First Look: Database Encryption

Version 2019.4
2020-01-28

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

First Look: Database Encryption
InterSystems IRIS Data Platform Version 2019.4 2020-01-28

InterSystems Corporation disclaims responsibility for errors which may appear in this document, and it reserves the right, in its sole discretion and without notice, to make substitutions and modifications in the products and practices described in this document.

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

First Look: Database Encryption

This First Look introduces you to how InterSystems IRIS® data platform handles database encryption, which is an important part of any organization's security strategy.

This First Look presents an introduction to database encryption and walks you through some initial tasks associated with creating an encrypted database. Once you've completed this guide, you will have created a key file, activated the key file, and then used it to encrypt a database. These activities are designed to use only the default settings and features, so that you can acquaint yourself with the fundamentals of the feature without having to deal with details that are off the topic (though these may be important when performing an implementation). For the full documentation on database encryption, see "Managed Key Encryption" in the *InterSystems IRIS Security Administration Guide*.

To browse all of the First Looks, including those that can be performed on a free evaluation instance of InterSystems IRIS, see InterSystems First Looks.

1 Why Database Encryption Is Important

While encryption does not prevent all improper or unauthorized use or disclosure of confidential or personal information, ensuring encryption of data at rest provides an important layer in the defense of the security of information. Putting encryption in place at the database level provides an added dimension to your information protection controls.

Additionally, many laws and regulations regarding sensitive or personal information recommend or require that the organization processing the data employ encryption as a first line of defense. These include laws and regulations such as:

- Health Insurance Portability and Accountability Act (HIPAA) — Requirement that Secure Protected Health information be unreadable, undecipherable, and unrecoverable
- Massachusetts 201 Code of Massachusetts Regulations (CMR) 17.00 — Requirement that personal information be encrypted in transit and at rest
- New York 23 New York Codes, Rules and Regulations (NYCRR) Part 500 — Financial and other covered organizations processing nonpublic information must utilize encryption as one means of safeguarding data
- European Union General Data Protection Regulation (GDPR) — Requirement for security safeguards to take into account encryption as a protecting control
- Italian Personal Data Protection Code (PDPC) — Section 24 of the Technical Specifications on Minimum Data Security Measures requires the processing of data disclosing health and sex life to be encrypted
- Australian Privacy Principles (APP) Principle 4 — Robust encryption implementation addresses necessary privacy enhancing technologies to secure personal information
- Japan Ministry of Economy, Trade, and Industry (METI) Guidelines — Regulatory investigation must be undertaken if compromise of personal or confidential information that was not encrypted, because under the Act on the Protection of Personal Information (APPI), Art. 20, the processor of personal information obligated to prevent leakage, loss, or damage of information

Note that many of these legal requirements focus on data breaches, as they are an increasingly common phenomenon, but the current framework obligates organizations to address risk through proper security controls, such as role-based access, password protections, intrusion detection, data loss prevention, and logging/auditing — as well as encryption. Encryption alone will not address all mandatory requirements, but provides a secure foundation. Encryption at the database level enhances protections by requiring an attacker to not only gain access to the system or file space, but to also have access to the database. This additional layer provides assurances to the organization, its customers, and any stakeholders.

2 How InterSystems IRIS Uses Database Encryption

For activities associated with database operations, the InterSystems IRIS encryption and decryption processes are transparent to users. From the perspective of the end user or the application developer, the application simply performs its usual activities and the data is automatically encrypted on disk. From the perspective of the system administrator, there are a few simple tasks to ensure that data encryption occurs; after performing these tasks, again, activities occur invisibly.

What's more, these activities use a minimum of processor time, so they are likely to have no visible impact on your applications. Further, because of how our databases are constructed, these activities are highly optimized.

Encryption and decryption use the United States Government Advanced Encryption Standard (AES) in Cipher Block Chaining (CBC) mode, often simply known as AES CBC. InterSystems IRIS supports all legal key sizes for AES CBC: 128–, 192–, and 256–bit keys.

InterSystems IRIS performs encryption and decryption using the fastest available implementation. Whenever available, encryption and decryption take advantage of the processor-based instruction sets and their inherent efficiencies. Modern Intel and IBM POWER8 processors have such instructions. InterSystems IRIS automatically detects and uses these instructions, so you don't have take any actions to make it happen. On Intel processors, these are the Advanced Encryption Standard New Instructions (AES-NI); on IBM, they are the AES VMX instruction set.

You can store database encryption keys either on key management servers that support the key management interoperability protocol (KMIP) or in files that contain encrypted copies of the database keys. Each has its own advantages:

- KMIP is an OASIS standard protocol for clients to communicate with key management systems. KMIP servers can be specialized hardware appliances or general-purpose servers running key management software.
- When database encryption keys are stored in files, InterSystems encrypts the keys using multiple layers of the AES key wrap algorithm, with individual administrator key-encryption keys derived using the PBKDF2 algorithm, thereby making dictionary and brute force attacks impractical.

It is important to keep in mind that, while database encryption is an integral part of a security strategy, it cannot address security vulnerabilities alone. Other tools, such as protection for data in motion, are also crucial. This is why database encryption is part of the suite of tools that InterSystems IRIS provides for protecting data. These include:

- Support for government standards — You can configure InterSystems IRIS to use libraries that are validated to conform to FIPS 140–2 Federal Information Processing Standards) for database encryption. This is available on Red Hat Linux.
- Protecting selected data elements — Known as data-element element encryption, this feature provides a programmatic approach that allows you to encrypt only selected parts of records, such credit card or Social Security numbers
- Protecting data in motion — InterSystems IRIS protects data in motion using the newest version of Transport-Layer Security (TLS). TLS, the industry standard protocol for protecting data communications, is the successor of the Secure Sockets Layer (SSL).
- Support for third-party authorization — InterSystems IRIS supports authorization for using resources on third-party sites, as is frequently seen on the web with logins through Facebook or Google to use a site. This is through the Open Authorization Framework version 2.0 (known as OAuth 2.0) and may include authentication through another layer, known as OpenID Connect.

3 Trying Database Encryption for Yourself

It's easy to use InterSystems IRIS database encryption. This simple procedure walks you through the basic steps of setting up an encrypted database.

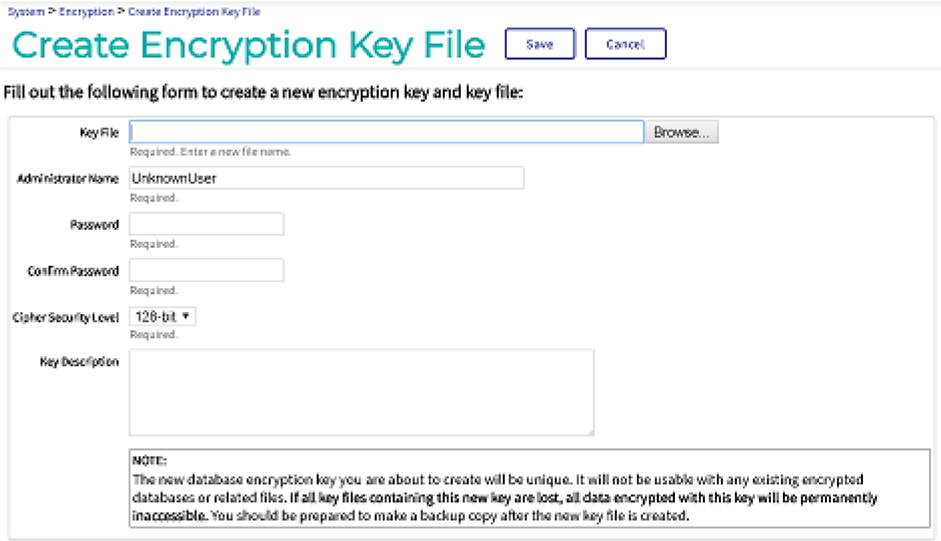
3.1 Before You Begin

To use the procedure, you will need a running instance of InterSystems IRIS. Your choices include several types of licensed and free evaluation instances. For information on how to deploy each type of instance if you do not already have one to work with, see Deploying InterSystems IRIS in *InterSystems IRIS Basics: Connecting an IDE*.

3.2 Creating an Encryption Key

First, create a key file, which automatically has a database encryption key in it:

1. Open the Management Portal for your instance in your browser, using the URL described for your instance in *InterSystems IRIS Basics: Connecting an IDE*.
2. Navigate to the **Create Encryption Key File** page (**System Administration** > **Encryption** > **Create New Encryption Key File**):


System > Encryption > Create Encryption Key File
Create Encryption Key File
Save
Cancel
Browse...

Fill out the following form to create a new encryption key and key file:

Key File: Required. Enter a new file name.

Administrator Name: UnknownUser
Required.

Password: Required.

Confirm Password: Required.

Cipher Security Level: 128-bit
Required.

Key Description:

NOTE:
The new database encryption key you are about to create will be unique. It will not be usable with any existing encrypted databases or related files. **If all key files containing this new key are lost, all data encrypted with this key will be permanently inaccessible.** You should be prepared to make a backup copy after the new key file is created.

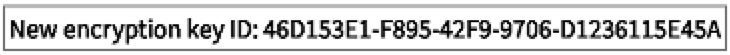
3. On the **Create Encryption Key File** page:

   a. In the **Key File** field, enter a name and path for the key file. When you click the **Browse** button, the File Selection Dialog opens by default in the instance's *install-dir*/mgr directory (where *install-dir* is the InterSystems IRIS installation directory), for example, C:\InterSystems\IRIS\mgr\testkeys.key; you can use this directory on all types of instance, or select another location in the host or container file system.

   b. In the **Administrator Name**, **Password**, and **Confirm Password** fields, enter values such as `testadmin` and `password`. This is just an example case, so don't reuse a password that you would use in a development environment.

   c. Select the **Save** button near the top of the page.

You just created the testkeys.key key file in the C:\InterSystems\directory with a key in it that you can use for database encryption. InterSystems IRIS displays a message with the key in it, such as



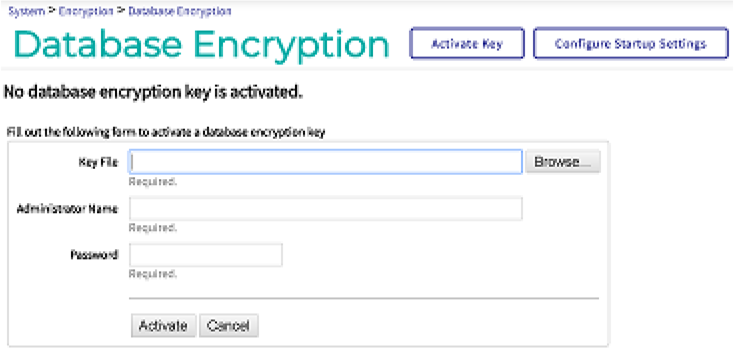
New encryption key ID: 46D153E1-F895-42F9-9706-D1236115E45A

For more details about creating a key file and its initial key, see “Creating a Key File” in the “Managed Key Encryption” chapter of the *InterSystems IRIS Security Administration Guide*.

3.3 Activating an Encryption Key

Next, activate the key that you just created:

1. In the Management Portal, go to the **Database Encryption** page (**System Administration** > **Encryption** > **Database Encryption**).
2. On the Database Encryption page, select the **Activate Key** button:

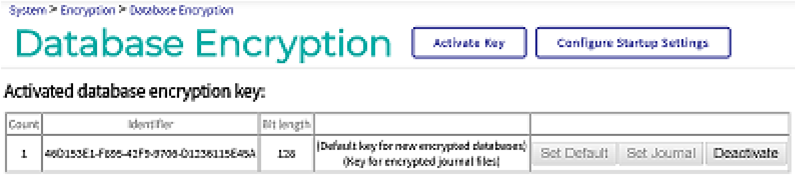


No database encryption key is activated.

Fill out the following form to activate a database encryption key

3. In the **Key File** field, enter the location of the key file you created, such as C:\InterSystems\IRIS\mgr\testkeys.key.
4. In the **Administration Name** and **Password** fields, enter the values you specified (`testadmin` and `password`).
5. Select the **Activate** button.

You can see the key ID on this page:



System > Encryption > Database Encryption
Database Encryption
Activate Key
Configure Startup Settings

Activated database encryption key:

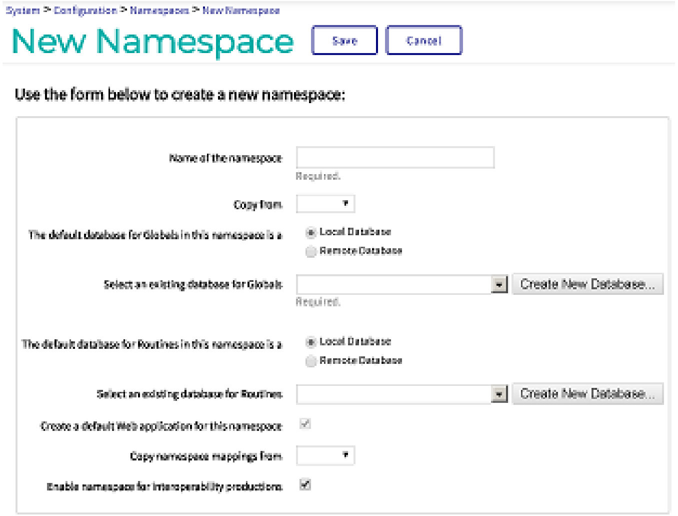
Count	Identifier	Bit length				
1	46D153E1-F895-42F9-9706-D1236115E45A	128	(Default key for new encrypted databases) (Key for encrypted journal files)	Set Default	Set Journal	Deactivate

For more details about activating a key, see “Activating a Database Encryption Key from a Key File” in the “Managed Key Encryption” chapter of the *InterSystems IRIS Security Administration Guide*.

3.4 Creating an Encrypted Database

Now, you can create an encrypted database:

1. Again in the Management Portal, go to the **Namespaces** page (**System Administration** > **Configuration** > **System Configuration** > **Namespaces**).
2. On the **Namespaces** page, select **Create New Namespace**. This displays the **New Namespace** page:

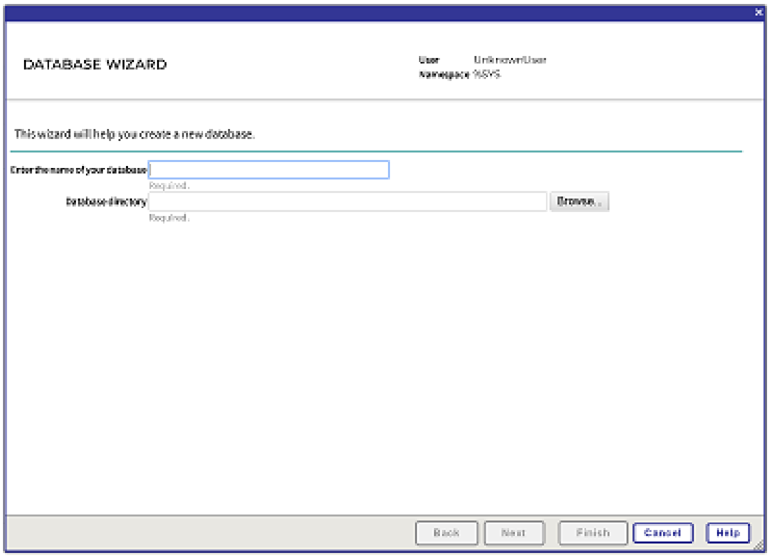


Use the form below to create a new namespace:

Name of the namespace
Required.
Copy from
The default database for Globals in this namespace is a ● Local Database ○ Remote Database
Select an existing database for Globals
Required.
The default database for Routines in this namespace is a ● Local Database ○ Remote Database
Select an existing database for Routines
Create a default Web application for this namespace ☑
Copy namespace mappings from
Enable namespace for interoperability productions ☑

3. On the **New Namespace** page, enter the name of the encrypted database that you are going to create, such as **ENCDB**.

4. Next to the **Select an existing database for Globals** drop-down menu, select the **Create New Database** button. This displays the **Database Wizard**:



User UnknownUser
Namespace %SYS

This wizard will help you create a new database.

Enter the name of your database
Required.
Database directory
Required.

5. On the first page of the **Database Wizard**, in the **Enter the name of your database** field, enter the name of the database you are creating, such as **ENCDB**. Enter a directory for the database, such as C:\InterSystems\IRIS\mgr\ENCDB. On that page, select **Next**.

6. On the next page, change the value of **Encrypt database** from `No` to `Yes`. On that page, select **Finish**.

7. Back on the **New Namespace** page, in the **Select an existing database for Routines** drop-down menu, select the database you just created, such as **ENCDB**.

8. Select the **Save** button near the top of the page and then select **Close** at the end of the resulting log.

You have now created an encrypted database called **ENCDB** that uses the key that InterSystems IRIS created when you created the key file. You can use this database just as you would use an unencrypted database. Because InterSystems IRIS hides all the machinery for encryption and decryption, you can perform all operations in the usual way and all your data will be encrypted.

For more details about creating a namespace and its associated database, see “Create/Modify a Namespace” in the “Configuring InterSystems IRIS” chapter of the *InterSystems IRIS System Administration Guide*. For background information, see “Namespaces and Databases” in the *Orientation Guide for Server-Side Programming*.

3.5 Looking at Encrypted Data

Once you have created the encrypted database, you can use it just as you would use any other, unencrypted database. The only difference is how the data is stored. To see the difference between data stored in encrypted and unencrypted databases, you can perform the following, simple test:

1. Open the Terminal for your InterSystems IRIS instance, using the procedure described for your instance in *InterSystems IRIS Basics: Connecting an IDE*.

2. Switch to the namespace for the encrypted database using the following command:

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

3. In the **ENCDB** namespace, run the following command:

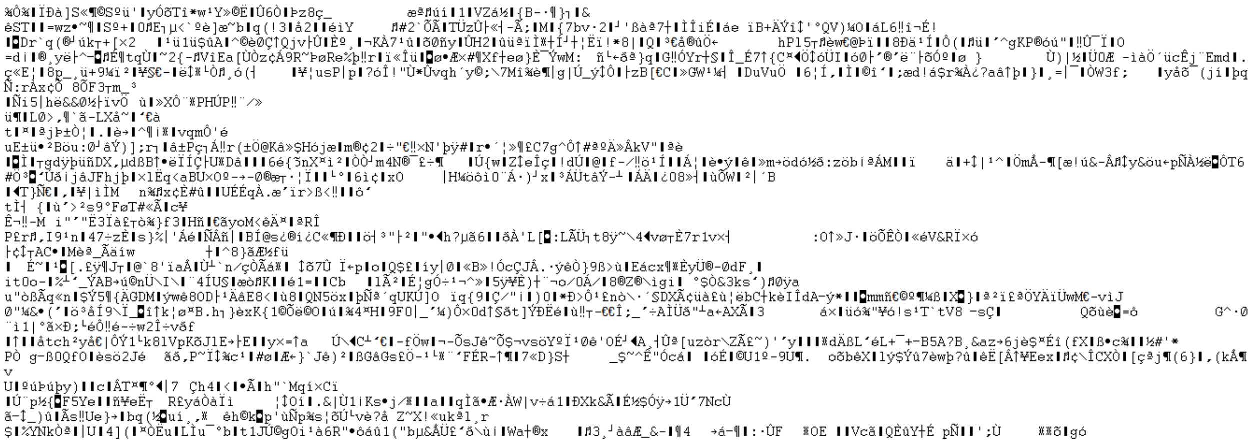
```
ENCDB>for i=1:1:1000 set ^x(i)="This is test number "_i
```

   This creates a thousand persistent variables with content such as `This is test number 22`.

4. To confirm that you have succeeded, look at the value of one variable:

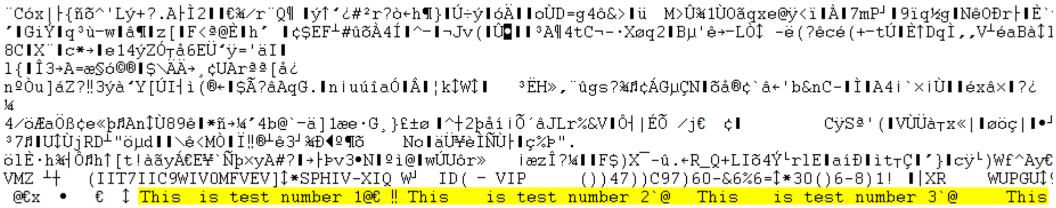
```
ENCDB>w ^x(22)
This is test number 22
ENCDB>
```

5. To open the database file, go to the directory where you created it in the previous section, such as C:\InterSystems\IRIS\mgr\ENCDB, and open the database file, IRIS.DAT. You will see content such as:



6. Try searching for the string “This is test number” in the file. You won’t find it — because the database is encrypted. In fact, the only unencrypted strings you’ll find are the name of the database or the identifier of its encryption key.

7. If you perform the same test on an unencrypted database, the resulting file will include content such as:



```
This is test number 1 This is test number 2 This is test number 3 This
```

Note that the last line of the screen shot contains the values of the variables set in the Terminal.

3.6 Other Features Related to Database Encryption

InterSystems IRIS also has other notable database encryption features that may be important for your implementation:

- KMIP — InterSystems IRIS allows you to store keys on servers that are separate from those that host your instances. To communicate with such servers, InterSystems IRIS supports the key management interoperability protocol (KMIP). This allows InterSystems IRIS to benefit from the standardized approach to key management that KMIP provides.
- Changing keys and adding or removing encryption — You can easily change a database's encryption key. It is also straightforward to encrypt an unencrypted database or make an unencrypted copy of an encrypted database, should either of these actions be necessary.
- Encryption for related data on disk — InterSystems allows you to easily encrypt its temporary cache databases and other on-disk content that it uses to keep its recent transaction records current (that is, its journal files).
- Chip-based encryption — Chips are available that perform encryption as part of their activities, which provides much faster speeds for operation. InterSystems IRIS supports the use of such chips. For more details on chip-based encryption, see the next section.

4 Learn More About Database Encryption

InterSystems has lots of resources to help you learn more about database encryption:

- *Encryption Awareness* — InterSystems Online Learning interactive course providing conceptual introduction to our encryption technology.
- "Managed Key Encryption" in the *Security Administration Guide* — InterSystems documentation on database encryption and related features.
- *FIPS 140–2 Compliance for Database Encryption* — InterSystems documentation on InterSystems IRIS support for the FIPS 140–2 standard.