Article
Bob Binstock · Sep 6, 2016 19m read

Caché Mirroring 101: a Brief Guide and FAQ

Mirroring 101

Caché mirroring is a reliable, inexpensive, and easy to implement high availability and disaster recovery solution for Caché and Ensemble-based applications. Mirroring provides automatic failover under a broad range of planned and unplanned outage scenarios, with application recovery time typically limited to seconds. Logical data replication eliminates storage as a single point of failure and a source of data corruption. Upgrades can be executed with little or no downtime.

Deploying a Caché mirror does, however, require significant planning, and involves a number of different procedures. And like any other critical infrastructure component, the operating mirror needs ongoing monitoring and maintenance.

You can use this article in two ways: as a **frequently asked questions** list, or as a brief sequential guide to understanding and evaluating mirroring, planning a mirror, configuring a mirror, and operating a mirror. Each answer contains links to detailed discussions of each topic and to step-by-step procedures for each task.

When you are ready to begin planning a mirror deployment, your starting place should always be the Mirroring Architecture and Planning section of the "Mirroring" chapter of the *Caché High Availability Guide*.

Frequently Asked Questions

Understanding and Evaluating Mirroring

What are the benefits of mirroring?

Can a mirror be deployed in a virtualized environment?

Can a mirror be deployed in the cloud?

What is the basic design of a mirror?

How are the database copies synchronized with
databases?

How is automatic failover triggered? Are there sit
cover?

Does a mirror provide disaster recovery?

Planning the Mirror

How do I plan the mirror's architecture? What members will be included, and where will they be?

What network and latency considerations apply? What kind of network configuration will the mirror need?

What are the options for redirecting application c
primary on failover?

What are the compatibility requirements for the C
mirror?

How will I migrate existing databases to the mirro

What should I consider if the mirror will be deploy
environment?

Understanding and Evaluating Mirroring

What are the benefits of mirroring?

There are three primary approaches to high availability for Caché and Ensemble-based applications: failover clusters, virtualization HA, and Caché mirroring. The most significant drawback of the first two is that they rely on shared storage, so storage failure is disastrous; optional storage-level redundancy can ameliorate this, but can also carry forward some types of data corruption. In addition, software upgrades require significant downtime, and application recovery time can be minutes for many failures.

By using two physically independent systems with separate storage and logical data replication, mirroring avoids the shared storage problem, upgrades require no or minimal downtime, and application recovery time is typically seconds. This approach also allows mirroring to provide a reliable and robust disaster recovery capability, with the disaster recovery site located at any appropriate distance from the production data center.

The main limitation of mirroring is that it replicates only the databases themselves; external files needed by the application require an additional solution, and security and configuration management is currently decentralized.

The following sources provide detailed analysis and comparison of these HA approaches as well as more information about mirroring's benefits:

- System Failover Strategies (*Caché High Availability Guide*)
- High Availability Strategies (white paper)
- High Availability for Business Continuity (video)
- Caché Mirroring: An Adventure in High Availability (video)

- Mirroring: Architecting for Throughput (online learning)
- InterSystems Caché: Database Mirroring: An Executive Overview (white paper)
- Mirroring Awareness (online learning)
- HealthShare: Achieving High Availability with Mirroring (online learning)

Can a mirror be deployed in a virtualized environment?

Mirroring is often deployed in virtualized environments. 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. Thus allows the failed member to quickly rejoin the mirror to act as backup (or to take over as primary if necessary).

See the InterSystems white paper High Availability Strategies for information about using this approach.

Can a mirror be deployed in the cloud?

Mirroring can be effectively deployed in the cloud. Use of a virtual IP address (mirror VIP) to redirect application connections after failover Is usually not possible due to cloud network restrictions, but this can be effectively overcome using network traffic managers such as load balancers.

What is the basic design of a mirror?

A Caché mirror typically includes two Caché instances on physically independent hosts, called *failover members*; the mirror automatically assigns the role of *primary* to one, while the other becomes the *backup*. Applications update the primary's databases, while the mirror keeps the backup's databases synchronized with the primary's.

When the primary fails or becomes unavailable, the backup automatically takes over as primary, and application connections are redirected to it. When the primary instance is restored to operation, it automatically becomes the backup.

Operator-initiated failover can be used to maintain availability during planned outages for maintenance or upgrades.

A mirror optionally contains additional members called *asyncs*, for disaster recovery and for business intelligence and data warehousing purposes.

A mirror can also operate with just one failover member and some number of asyncs, for example when disaster recovery is the primary goal.

How are the database copies synchronized with the live production databases?

The backup and async members of a mirror are kept synchronized with the primary using journal files, which contain a time-sequenced record of the changes made to the databases in a Caché instance since they were last backed up. Within a mirror, journal files from the primary are sent to and *dejournaled* on other members—that is, the changes recorded in them are applied to the local copies of the databases, keeping them up to date with the primary.

Transfer of journal records from primary to backup is synchronous, with the primary waiting for acknowledgement from the backup at key points. This keeps the failover members closely synchronized and the backup active, or ready to take over as primary. Asyncs receive journal data from the primary asynchronously, and as a result may sometimes be a few journal records behind.

How is automatic failover triggered? Are there situations it doesn't cover?

The backup can take over automatically only if it confirms that the primary can no longer operate as primary without manual intervention. When direct communication between the failover members is interrupted, the backup gets

help confirming this from a third system, the arbiter, which maintains independent contact with both failover members.

In addition, automatic failover cannot occur if the backup cannot confirm that it has or can obtain the latest journal data from the primary. Agent processes running independently of the Caché instance on each failover host, called ISCAgents, are involved in this and other aspects of automatic failover logic and mechanics.

Assuming the arbiter is functioning properly, almost all unplanned primary outages are covered; only a network failure that isolates the failover members both from each other and from the arbiter prevents an active backup from taking over for a failed or unavailable primary.

Does a mirror provide disaster recovery?

One type of async mirror member is the disaster recovery (DR) async. A DR async has a copy of all mirrored databases on the primary and can be promoted to failover member at any time. When an outage leaves the mirror without a functioning failover member, you can manually fail over to a promoted DR async; the extent of data loss will depend on how far behind the primary the DR async was when the outage occurred, and whether the former primary's host system is functioning, allowing it to obtain additional journal data. A promoted DR async can also be useful in a number of other planned and unplanned outage situations.

Planning the Mirror

How do I plan the mirror's architecture? What members will be included, and where will they be?

The size, membership, and physical distribution of your mirror will depend on your reasons for deploying it and a number of infrastructural and operational factors, allowing for a great many possible configurations

A mirror with two failover members provides high availability through automatic failover. Of the optional async members, one or more DR asyncs can provide data security and disaster recover capability, while reporting asyncs are used for purposes such as data mining and business intelligence. A single reporting async can belong to up to 10 separate mirrors, allowing it to function as an enterprise-wide data warehouse bringing together sets of related databases from separate locations.

A mirror can also consist of a single failover member and a number of asyncs for disaster recovery and reporting purposes, as long as automatic failover is not required.

A mirror can include up to 16 members. While failover members require a low latency connection and are therefore typically colocated, async members can be local or in separate data centers, including geographically remote locations that provide the greatest security for the data on DR asyncs.

Multiple mirror members can be installed on a single host, but additional planning is required.

What network and latency considerations apply? What kind of network configuration will the mirror need?

Primary network configuration considerations include reliability, bandwidth, and network latency, an important consideration in application performance. Compression of journal data transmitted by the primary to other members is usually but not always be selected.

Each mirror member has several different network addresses, used for different purposes, which should be well understood before planning the network configuration needed to support your mirror. Sample mirror and network configurations for mirrors contained within a single data center, computer room, or campus and for mirrors involving dual data centers and geographically separated disaster recovery will help you define the needed network configuration.

What are the options for redirecting application connections to the new primary on failover?

Several options for automatic redirection are built into mirroring and Caché , including the use of a virtual IP address (VIP) for the mirror, identification of ECP data servers as mirror connections, and mirror-aware CSP gateways.

A mirror VIP is typically a very effective solution, but does require some advance planning, particularly in regard to network configuration.

A range of external technology options is also available, including the use of network traffic managers such as load balancers, automatic or manual DNS update, application-level programming, and user-level procedures.

What are the compatibility requirements for the Caché instances in the mirror?

Before identifying the systems to be added to a mirror, be sure to review the requirements for Caché instance and platform endianness compatibility. Since the failover members can trade roles as primary and backup at any time, they should be as similar as possible; CPU and memory configuration should be the same or close, and the storage subsystems should be comparable.

How will I migrate existing databases to the mirror?

Any Caché database can easily be added to a mirror; all that is required it the ability to either back up and restore the database, or copy its **CACHE.DAT** file. Procedures are noted in the following section.

What should I consider if the mirror will be deployed in a virtualized environment?

When using mirroring in a virtualized environment, it is important to plan the correct relationship between virtual mirror member hosts and physical hosts and storage; there are also important operational considerations from both the mirror and the virtualization platform sides.

Configuring the Mirror

What configuration guidelines do I need to consider?

If you plan to configure a mirror virtual IP address (VIP), InterSystems recommends configuring the failover members to use the same superserver port and web server port.

Neither Caché instance configurations (such as users, roles, namespaces, and mappings) or unmirrored data (such as files related to SQL gateway and webserver configuration) on the primary failover member are replicated by the mirror on other mirror members. Therefore, any settings or files required to enable the backup or any DR async members (which may be promoted) to take over from the primary in the event of failover must be manually replicated on those members and updated as needed.

Do not disable Internet Control Message Protocol (ICMP) on any system that is configured as a mirror member; mirroring relies on ICMP to detect whether or not members are reachable.

As journaling is the basis for mirror synchronization, it is essential to monitor and optimize journaling performance on the failover members and follow journaling best practices generally. In particular, InterSystems recommends that you increase the shared memory heap size on all mirror members.

How do I secure the mirror?

The primary means of securing mirroring communication is SSL/TLS, which encrypts all traffic within the mirror using X.509 certificates. SSL/TLS security is *strongly recommended*. To enable SSL/TLS on a mirror, you must first create a mirror SSL/TLS configuration on each mirror member; you may find it most convenient to do this before

creating the mirror. When SSL/TLS is enabled, each member added to the mirror must be authorized on the primary; the same applies when a member's X.509 certificate is updated.

For another layer of protection for a mirror using SSL/TLS , you can activate journal encryption. This means that journal records are encrypted as they are created on the primary, using one of its active encryption keys, and decrypted before being dejournaled on other members. The backup and all asyncs must have the same key activated, and the backup and DR asyncs must also use it to encrypt data.

The way in which you configure the network used by a mirror also has important implications for the mirror's security.

How do I configure a mirror virtual IP address (mirror VIP)?

The mirror VIP is configured by specifying details when creating and adding members to the mirror, or when modifying the mirror, but some preparation is required, including identification of the needed information and possibly configuration of the mirror members' hosts and Caché instances.

Where and how do I install the arbiter?

The arbiter should be located to minimize the risk of unplanned simultaneous outage of the arbiter and a failover member (if both failovers are out, the arbiter becomes irrelevant), so its location depends primarily on the locations of the failover members. A single system can be configured as arbiter for multiple mirrors, provided its location is appropriate for each. A system hosting one or more failover or DR async members of a mirror should not be configured as arbiter for that mirror.

Any system with a running ISCAgent of version 2015.1 or later, including one that hosts one or more instances of Caché version 2015.1 or later, is ready be configured as arbiter. You can prepare any other supported system (except an OpenVMS system), including one hosting a pre-2015.1 Caché instance, to be configured as arbiter by installing the ISCAgent.

How do I install and start the ISCAgent?

The ISCAgent is automatically installed with Caché, and is therefore installed on any mirror member. However, the agent must be configured to start on system startup on each mirror member.

How do I create and configure the mirror?

Configuring a mirror is a multistep process:

1. Create a mirror and configure the first failover member
2. Configure the second failover member (if desired)
3. Authorize the second failover member, if using SSL/TLS (recommended)
4. Configure async mirror members (if desired, either DR or reporting)
5. Authorize new async members, if using SSL/TLS (recommended)

After any of these steps, you can review the mirror's status in the Mirror Monitor to confirm that results are as intended.

How do I create a mirrored database? How do I add an existing database to the mirror?

Before adding databases to the mirror, you may want to review certain mirrored database considerations relating to what can and cannot be mirrored, simultaneous use of mirroring and shadowing, propagation of mirrored database properties, and the maximum number of databases per instance under mirroring.

The procedures for creating mirrored databases and adding existing databases are different because changes to

mirrored databases are recorded in mirror journal files, which are different from non-mirror journal files. If a database is created as a mirrored database, it uses mirror journal files from the start, which makes it easy to add a new database to the mirror by creating a mirrored database with the same mirror name on each of the mirror members, beginning with the primary.

When you add an existing, non-mirrored database as a mirrored database on the primary, it switches from using non-mirror journal files to mirror journal files. You therefore cannot simply create the database on the other members, because the mirror cannot convey the non-mirror journal files to the other members. Instead, after the database has been added to the mirror on the primary, you must either back it up there and restore it on the other members, or copy its **CACHE.DAT** file to the other members.

How do I ensure that ECP redirects application server connections after failover?

Whether or not you have configured a mirror VIP, you can ensure that ECP connections are redirected to the new primary by configuring the mirrored ECP data server as a **Mirror Connection** on each ECP application server that connects to it. (An application server does not use the VIP; since it regularly collects information from the specified host, it automatically detects a failover and switches to the new primary.)

How do I redirect application connections when a mirror VIP is not possible, for example in the cloud?

A mirror VIP can be used only when mirror members reside on the same network subnet, which is not typically the case when they are located in separate data centers. For similar reasons, a VIP is usually not an option for deployments in the cloud.

A range of external technology alternatives is available, including the use of network traffic managers such as load balancers, (physical or virtual), which can be used to achieve the same level of transparency as a VIP, presenting a single address to client applications or devices. Other possible mechanisms include automatic or manual DNS update, application-level programming, and user-level procedures.

How do I convert a Caché shadow into a mirror?

Mirroring provides a shadow-to-mirror utility that allows you to convert a shadow source and destination and the shadowed databases mapped between them to a mirror with primary, backup or async, and mirrored databases.

What other configuration details should I look into?

While the default is typically all that is needed, you may want to customize the ISCAgent port number.

On the primary failover member, you may want to move code from existing **^ZSTU** or **^ZSTART** routines to the user-defined ^ZMIRROR routine, which lets you implement custom, configuration-specific logic and mechanisms for specific mirroring events, so that it is not executed until the mirror is initialized.

When using mirroring with Ensemble, you should be aware of special requirements for Ensemble namespaces with mirrored data and the functioning of Ensemble Autostart in a mirrored environment.

Managing the Mirror

How do I monitor the mirror's operation?

The Mirror Monitor, which you can load in the Caché management portal of any mirror member, provides detailed information about

- The operating status of the mirror and each of its members, including the x.509 DNs of members when

SSL/TLS is in use.
- On the failover members, the network addresses and arbiter connection status of the both failover members, along with the arbiter's address; on an async, the mirrors a reporting async belongs to.
- On the backup and async members, the status of journal data transfer from the primary and dejournaling of journal data, and the rate at which journal data is arriving from the primary.
- The status of mirrored databases on the member on which you load the Mirror Monitor.

The Mirror Monitor also lets you perform a number of operations, including viewing and searching through the member's journal files, promoting a DR async to failover member or demoting the backup to DR async, and activating, catching up and removing mirrored databases.

You can use the Caché system status routine (**^%SS**) in the **%SYS** namespace on a mirror member to monitor its mirroring communication processes.

How do I modify the mirror? What can I modify?

Edit the mirror on the primary to change the mirror's configuration (including SSL/TLS, mirror VIP, and so on) and to update members' network addresses when your network configuration changes. You must also edit the mirror on the primary to authorize X.509 certificate updates on other members.

Edit the mirror on an async to change the async type, add a reporting async to another mirror, and make other async-specific changes.

You can remove mirrored databases from the mirror on any member (and only on that member) using the Mirror Monitor, although the implications vary depending on the type of member involved.

Can I add a member to the mirror? Remove one? How do I remove a mirror altogether?

You can always add async members to the mirror, up to the limit of 16 members total. If you have one failover member and fewer than 15 asyncs, you can always add a backup. You can also replace the backup by promoting a DR async to failover member, which automatically demotes the current backup to DR async.

You can edit the mirror on any member to remove that member from the mirror. To remove the mirror entirely, you must remove members in a specific order and take additional steps.

What if I need to temporarily remove a member from the mirror?

You can use the Mirror Monitor to indefinitely stop mirroring on the backup or an async member by disconnecting the member from the mirror, for example for maintenance, or (in the case of the async) to reduce network load.

On an async, you can also pause dejournaling for all of the databases in a mirror without pausing transfer of journal data from the primary to the async.

Must I upgrade the mirror all at once? Must I take the mirror out of production to do it?

All failover and DR async members of a mirror must be of the same Caché version, and can differ only for the duration of a mirror upgrade. Once an upgraded member becomes primary, you cannot make use of the other failover member or any DR async members until they are upgraded as well. Generally, the best practice is to upgrade reporting asyncs to the same version at the same time.

The upgrade procedure you choose depends on whether you are making a maintenance release upgrade, a major upgrade without any changes to mirrored databases, or a major upgrade with mirrored database changes. The procedures provided are designed to minimize application downtime; in the first two cases, you will generally be able to avoid it entirely, and in the last case, it will typically be limited to the time it takes to execute a planned failover and make the required mirrored database changes.

There is also a simpler procedure you might want to use when you are making a major upgrade during planned downtime and don't need to minimize application downtime.

What other mirror or mirror-related management procedures and details should I know about?

You can enable SSL/TLS security on a mirror that is not using it, as long as every member has a valid mirror SSL/TLS configuration.

You can activate journal encryption for a mirror that is not using it, as long as the mirror uses SSL/TLS security and the active encryption key used to encrypt the journal data on the primary is also active on the backup and all asyncs.

Depending on your hardware and network configuration, you may want to adjust the mirror's Quality of Service Timeout (QoS timeout) setting, which plays an important role in failover mechanics. Typically, this setting can be reduced, if a faster response to outages is desired, on mirrors deployed on physical (non-virtualized) hosts with a dedicated local network.

if the vast majority of mirrored database updates consist of highly compressed data (such as compressed images) or encrypted data, journal data compression is not expected to be effective, and may therefore waste CPU time. In such a case, you may choose to configure or modify the mirror to set journal data to **Uncompressed**. (Use of Caché database encryption or journal encryption is not a factor in selecting compression.)

If network latency between the primary and other mirror members becomes an issue, you may be able to reduce it by fine-tuning operating system TCP parameters to allow the primary and backup/async members to establish send and receive buffers, respectively, of appropriate size.

The ^MIRROR routine provides a command-line alternative to the management portal for all mirroring tasks. The SYS.Mirror API provides methods for programmatically calling the mirror operations available through the management portal and the **^MIRROR** routine.

Mirror Outage Procedures

For an overview of recommended procedures for dealing with a variety of planned and unplanned mirror outage scenarios, see Mirror Outage Procedures.

#Best Practices #Cloud #Databases #Ensemble #High Availability #Failover #Mirroring #InterSystems IRIS #Tips & Tricks #Caché

Source URL: https://community.intersystems.com/post/cach%C3%A9-mirroring-101-brief-guide-and-faq