First Look: Data Resiliency and Mirroring

Version 2018.1
2018-10-22

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

First Look: Data Resiliency and Mirroring
InterSystems IRIS Data Platform Version 2018.1 2018-10-22

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: Data Resiliency and Mirroring

This First Look guide introduces you to the data resiliency features of InterSystems IRIS™. Data resiliency consists of three goals – crash recovery, high availability, and disaster recovery – that are achieved by several InterSystems IRIS features.

One of the key features of InterSystems IRIS Data Platform™ is the capability to provide continuous and undisrupted access to your data by utilizing logical data replication. Data can be replicated synchronously to allow automatic failover with no data loss under a broad range of outage scenarios, either planned (such as software upgrade) and unplanned (such as hardware failure).

Synchronously replicating data requires low latency between two nodes and therefore is not always suitable for disaster recovery (DR) scenarios where one would like to transfer data across country. For those scenarios, InterSystems IRIS Data Platform provides built-in asynchronous data replication.

The mirroring feature of InterSystems IRIS provides high availability. After learning about data resiliency in InterSystems IRIS, you will create a mirror, make data changes on the primary member that are automatically synchronously replicated on the backup member, then shut down the primary so that the mirror fails over to the backup.

These activities use only the default settings, so you can acquaint yourself with the fundamentals of these features. For the full documentation, see the *InterSystems IRIS Data Integrity Guide*.

1 Data Resiliency: What InterSystems IRIS Provides

Data resiliency covers multiple topics, but they all revolve around one principle: once data is recorded in the data platform, it is accessible, no matter what happens.

In order to achieve this goal, there are a few different areas that need to be addressed:

- Guaranteeing data validity even in the event of errors or power failures
- Crash recovery from a local failure of the InterSystems IRIS server, be it physical failure or software related
- Protection against a full site disaster (loss of power, network issues, etc.)

To address these requirements, InterSystems IRIS provides:

- Structural database integrity (the contents of the database blocks on disk) and protection against internal integrity failures (the data represented within the database)
- Logical database integrity through transaction processing, locking, and automatic rollback
- A built-in high availability solution with automatic failover
- Logical data replication that minimizes risks of carry-forward physical corruption and has no dependency on shared resources
- A solution for both planned and unplanned downtime
- Business continuity benefits via a geographically dispersed disaster recovery configuration

InterSystems IRIS data resiliency protects the data on your production systems 24/7.

2 Data Resiliency: How InterSystems IRIS Provides It

The cost of system downtime can range from thousands to millions of dollars, depending on the type and length of outage and the type of system affected. Not only is fast recovery from downtime important but also the ability to recover data and ensure your data is protected from loss and corruption. InterSystems IRIS write image journaling technology provides structural database integrity (the contents of the database blocks on disk) and protects against internal integrity failures (the data represented within the database) due to system crashes. InterSystems IRIS backup and journaling systems provide rapid recovery from physical integrity failures. Data resiliency ensures logical database integrity through transaction processing, locking, and automatic rollback. In addition to the features provided with journaling, InterSystems IRIS also allows you to mirror your databases to provide fast, efficient data replication and disaster recovery.

2.1 Write Image Journaling: Protection Against Physical Data Corruption and Loss

Any sudden, unexpected interruption of disk or computer operation can halt the update of multiple database blocks after the first block has been written but before the last block has been updated. The consequences could be as severe as a database that is totally unusable, with all data irretrievable by normal means. The InterSystems IRIS write image journaling technology protects against this kind of data corruption. It prevents an incomplete update from leading to an inconsistent database structure.

Write image journaling safeguards database updates by using a two-phase approach. The InterSystems IRIS instance first writes updates from memory to a transitional journal, the IRIS.WIJ file, and then to the database. If the system crashes during the first phase, the update is dropped without any changes to the database; if the crash is during the second phase, you can reapply updates from the write image journal (WIH) upon recovery, ensuring that all updates to the database are made.

The write daemon creates the write image journal (WIJ) file when InterSystems IRIS starts and records database updates in the WIJ before writing them to the InterSystems IRIS database. Once it enters all updates to the WIJ, it sets a flag in the file and the second phase begins in which the write daemon writes the same set of blocks recorded in the WIJ to the database on disk. When this second phase completes, the write daemon sets a flag in the WIJ to indicate it is deleted. When InterSystems IRIS starts, it automatically checks the WIJ and runs a recovery procedure if it detects that an abnormal shutdown occurred. When the procedure completes successfully, the internal integrity of the database is restored. InterSystems IRIS also runs WIJ recovery following a shutdown as a safety precaution to ensure that data can be safely backed up. Depending on where the WIJ is in the two-phase write protocol process, recovery does the following:

- If the crash occurred after the last update to the WIJ was completed but before completion of the corresponding update to the databases, the WIJ is restored.
- If the crash occurred after the last WIJ update was durably written to the databases, a block comparison is done between the most recent WIJ updates and the affected databases.

2.2 Journaling: Protection Against Logical Data Corruption and Loss

Journaling is a feature that you can enable on a per-database basis that provides a complete record of all database modifications. In the event of a crash, updates made since the most recent backup can be reapplied from the journal after the backup is restored to bring the database to the point at which the crash occurred.

The InterSystems IRIS recovery process affords maximal protection by using the “roll forward” approach. If a system crash occurs, the recovery mechanism completes the updates that were in progress. It also protects the sequence of updates; if an update is present in the database following recovery, then all preceding updates are also present. This protects the

incremental backup file structures, as well as the database. You can run a valid incremental backup following recovery from a crash.

Journaling is the basis for the third feature of data resiliency: mirroring.

2.3 Mirroring: High Availability and Disaster Recovery Solutions

InterSystems IRIS mirroring falls within the automatic failover category of system availability strategies. Mirroring provides a comprehensive, reliable, and robust enterprise solution for database availability. Traditional availability solutions that rely on shared resources (such as shared disk) are often susceptible to a single point of failure with respect to that shared resource. Mirroring reduces that risk by maintaining independent resources on the primary and backup mirror members. Further, by utilizing logical data replication, mirroring avoids the risks associated with physical replication technologies such as SAN-based replication, including out-of-order updates and carry-forward corruption.

InterSystems IRIS provides two options for high availability mirroring: Mirroring and Virtualization.

2.3.1 Mirroring

InterSystems IRIS mirroring with automatic failover relies on logical data replication between fully independent systems. This avoids the risk of having a single point of failure when using shared storage for multiple systems. It also ensures that a production system can immediately fail over to an alternate InterSystems IRIS instance in all failure scenarios — system, storage, and network.

In an InterSystems IRIS mirror, one InterSystems IRIS instance, called the primary failover member, provides access to the production databases. Another instance on a separate host, called the backup failover member, communicates synchronously with the primary, retrieving its journal records, acknowledging their receipt, and applying them to its own copies of the same databases. In this way, both the primary and the backup always know whether the backup has the most recent journal files from the primary, and can therefore precisely synchronize its databases with those on the primary.

Another great feature of InterSystems IRIS mirroring allows for configuration of a special async member, which can receive updates from multiple mirrors across the enterprise. You can configure an async member for disaster recovery of a single mirror, which allows it to seamlessly take the place of one of the failover members should the need arise. A single mirror can include up to 16 members, which allows you to configure numerous geographically dispersed DR async members. This model provides a robust framework for distributed data replication, thus ensuring business continuity benefits to the organization.

As an added benefit, mirroring with an async member allows a single system to act as a comprehensive enterprise data warehouse. This gives you enterprise-wide data mining and business intelligence.

2.3.2 Virtualization Platforms

Virtualization platforms generally provide HA capabilities, which typically monitor the status of both the guest operating system and the hardware it is running on. The use of mirroring in a virtualized environment, in which the InterSystems IRIS instances constituting a mirror are installed on virtual hosts, creates a hybrid high availability solution combining the benefits of mirroring with those of virtualization. While the mirror provides the immediate response to planned or unplanned outages through automatic failover, virtualization HA software automatically restarts the virtual machine hosting a mirror member following an unplanned machine or OS outage. This allows the failed member to quickly rejoin the mirror to act as backup (or to take over as primary if necessary). On the failure of either, the virtualization platform automatically restarts the failed virtual machine, on alternate hardware as needed. When the InterSystems IRIS instance restarts, it automatically performs the normal startup recovery, maintaining structural and logical integrity as if you restarted InterSystems IRIS on a physical server.

3 Mirroring a Database with InterSystems IRIS: Try It

Now that you have some background knowledge of InterSystems IRIS data resiliency and what it offers, let's see a part of it in action through mirroring.

It's easy to set up a mirrored environment with InterSystems IRIS. You can do it using InterSystems Cloud Manager (ICM), which automatically provisions, deploys, and configures the mirror, or manually using the management portal. This document describes using ICM to deploy the mirror. (For procedures for using the management portal to create a mirror using two installed InterSystems IRIS instances, see Creating a Mirror in the "Mirroring" chapter of the *High Availability Guide*.)

In this experience, you will deploy a mirrored pair of InterSystems IRIS instances in the Amazon Web Services public cloud. To do this, follow the procedure in Try It! Deploy InterSystems IRIS in the Cloud with ICM in *First Look: InterSystems Cloud Manager* and, in the section Customize the Sample Configuration Files, make the following changes:

- In the defaults.json file, change "Mirror": "False", which appears at the end of the file, to "Mirror": "True".
- Customize the definitions.json file:
  - in the DM node definition, change "Count": "1" to "Count": "2"
  - remove the DS node definition
  - add an AR node definition

  The result is the following

```
[
   {
    "Role": "DM",
    "Count": "2",
    "ISCLicense": "/Samples/license/ubuntu/ShardMaster/iris.key"
    },
    {
    "Role": "AR",
    "Count": "1",
    "StartCount": "3",
    "intersystems/arbiter:stable"
    }
]
```

The DockerImage property must be part of the AR node definition because the AR node uses a different InterSystems image from the InterSystems IRIS nodes such as DM and DS. You must ensure that the arbiter image is available locally; for information about doing this, see Identify Docker Repository and Credentials in *First Look: InterSystems Cloud Manager.*

Once you have customized the configuration files, including these changes, continue with the procedure. When you have completed the "**Deploy InterSystems IRIS**" step (the **icm run** command), and the mirror is deployed in the cloud, continue using the ICM command line to try out the mirror by

- Setting a global on the primary failover member and verifying that it is also set on the backup failover member.
- Triggering a planned failover from the primary to the backup.

To do this, use the following procedure:

1. On the ICM command line, review the names and mirror roles of your mirror members by entering the **icm inventory** command, which lists the cloud compute nodes you have provisioned. Nodes provisioned by ICM are named `Label-Role-Tag-NNNN`; where Role is the ICM node type, in this case DM, and the other parts are assigned by you. Mirror members are indicated by a **+** (plus) for the primary and a **-** (minus) for the backup. The examples in the rest of this procedure assume you set the label field to TRYIT and the tag field to TEST, resulting in **icm inventory** output like the following:

```
$ icm inventory
Machine               IP Address       DNS Name
-------               ----------       --------
TRYIT-DM-TEST-0001+   00.53.183.209    ec2-52-53-183-209.us-west-1.compute.amazonaws.com
TRYIT-DM-TEST-0002-   00.53.183.185    ec2-52-53-183-185.us-west-1.compute.amazonaws.com
TRYIT-AR-TEST-0003    00.53.183.112    ec2-52-53-183-112.us-west-1.compute.amazonaws.com
```

Note: There is no relationship between node numbering and mirror role; always be sure to check for the **+** to identify the primary.

2. Open an InterSystems IRIS Terminal on the primary mirror member by entering the command:

```
$ icm session –machine TRYIT-DM-TEST-0001
```

3. When the Terminal window opens, you are in the **DB** namespace, which is mapped to the mirrored database DB; ICM created both by default. Enter the following command to create a *global* in the DB database (globals store data in InterSystems IRIS databases):

```
DB> Set ^myfirstglobal="congratulations"
```

4. Enter the following command to display the global's value:

```
DB> Write ^myfirstglobal
congratulations
```

5. Open a Terminal window for the backup failover member with the following command:

```
$ icm terminal –machine TRYIT-DM-TEST-0002
```

6. Enter the same **Write** command; the global exists and the value is what you set on the primary! The data was automatically synchronously mirrored to the DB database on the backup.

```
DB> Write ^myfirstglobal
congratulations
```

7. Mirrored databases on the backup are read-only; if you try to change the value of the global, you get an error.

```
DB> Set ^myfirstglobal="what next?"
<PROTECT>
```

8. Returning to the ICM command line, shut down and restart the primary InterSystems IRIS instance (named IRIS by default by ICM) inside its container with the following command:

```
$ icm exec -command "iris stop IRIS quietly restart" -machine TRYIT-DM-TEST-0001
```

Note: The **-quietly** argument prevents the **iris stop** command from prompting the issuer, which would cause ICM to wait until it timed out. Alternatively, you could execute the command interactively, as follows, allowing you to respond to prompts:

```
$ icm exec -command "iris stop IRIS restart" -interactive -machine TRYIT-DM-TEST-0001
```

9. Wait a few seconds, then return to the Terminal window on TRYIT-DM-TEST-0002. When the primary becomes unavailable due to a planned or unplanned outage, the mirror automatically fails over to the backup, which becomes the new primary. So TRYIT-DM-TEST-0002 has become the primary and you are now able to change the value of the global.

```
DB> Set ^myfirstglobal="what next?"
DB> Write ^myfirstglobal
what next?
```

10. Display the new mirror roles with the **icm inventory** command:

```
$ icm inventory
Machine               IP Address       DNS Name
-------               ----------       --------
TRYIT-DM-TEST-0001-   00.53.183.209    ec2-52-53-183-209.us-west-1.compute.amazonaws.com
TRYIT-DM-TEST-0002+   00.53.183.185    ec2-52-53-183-185.us-west-1.compute.amazonaws.com
```

11. Open a new Terminal window for the former primary, TRYIT-DM-TEST-0001, which on restarting found that TRYIT-DM-TEST-0002 was now primary and so became the new backup. As a result, the global's value has been updated to what you set on the new primary, and you cannot change it.

```
$ icm terminal –machine TRYIT-DM-TEST-0001
DB> Write ^myfirstglobal
what next?
DB> Set ^myfirstglobal="another First Look!"
<PROTECT>
```

The backup mirror member is always ready to take over from the primary to keep your databases available and protect your data from corruption and loss.

When you have finished your mirror experience, be sure to return to Unprovision the Infrastructure in *First Look: InterSystems Cloud Manager* and follow the procedure for unprovisioning your cloud nodes. Because AWS and other public cloud platform instances continually generate charges, it is important that you unprovision your infrastructure as soon as you are done with it.

4 Learn More About InterSystems Data Resiliency

To learn more about InterSystems IRIS data resiliency and ICM, see

Further reading:

- The High Availability and Disaster Recovery Resource Guide
- High Availability Guide
- Data Integrity Guide
- InterSystems Cloud Manager Guide
- What is InterSystems Cloud Manager? (video)
- The ICM Experience: Pat and Tracy (video)
- InterSystems in the Cloud Experience (online experience)