyDecrypt** class reference content.

4.2.3 $SYSTEM.Encryption.AESCBCManagedKeyEncryptStream

The signature of this method as it is usually called is:

```
$SYSTEM.Encryption.AESCBCManagedKeyEncryptStream
        (
        plaintext As %Stream.Object,
        ciphertext As %Stream.Object,
        keyID As %String,
        )
    As %Status
```

where:

- *plaintext* — The unencrypted stream to be encrypted.
- *ciphertext* — The variable to receive the encrypted stream.
- *keyID* — The GUID of the data-encryption key to be used to encrypt *plaintext*.
- The method returns a %Status code.

For more details, see the **$SYSTEM.Encryption.AESCBCManagedKeyEncryptStream** class reference content.

4.2.4 $SYSTEM.Encryption.AESCBCManagedKeyDecryptStream

The signature of this method as it is usually called is:

```
$SYSTEM.Encryption.AESCBCManagedKeyDecryptStream
        (
        ciphertext As %Stream.Object,
        plaintext As %Stream.Object
        )
    As %Status
```

where:

- *ciphertext* — The encrypted stream to be decrypted.
- *plaintext* — The variable to receive the unencrypted stream.
- The method returns a %Status code.

You do not need to include the key ID with this call, as the key ID is associated with the ciphertext to be decrypted.

For more details, see the **$SYSTEM.Encryption.AESCBCManagedKeyDecryptStream** class reference content.

4.3 Support for Re-Encrypting Data in Real Time

Data-element encryption allows InterSystems IRIS applications to support re-encrypting an encrypted data element with a new key.

Because an encrypted data element has its encrypting key identifier stored with it, this simplifies the process of re-encrypting data. Given merely the handle to ciphertext and an activated key, an application can perform re-encryption. For example, data-element encryption supports the ability to re-encrypt sensitive data without any downtime; this is particularly useful for users required to perform this action for legal reasons, such as those fulfilling PCI DSS (Payment Card Industry Data Security Standard) requirements.

If you need to re-encrypt data, create a new key and specify to your application that this is the new encryption key. You can then perform an action such as running a background application to decrypt the elements and encrypt them with the new key. This uses the specified key for encryption and always uses the correct key for decryption, since it is stored with the encrypted data.

5

Protecting Against Data Loss

To ensure that encrypted data is always available, InterSystems strongly suggests that you take the following preventative actions:

- If you are using key files, then, for each key that you are using:
  1. Create an additional administrator for the key file.
  2. Record the username and password of that administrator on paper.
  3. Place the recorded username and password in a physically secure location, such as a fireproof safe that is sufficiently far from where the key is in use.
  4. Create a backup copy of the key file and place it in the same secure location as the recorded username and password.
- If you are using a KMIP server, back up the contents of that server according to your server vendor's instructions.

CAUTION: Failure to take these precautions can result in a situation where the encrypted data will be *permanently inaccessible* — there will be no way to read it.

6

Handling Emergency Situations

Emergency situations involving encrypted data can result permanently losing access to the encrypted data. If an emergency situation arises, you must act immediately to minimize the risk of the data being lost forever.

This topic describes the steps to take if an emergency situation with encrypted data arises. To take preventative steps against an emergency situation, see Protecting Against Data Loss.

6.1 Handle Emergency Situations When Using Key Files

This topic describes what to do under certain circumstances when you are in danger of losing data. These situations include:

- If the File Containing an Activated Key is Damaged or Missing
  - If There Is a Backup Copy of the Key File with a Known Administrator Username and Password
  - If There Is No Backup Copy of the Key File or the Key Has No Known Administrator Username and Password

    CAUTION: This is a *dire* situation. Act *immediately*.

- If the Database Encryption Key File Is Required at Startup and Is Not Present
  - If You Can Make the Key File Available
  - If a Backup Key File Is Available
  - If No Key File Is Available

6.1.1 If the File Containing an Activated Key is Damaged or Missing

In this situation, the following circumstances have occurred:

- A database encryption key has been activated for the InterSystems IRIS® data platform instance.
- InterSystems IRIS is using encrypted data.
- The key file containing the database encryption key becomes damaged.

6.1.1.1 If There Is a Backup Copy of the Key File with a Known Administrator Username and Password

CAUTION: This procedure is for an emergency situation, where encrypted data in InterSystems IRIS databases is in danger of being lost.

If the file containing an activated key becomes inaccessible or damaged, *immediately* perform the following procedure:

1. Get the backup copy of the key file. This is the copy that you stored as described in Protection from Accidental Loss of Access to Encrypted Data.
2. Make a new backup copy of the key file and store it in a safe place.
3. Set up InterSystems IRIS to use the new copy of the key:
   - If you are using interactive startup, incorporate the new copy of the key into your startup procedures.
   - If you are using unattended startup, then reconfigure your startup options using the new copy of the key file — even if you are setting it up for the same options as before.

6.1.1.2 If There Is No Backup Copy of the Key File or the Key has No Known Administrator Username and Password

WARNING! THIS PROCEDURE IS FOR AN EMERGENCY SITUATION, WHERE ENCRYPTED DATA IN INTERSYSTEMS IRIS DATABASES IS IN DANGER OF BEING LOST.

If the file containing the activated key becomes inaccessible or damaged while InterSystems IRIS is running, *immediately* perform the following procedure for each database encrypted with that key:

1. **WARNING!** Shutting down InterSystems IRIS or deactivating the active key will cause the permanent loss of your data.

   Do not shut down InterSystems IRIS.

   Do not deactivate the currently active key.

2. Contact the InterSystems Worldwide Response Center. Engineers there can help guide you through the following procedure and answer any questions that may arise.
3. Dismount the database. This prevents all users from making any changes to the database with encrypted content while copying its data to an unencrypted database:
   a. From the Management Portal home page, go to the **Databases** page (**System Operation** > **Databases**).
   b. On the **Databases** page, if the encrypted database is mounted, select the **Dismount** option in the next-to-last column in that database’s row. Then select **OK** in the confirmation dialog.
   c. When the **Databases** page appears again, select the **Mount** option in the last column in the database’s row.
   d. On the **Mount database** confirmation screen, check the **Read Only** box and select **OK**.

   It is critical that no one makes any changes to the database during this procedure. Mounting the database read-only prevents any user from changing any data.

4. Copy all data in unencrypted form to another database. The procedure for copying the data is:
   a. In the Terminal, go to the %SYS namespace:

   ```
   REGULARNAMESPACE>set $namespace="%SYS"
   ```

b. From that namespace, run the **^GBLOCKCOPY** command:

```
%SYS>d ^GBLOCKCOPY

This routine will do a fast global copy from a database to another database or
to a namespace. If a namespace is the destination, the global will follow any
mappings set up for the namespace.

1) Interactive copy
2) Batch copy
3) Exit

Option?1
```

c. At the **^GBLOCKCOPY** prompt, specify 1, for an interactive copy:

```
Option? 1

1) Copy from Database to Database
2) Copy from Database to Namespace
3) Exit

Option?
```

d. When **^GBLOCKCOPY** prompts for a copy type, select 1, for copying from database to database

```
Option? 1
Source Directory for Copy (? for List)?
```

Here, either specify the name of the encrypted database or enter ? to see a numbered list of databases, which includes the encrypted database. If you enter ?, **^GBLOCKCOPY** displays a list such as this one:

```
Source Directory for Copy (? for List)? ?

1) C:\InterSystems\MyIRIS\mgr\
2) C:\InterSystems\MyIRIS\mgr\irislocaldata\
3) C:\InterSystems\MyIRIS\mgr\irisaudit\
4) C:\InterSystems\MyIRIS\mgr\irislib\
5) C:\InterSystems\MyIRIS\mgr\iristemp\
6) C:\InterSystems\MyIRIS\mgr\encrypted1\
7) C:\InterSystems\MyIRIS\mgr\encrypted2\
8) C:\InterSystems\MyIRIS\mgr\unencrypted\

Source Directory for Copy (? for List)?
```

Enter the number of the encrypted database, such as 7 here.

e. When **^GBLOCKCOPY** prompts for a destination directory for copying the data, enter the name of an unencrypted database or ? for a list similar to the one for the source directory.

f. When **^GBLOCKCOPY** asks if you wish to copy all globals, enter `Yes` (can be `Yes`, `Y`, `y`, and so on):

```
All Globals? No => y
```

g. If there is an empty global in the database, **^GBLOCKCOPY** will now ask if you wish to copy it. This will appear something like the following:

```
All Globals? No => y

^oddBIND     contains no data
Include it anyway? No =>
```

Enter `No` (can be `No`, `N`, `n`, and so on), which is the default.

h. **^GBLOCKCOPY** then asks if you wish to skip all the other empty globals. Enter `Yes` (can be `Yes`, `Y`, `y`, and so on), which is the default:

```
Exclude any other similar globals without asking again? Yes =>
```

There then appears a list of all the empty globals that are not being copied:

```
Exclude any other similar globals without asking again? Yes => Yes
^oddCOM      contains no data --  not included
^oddDEP      contains no data --  not included
^oddEXT      contains no data --  not included
^oddEXTR     contains no data --  not included
^oddMAP      contains no data --  not included
^oddPKG      contains no data --  not included
^oddPROC     contains no data --  not included
^oddPROJECT  contains no data --  not included
^oddSQL      contains no data --  not included
^oddStudioDocument contains no data --  not included
^oddStudioMenu contains no data --  not included
^oddTSQL     contains no data --  not included
^oddXML      contains no data --  not included
^rBACKUP     contains no data --  not included
^rINC        contains no data --  not included
^rINCSAVE    contains no data --  not included
^rINDEXEXT   contains no data --  not included
^rINDEXSQL   contains no data --  not included
^rMACSAVE    contains no data --  not included
9 items selected from
29 available globals
```

i. **^GBLOCKCOPY** then asks if you wish to disable journaling for this operation:

```
Turn journaling off for this copy? Yes =>
```

Enter `Yes` (can be `Yes`, `Y`, `y`, and so on), which is the default.

j. **^GBLOCKCOPY** then asks if to confirm that you wish to copy the data:

```
Confirm copy? Yes =>
```

Enter `Yes` (can be `Yes`, `Y`, `y`, and so on), which is the default. Depending on the size of the database and the speed of the processor, you may see the status of the copy as it progresses. When it completes, **^GBLOCKCOPY** displays a message such as:

```
Copy of data has completed
```

k. **^GBLOCKCOPY** then asks if you wish to save statistics associated with the copy. Enter `No` (can be `No`, `N`, `n`, and so on), which is the default:

```
Do you want to save statistics for later review? No =>
```

Control then returns to the main prompt.

5. Test that the copied data is valid. You can do this by examining the classes, tables, or globals in the Management Portal’s System Explorer for the database into which **^GBLOCKCOPY** has copied the data.
6. If the data is valid, perform steps 3 and 4 of this procedure for each database encrypted with the inaccessible or damaged key.
7. Once you have made copies of every encrypted database into an unencrypted database, make a second copy of each database, preferably to a different machine than that which holds the first copy of each.
8. Now — *and only now* — you can dismount all encrypted databases and deactivate the active key (that is, the key for which the key file is missing or damaged). InterSystems IRIS requires that you dismount all encrypted databases prior to deactivating their key.

You now have your data in one or more unencrypted databases and there is no activated key.

To re-encrypt the formerly encrypted databases, the procedure is:

1. Create a new database encryption key according to the procedure described in Creating a Key.

2. Create a new backup copy of the key file as described in Protection from Accidental Loss of Access to Encrypted Data.

   CAUTION: Make sure you take the precautions described in Protection from Accidental Loss of Access to Encrypted Data; failure to follow these procedures can result in the permanent loss of data.

3. Create one or more new encrypted databases, using the new key.
4. Import the data exported in the previous procedure into the new encrypted database(s).

Your data is now stored in encrypted databases for which you have a valid key and a backup copy of the key file containing that key.

6.1.2 If the Database Encryption Key File Is Required at Startup and Is Not Present

Under certain conditions related to the required use of a database encryption key file at startup, the system starts in single-user mode. These conditions are:

- InterSystems IRIS is configured for either interactive or unattended startup.
- Startup specifies that journal files and/or the IRISTEMP and IRISLOCALDATA databases are encrypted, or an encrypted database is specified as required at startup.
- The database encryption key file is not present.

6.1.2.1 If You Can Make the Key File Available

This situation may have been caused simply by the appropriate key file not being present at InterSystems IRIS startup time — such as if the media holding it is not currently available.

To correct the condition, after InterSystems IRIS starts running in single-user mode, then the procedure is:

1. Shut down InterSystems IRIS. For example, if the instance of InterSystems IRIS is called “MyIRIS”, the command to do this would be:

   ```
   iris force MyIRIS
   ```

2. If you know the location where InterSystems IRIS is expecting to find the database encryption key file, then place the key file there. (Otherwise, you need to run **^STURECOV** as specified in the next section.)
3. Start InterSystems IRIS again.

InterSystems IRIS should start in its typical mode (multi-user mode) and operate as expected.

6.1.2.2 If a Backup Key File Is Available

If the appropriate key file is not present at InterSystems IRIS startup time and is not available, you may have a backup key file available. If so, then to correct the condition, after InterSystems IRIS starts running in single-user mode, then the procedure is:

1. Contact InterSystems Worldwide Response Center. Engineers there can help guide you through the following procedure and answer any questions that may arise.
2. Start a Administrator Terminal Session according to the instructions in the most recent entry in the messages.log file. Typically, this specifies starting a Terminal session with the `-B` flag..

For example, at a Windows command line, for an instance of InterSystems IRIS called “MyIRIS” that is installed in the default location, the command would be:

```
c:\InterSystems\MyIRIS\bin\irisdb -sc:\InterSystems\MyIRIS\mgr -B
```

This connects you with InterSystems IRIS in the operating system terminal window; the prompt in that window changes from the operating system prompt to the InterSystems IRIS `%SYS` prompt.

3. If you have or can obtain a copy of the database encryption key file (such as a backup), then place a copy of the key file in a location accessible to InterSystems IRIS.
4. Run the **^STURECOV** (startup recovery) routine at the Terminal prompt. In that routine, activate the encryption key using an administrator username and password in that file. (You do not need to exit **^STURECOV** when you have completed this process.)
5. When you are satisfied that InterSystems IRIS is ready for use, use **^STURECOV** to complete the startup procedure. InterSystems IRIS then starts in multi-user mode.

InterSystems IRIS should now operate as expected.

6.1.2.3 If No Key File Is Available

If you do not have any copy of the database encryption key file, then the procedure is:

1. Contact InterSystems Worldwide Response Center (WRC). Engineers there can help guide you through the following procedure and answer any questions that may arise.
2. Start a Administrator Terminal Session according to the instructions in the most recent entry in the messages.log file. Typically, this specifies starting a Terminal session with the `-B` flag..

   For example, at a Windows command line, for an instance of InterSystems IRIS called “MyIRIS” that is installed in the default location, the command would be:

   ```
   c:\InterSystems\MyIRIS\bin\irisdb -sc:\InterSystems\MyIRIS\mgr -B
   ```

   This connects you with InterSystems IRIS in the operating system terminal window; the prompt in that window changes from the operating system prompt to the InterSystems IRIS `%SYS` prompt.
3. If any encrypted databases require mounting at startup, disable this feature for them:
   a. From the Management Portal Home page, go to the **Local Databases** page (**System Administration** > **Configuration** > **System Configuration** > **Local Databases**).
   b. Click the name of the database in the table of databases. This displays the **Edit:** page for the database.
   c. On the **Edit:** page, clear the **Mount Required at Startup** check box.
   d. Click **Save**.
4. Run the **^STURECOV** routine at the Terminal prompt. In that routine, configure InterSystems IRIS database startup options not to require a database encryption key. This means that the IRISTEMP and IRISLOCALDATA databases as well as journal files should now operate as expected; it also means that any encrypted databases cannot be mounted.
5. When you are satisfied that InterSystems IRIS is ready for use, use **^STURECOV** to complete the startup procedure. InterSystems IRIS then starts in multi-user mode.

As you perform this procedure, you may need to perform other actions, according to the instructions of the representative from the WRC. Follow these instructions.

CAUTION: If you have not performed the actions described in Protection from Accidental Loss of Access to Encrypted Data, then your data may no longer be available in any form. This is a very serious problem, but if you do not have a key, there is no way to retrieve the lost data.

6.2 Handle Emergency Situations When Using a KMIP Server

This topic describes what to do under certain circumstances when you are using a KMIP server and are in danger of losing data. These situations include:

- If the KMIP Server Holding an Activated Key is Damaged or Missing
  - If There Is a Backup Copy of the Key on the KMIP Server
  - If There Is No Backup Copy of the Key on the KMIP Server

    WARNING! This is a *dire* situation. Act *immediately*.

- If the KMIP Server Is Required at Startup and Is Not Accessible
  - If the Connection to the KMIP Server is Briefly Unavailable
  - If the KMIP Server Suffers a Longer-Term Outage

6.2.1 If the KMIP Server Holding an Activated Key is Damaged or Missing

In this situation, the following circumstances have occurred:

- A database encryption key has been activated for the InterSystems IRIS instance
- InterSystems IRIS is using encrypted data
- The KMIP server the database encryption key becomes damaged

6.2.1.1 If There Is a Backup Copy of the Key on the KMIP Server

If KMIP server holding an activated key becomes inaccessible or damaged, *immediately* perform restore procedures for the KMIP server according to your vendor’s instructions.

6.2.1.2 If There Is No Backup Copy of the Key on the KMIP Server

WARNING! THIS PROCEDURE IS FOR AN EMERGENCY SITUATION, WHERE ENCRYPTED DATA IN INTERSYSTEMS IRIS DATABASES IS IN DANGER OF BEING LOST.

If there is no way to restore the KMIP server holding the activated key from backup, *immediately* perform the following procedure for each database encrypted with that key:

1. **WARNING!** Shutting down InterSystems IRIS or deactivating the active key will cause the permanent loss of your data.

   Do not shut down InterSystems IRIS.

   Do not deactivate the currently active key.

2. Contact the InterSystems Worldwide Response Center. Engineers there can help guide you through the following procedure and answer any questions that may arise.
3. Dismount the database. This prevents all users from making any changes to the database with encrypted content while copying its data to an unencrypted database:
   a. From the Management Portal home page, go to the **Databases** page (**System Operation** > **Databases**).
   b. On the **Databases** page, if the encrypted database is mounted, select the **Dismount** option in the next-to-last column in that database's row. Then select **OK** in the confirmation dialog.
   c. When the **Databases** page appears again, select the **Mount** option in the last column in the database's row.
   d. On the **Mount database** confirmation screen, check the **Read Only** box and select **OK**.

   It is critical that no one makes any changes to the database during this procedure. Mounting the database read-only prevents any user from changing any data.

4. Copy all data in unencrypted form to another database. The procedure for copying the data is:
   a. In the Terminal, go to the %SYS namespace:

   ```
   REGULARNAMESPACE>set $namespace="%SYS"
   ```

   b. From that namespace, run the **^GBLOCKCOPY** command:

   ```
   %SYS>do ^GBLOCKCOPY

   This routine will do a fast global copy from a database to another database or
   to a namespace. If a namespace is the destination, the global will follow any
   mappings set up for the namespace.

   1) Interactive copy
   2) Batch copy
   3) Exit

   Option?1
   ```

   c. At the **^GBLOCKCOPY** prompt, specify 1, for an interactive copy:

   ```
   Option? 1

   1) Copy from Database to Database
   2) Copy from Database to Namespace
   3) Exit

   Option?
   ```

   d. When **^GBLOCKCOPY** prompts for a copy type, select 1, for copying from database to database

   ```
   Option? 1
   Source Directory for Copy (? for List)?
   ```

Here, either specify the name of the encrypted database or enter ? to see a numbered list of databases, which includes the encrypted database. If you enter ?, **^GBLOCKCOPY** displays a list such as this one:

```
Source Directory for Copy (? for List)? ?

1) C:\InterSystems\MyIRIS\mgr\
2) C:\InterSystems\MyIRIS\mgr\irislocaldata\
3) C:\InterSystems\MyIRIS\mgr\irisaudit\
4) C:\InterSystems\MyIRIS\mgr\irislib\
5) C:\InterSystems\MyIRIS\mgr\iristemp\
6) C:\InterSystems\MyIRIS\mgr\encrypted1\
7) C:\InterSystems\MyIRIS\mgr\encrypted2\
8) C:\InterSystems\MyIRIS\mgr\unencrypted\

Source Directory for Copy (? for List)?
```

Enter the number of the encrypted database, such as 7 here.

e. When **^GBLOCKCOPY** prompts for a destination directory for copying the data, enter the name of an unencrypted database or ? for a list similar to the one for the source directory.

f. When **^GBLOCKCOPY** asks if you wish to copy all globals, enter `Yes` (can be `Yes`, `Y`, `y`, and so on):

```
All Globals? No => y
```

g. If there is an empty global in the database, **^GBLOCKCOPY** will now ask if you wish to copy it. This will appear something like the following:

```
All Globals? No => y

^oddBIND     contains no data
Include it anyway? No =>
```

Enter `No` (can be `No`, `N`, `n`, and so on), which is the default.

h. **^GBLOCKCOPY** then asks if you wish to skip all the other empty globals. Enter `Yes` (can be `Yes`, `Y`, `y`, and so on), which is the default:

```
Exclude any other similar globals without asking again? Yes =>
```

There then appears a list of all the empty globals that are not being copied:

```
Exclude any other similar globals without asking again? Yes => Yes
^oddCOM      contains no data --  not included
^oddDEP      contains no data --  not included
^oddEXT      contains no data --  not included
^oddEXTR     contains no data --  not included
^oddMAP      contains no data --  not included
^oddPKG      contains no data --  not included
^oddPROC     contains no data --  not included
^oddPROJECT  contains no data --  not included
^oddSQL      contains no data --  not included
^oddStudioDocument contains no data --  not included
^oddStudioMenu contains no data --  not included
^oddTSQL     contains no data --  not included
^oddXML      contains no data --  not included
^rBACKUP     contains no data --  not included
^rINC        contains no data --  not included
^rINCSAVE    contains no data --  not included
^rINDEXEXT   contains no data --  not included
^rINDEXSQL   contains no data --  not included
^rMACSAVE    contains no data --  not included
9 items selected from
29 available globals
```

i. **^GBLOCKCOPY** then asks if you wish to disable journaling for this operation:

```
Turn journaling off for this copy? Yes =>
```

Enter `Yes` (can be `Yes`, `Y`, `y`, and so on), which is the default.

   j. **^GBLOCKCOPY** then asks if to confirm that you wish to copy the data:

      ```
      Confirm copy? Yes =>
      ```

      Enter `Yes` (can be `Yes`, `Y`, `y`, and so on), which is the default. Depending on the size of the database and the speed of the processor, you may see the status of the copy as it progresses. When it completes, **^GBLOCKCOPY** displays a message such as:

      ```
      Copy of data has completed
      ```

   k. **^GBLOCKCOPY** then asks if you wish to save statistics associated with the copy. Enter `No` (can be `No`, `N`, `n`, and so on), which is the default:

      ```
      Do you want to save statistics for later review? No =>
      ```

      Control then returns to the main prompt.

5. Test that the copied data is valid. You can do this by examining the classes, tables, or globals in the Management Portal’s System Explorer for the database into which **^GBLOCKCOPY** has copied the data.
6. If the data is valid, perform steps 3 and 4 of this procedure for each database encrypted with the inaccessible or damaged key.
7. Once you have made copies of every encrypted database into an unencrypted database, make a second copy of each database, preferably to a different machine than that which holds the first copy of each.
8. Now — *and only now* — you can dismount all encrypted databases and deactivate the active key (that is, the key for which the key file is missing or damaged). InterSystems IRIS requires that you dismount all encrypted databases prior to deactivating their key.

You now have your data in one or more unencrypted databases and there is no activated key.

To re-encrypt the formerly encrypted databases, the procedure is:

1. Create a new database encryption key according to the procedure described in Create a Key on the KMIP Server.
2. Create a new backup copy of the key file as described in Protection from Accidental Loss of Access to Encrypted Data.

   CAUTION: Make sure you take the precautions described in Protection from Accidental Loss of Access to Encrypted Data; failure to follow these procedures can result in the permanent loss of data.

3. Create one or more new encrypted databases, using the new key.
4. Import the data exported in the previous procedure into the new encrypted database(s).

Your data is now stored in encrypted databases for which you have a valid key and a backup copy of the key file containing that key.

6.2.2 If the KMIP Server Is Required at Startup and Is Not Accessible

Under certain conditions related to the required use of one or more database encryption keys at startup, the system starts in single-user mode. These conditions are:

- InterSystems IRIS is configured for either interactive or unattended startup.
- Startup specifies that journal files and/or the IRISTEMP and IRISLOCALDATA databases are encrypted, or an encrypted database is specified as required at startup.
- The KMIP server that holds the required database encryption key is not accessible.

6.2.2.1 If the Connection to the KMIP Server is Briefly Unavailable

The simplest solution to this case is when there has been a problem such as a network outage or the KMIP server is otherwise temporarily not running; in these cases, address networking or server problem and, if required, restart InterSystems IRIS.

6.2.2.2 If the KMIP Server Suffers a Longer-Term Outage

If it is not possible to connect to the KMIP server longer-term:

1. Start a Administrator Terminal Session according to the instructions in the most recent entry in the messages.log file. Typically, this specifies starting a Terminal session with the `-B` flag..

   For example, at a Windows command line, for an instance of InterSystems IRIS called "MyIRIS" that is installed in the default location, the command would be:

   ```
   c:\InterSystems\MyIRIS\bin\irisdb -sc:\InterSystems\MyIRIS\mgr -B
   ```

   This connects you with InterSystems IRIS in the operating system terminal window; the prompt in that window changes from the operating system prompt to the InterSystems IRIS `%SYS` prompt.

2. If any encrypted databases require mounting at startup, disable this feature for them:

   a. From the Management Portal Home page, go to the **Local Databases** page (**System Administration** > **Configuration** > **System Configuration** > **Local Databases**).

   b. Click the name of the database in the table of databases. This displays the **Edit:** page for the database.

   c. On the **Edit:** page, clear the **Mount Required at Startup** check box.

   d. Click **Save**.

3. Run the **^STURECOV** routine at the Terminal prompt. In that routine, configure InterSystems IRIS database startup options not to require a database encryption key. This means that the IRISTEMP and IRISLOCALDATA databases as well as journal files should now operate as expected; it also means that any encrypted databases cannot be mounted.

4. When you are satisfied that InterSystems IRIS is ready for use, use **^STURECOV** to complete the startup procedure. InterSystems IRIS then starts in multi-user mode.

CAUTION: If you have not performed the actions described in Protection from Accidental Loss of Access to Encrypted Data, then your data may no longer be available in any form. This is a very serious problem, but, if you do not have a key, there is no way to retrieve the lost data.

7

Additional Encryption Information

This topic addresses additional information about InterSystems IRIS® data platform encryption.

7.1 Key File Encryption Information

Database encryption administrator names are stored in the clear in the key file. Database encryption administrator passwords are not stored; when entered, they are used, along with other data, to derive key-encryption keys. If someone can successfully guess a valid password, the password policy is too weak. Key-encryption keys are derived using the PBKDF2 algorithm with 512 bits of salt and 65,536 iterations, making dictionary and brute force attacks impractical.

7.2 Encryption and Database-Related Facilities

InterSystems IRIS database encryption protects database files themselves. Regarding related facilities in InterSystems IRIS:

- InterSystems IRIS online backups are not encrypted. To ensure that the InterSystems IRIS database is encrypted in a backup, it is recommended that you quiesce InterSystems IRIS and then perform a file system backup (as described in External Backup).
- In the write image journal (WIJ) file, the blocks for encrypted databases are encrypted.
- The IRISTEMP and IRISLOCALDATA databases can optionally be encrypted. To provide encryption for IRISTEMP and IRISLOCALDATA, see Configure Encryption Startup Settings.
- You can optionally encrypt journal files; see Configuring Database Encryption Settings.

7.3 About Calls to Perform Encryption, Hashing, and Other Key-Related Operations

InterSystems IRIS allows you to perform actions related to data encryption, Base64 encoding, hashing, and generating message authentication codes using various methods of the %SYSTEM.Encryption class. It includes methods that invoke AES encryption, various RSA algorithms, SHA-256 hash functions, and more. Some of the calls include:

- **$System.Encryption.AESCBCManagedKeyEncrypt** and **$System.Encryption.AESCBCManagedKeyDecrypt**
- **$System.Encryption.AESKeyWrap** and **$System.Encryption.AESKeyUnwrap**
- **$System.Encryption.Base64Encode** and **$System.Encryption.Base64Decode**
- **$System.Encryption.RSASHASign** and **$System.Encryption.RSASHAVerify**
- **$System.Encryption.RSAEncrypt** and **$System.Encryption.RSADecrypt**
- **$System.Encryption.SHAHash**

7.3.1 An Example of Using RSAEncrypt and RSADecrypt

Below is an example of using the **RSAEncrypt** and **RSADecrypt** calls. It assumes that:

- The code is running on Windows.
- There is an available certificate, private key, and certificate authority (CA) certificate. (To try this example, you will need to obtain these.)
- All three of these items are in the C:\Keys\ directory.

See the comments within the example for more details of its operations.

ObjectScript

```
set dir = "C:\Keys\"

// certificate for the instance performing encryption and decryption
// and private key associated with that above certificate
set cert = dir_"test.crt"
set key = dir_"test.key"

// certificate for the CA of the instance
set cacert=dir_"ca.crt"

set data = "data to be encrypted"

// create a local set of X.509 credentials with the
// certificate and private key
set credentials = ##class(%SYS.X509Credentials).%New()
set credentials.Alias="TestCreds"
write credentials.LoadCertificate(cert)
write credentials.LoadPrivateKey(key)
write credentials.Save(),!

// encrypt the data using the public key in the certificate, write it
// to the display, and display error information, if there is any
set ciphertext=$System.Encryption.RSAEncrypt(data,credentials.Certificate,cacert)
write ciphertext,!
write $System.Encryption.RSASHA1GetLastError()

// decrypt the data using the private key, write it to the display,
// and display error information, if there is any
write "now decrypting -=-=-=-=-=-=-=-=-=-=-",!
set cleartext=$System.Encryption.RSADecrypt(ciphertext,credentials.PrivateKey)
write cleartext,!
write $System.Encryption.RSASHA1GetLastError()
```

A

FIPS 140–2 Compliance for Database Encryption

On specific platforms, InterSystems IRIS® data platform supports FIPS 140–2 compliant cryptography for database encryption. (FIPS 140–2 refers to Federal Information Processing Standard Publication 140-2, which is available at https://csrc.nist.gov/csrc/media/publications/fips/140/2/final/documents/fips1402.pdf.)

This version of InterSystems IRIS supports FIPS 140-2–compliant cryptography for database encryption on Red Hat Enterprise Linux 8 for x86-64. Red Hat has certificates of validation for the OpenSSL libcrypto.so and libssl.so libraries. When running in FIPS mode, InterSystems IRIS uses these certified libraries. To determine if a minor version of Red Hat Linux has current certification, consult the Red Hat documentation.

Note: With FIPS mode enabled:

- With FIPS mode enabled, Red Hat 8 supports only TLSv1.2 and TLSv1.3.
- InterSystems IRIS does not support FIPS mode on Red Hat 9.

For information about Red Hat support for government standards, see https://access.redhat.com/articles/2918071.

A.1 Enabling FIPS Support

To enable InterSystems IRIS support for FIPS 140–2 compliant cryptography for database encryption, do the following:

1. Download and install the openssl package from the RedHat repository (rhel-8-server-rpms).
2. Enable FIPS mode for the operating system. For these instructions, see the article How can I make RHEL 6/7/8 FIPS 140-2 compliant? on the Red Hat web site. (Access to this article requires Red Hat login credentials.)
3. Check the directory /usr/lib64 for the following symbolic links. If these do not exist, create them:
   - The symbolic link libssl.so.1.1 should point to the appropriate file (such as libssl.so.1.1.1g), in the same directory.
   - The symbolic link libcrypto.so.1.1 should point to the appropriate file (such as libcrypto.so.1.1.1g), in the same directory.
4. In InterSystems IRIS, specify the **FIPSMode** CPF parameter as **True** (1). To do so:
   a. Open the Management Portal.
   b. Select **System Administration** > **Configuration** > **Additional Settings** > **Startup**.

Here you will see a row for **FIPSMode**.

c. Specify the value for **FIPSMode** as **True** and save your change.

5. Restart InterSystems IRIS.
6. Enable and configure encrypted databases as outlined in Using Encrypted Databases.

A.2 Startup Behavior and messages.log

When InterSystems IRIS is started:

- If **FIPSMode** is 0, InterSystems IRIS native cryptography is used, including optimized assembly code using Intel AES-NI hardware instructions, if supported by the CPU. In this mode, InterSystems IRIS writes the following to messages.log upon startup:

  ```
  FIPS 140-2 compliant cryptography for database encryption is not configured in iris.cpf
  ```

- If **FIPSMode** is 1, InterSystems IRIS attempts to resolve references to functions in the /usr/lib64/libcrypto.so FIPS-validated library, and then attempts to initialize the library in FIPS mode. If these steps are successful, InterSystems IRIS writes the following to messages.log:

  ```
  FIPS 140-2 compliant cryptography for database encryption is enabled for this instance.
  ```

- If **FIPSMode** is 1, but the initialization of the library is unsuccessful, InterSystems IRIS does not start. In this case, messages.log contains the following message:

  ```
  FIPS 140-2 compliant cryptography for database encryption initialization failed. Aborting.
  ```

- On platforms other than lnxrhx64, if **FIPSMode** is 1, InterSystems IRIS native cryptography is used, and InterSystems IRIS writes the following to messages.log:

  ```
  FIPS 140-2 compliant cryptography for database encryption is not supported on this platform.
  ```