Connectivity

Alliance Access 7.0

Database Recovery

Information Paper

Table of Contents

Preface

Purpose of this document

This document gives an overview on how to recover Alliance Access from typical failures and explains how to increase and simplify your resiliency with the new Database Recovery option.

Intended audience

This document is aimed at Alliance Access customers who want to know how the new Database Recovery option can simplify and improve their resiliency setup.

We encourage you to read this document in case any of the following applies to you:

- You have put resiliency measures in place, which include manual, complex and error prone operations and you want to know how to improve them.
- You have invested in hardware clustering solutions or SAN disks, but this does not protect you against a database loss.
- You want to optimise your disaster site setup and are looking for ways to replicate data there.
- You want to review your resiliency measures and check whether you are fully covered.

Related documentation

- Alliance Access 6.3 release overview
- Alliance Access/Entry 7.0 Functional Overview
- *Alliance Access - System Management Guide*
- *Alliance Access - Installation and Administration Guide*

1 Overview

Access holds critical living business data

Alliance Access core function is to exchange with SWIFT the financial messages managed by the back office applications. Access stores these financial messages into its database to manage their life cycle, which represents the work flow necessary to complete the message exchange between SWIFT network and the back office systems.

As part of this workflow, Access has at any time a set of 'live' messages, which represent business transactions, being exchanged with correspondents but not yet finalised. It can either be transactions waiting to be sent to correspondents or sent but waiting to be confirmed by correspondents. It can also be transactions received from correspondents and waiting to be processed in the back office systems. Once fully treated by the back office applications, the messages become 'completed' in Access, meaning that they do not need to be maintained anymore and can be archived.

Business Impact of losing live business data

When a major failure leads to the loss of the Access database (either due to the unavailability of the whole Access system or due to a corruption of the database itself), it also leads to the loss of these live messages present in the Access database.

These live messages represent business critical information that if not recovered, will definitely cause the loss of financial operation in the back office applications.

Manual Recovery procedures

The Access Database Backup & Restore tool can be used to restore the Access configuration data, but it does not restore messages, which means that that it cannot be used to recover these lost live messages.

Prior to Access 6.3, customers had to implement their own recovery procedures to circumvent this risk:

- For the messages sent to SWIFT, customers must adapt their back office applications to be able to re-send, with a PDE trailer, all messages waiting for a SWIFT acknowledgment
- For the messages received from SWIFT, customers must rely on SWIFT retrieval service to request missing messages, using the MT020 system message. The retrieval operation requires customer to precisely identify across all their back office applications, the OSN of the last messages received, and to generate an OSN gap retrieval message. This retrieval must be executed for each LT (BIC9 logical terminal) in use.

Database Recovery - Simpler, faster, cheaper and more reliable recovery

As of Access 6.3, the recovery of the whole database content, including live messages, is now possible. It is provided by the new Database Recovery option, an easy-to-use tool that requires a single command to restore the whole database content. Database Recovery is the function positioned to support recovery of live messages.

For customers who have not yet implemented recovery procedures, Database Recovery provides an easy and fast way to protect against loss of their database content.

For customers who have already implemented their recovery procedures (as explained above) Database Recovery provides a simpler and more reliable alternative:

- More reliable

  For messages sent to SWIFT, customers do not need to rely anymore on message re-emission with a PDE trailer. This re-emission always has the risk that the correspondent ignores the PDE trailer and treats the transaction twice.
- Simpler

  For messages received from SWIFT, customers do not need to implement the complex recovery procedures based on OSN gap retrieval service.
- Cheaper

  The customer does not need to rely on the message retrieval service, which comes with an extra cost.
- Faster

  Database Recovery relies on native Oracle database features, therefore providing the most efficient way to restore the database content.

Disaster Site Recovery Support

When the distance between the primary and disaster site is of such an order that only asynchronous data replication works, not all data will be replicated on the disaster site when the prime site goes down. So, asynchronous data replication always results in minimal data loss.

Database Recovery will provide a new feature as of Access 7.0, which restores the database content from incomplete data available in the disaster site. In that mode, Database Recovery will restore the database up to the last valid transaction found in the available data. The recovered database will be nearly up to date, missing only the last updates done on the main Access system before losing the primary site.

2 Resiliency Concepts

2.1 Database Loss Business Impact

This section explains in more details the business impact of losing an Access database.

2.1.1 Database Business Purpose

Alliance Access is SWIFT's prime interface to manage the communication between back office applications and SWIFTNet.

To achieve this role, a key function of Alliance Access is to manage the life cycle of messages, which represents the period necessary for Access to complete the exchange of a message between SWIFT and the back office applications.

The life cycle of a message in Alliance Access is as follows:

- Outgoing messages (the ToSWIFT flow),
  - The life cycle starts when the message is received from a back office or when it is created manually by a user. It is created in Access database as 'live'.
  - The message is eventually sent to SWIFT (possibly after manual verification and authorisation steps).
  - The life cycle completes when the network acknowledgement received from SWIFT has been reconciled with the message and sent back to the back office.

  During this life cycle period, the message is considered as 'live'. When the acknowledgment has been successfully processed, the message is marked as 'completed'. From that moment, the completed message is kept for a user defined period in Access database, primarily for audit purposes (archive consultation).
- Incoming messages (the FromSWIFT flow)
  - The life cycle starts when the message is received from SWIFT and created as 'life' in Access database
  - The life cycle completes when the message has been successfully passed to a back office application or to a printer (for manual processing).

  The message is then marked as 'completed'. From the moment, as for the ToSWIFT flow, the completed messages are kept for a user defined period in Access database, for audit purposes.

As explained further, the life cycle concept is an important notion to take into account for a resilient configuration, as it introduces a notion of 'in-progress' data that must be recovered in a failure scenario.

2.1.2 Database Loss Impact

Before going further into the analysis of the different recovery options of Access, this section explains the business impact of losing the Access database.

Note This document assumes that the connectivity resiliency recommendations, as described in the Connectivity Resilience Guide are implemented, meaning that a failure of the Gateway or SWIFTNet component will be transparently recovered (Access has an automatic failover option to a secondary Gateway connection, providing transparent recovery against a Gateway failure).

This document focuses on failures affecting the Access system and resulting in the temporary and permanent unavailability of the Access system.

Connection loss

The first consequence of an Access failure is the temporary unavailability of network connectivity, affecting the back office applications and user. The main impact of this connectivity loss is the risk of missing 'cut offs'. Customers are therefore looking at minimising the connectivity downtime.

Operational data loss

In some situations, the Access failure will not be limited to a temporary unavailability of the connection, but it will also lead to the unrecoverable loss of the Access database.

The database loss can be caused by:

- System unavailability

  The Access system, owning the database, becomes unavailable. Causes for such unavailability are multiple (major hardware failure, main power shutdown, primary site loss).
- Database unavailability

  The Access system is still available, but its database becomes unusable. Causes for such database loss are often linked to a human error when manipulating the database or the underlying operating system. The probability of losing the database due to an internal corruption or a malfunction of the database engine is still theoretically possible, but the probability of occurring has been strongly reduced with the introduction of an Oracle database since Access 6.2.

The database loss will result in losing all events and messages stored in the Access system, with the following impact:

- Live messages

  The loss of live messages represents the most critical business loss, as they represent business transactions that have not been completed:
  - Live FromSWIFT messages represent business transactions received from SWIFT but not yet known to the back office applications. SWIFT will not re-send these messages, as they have been confirmed (UAcked) by Access.
  - Live ToSWIFT messages represent business transactions sent to SWIFT but with the uncertainty whether SWIFT has really received them. The messages could have been sent, but the SWIFT acknowledgment was not yet received by the back office. Or the messages were accepted by Access but were not yet sent to SWIFT.
- Completed messages and events

  The loss of completed messages and events is less critical as there is no business impact (the transactions are completed), but there is only an audit issue (as the transaction history is lost and cannot be consulted anymore).

In summary, the business impact of losing operational data is critical, as without any recovery action, it will definitely lead to the loss of business transactions exchanged with correspondents.

2.2 Database Recovery Tools

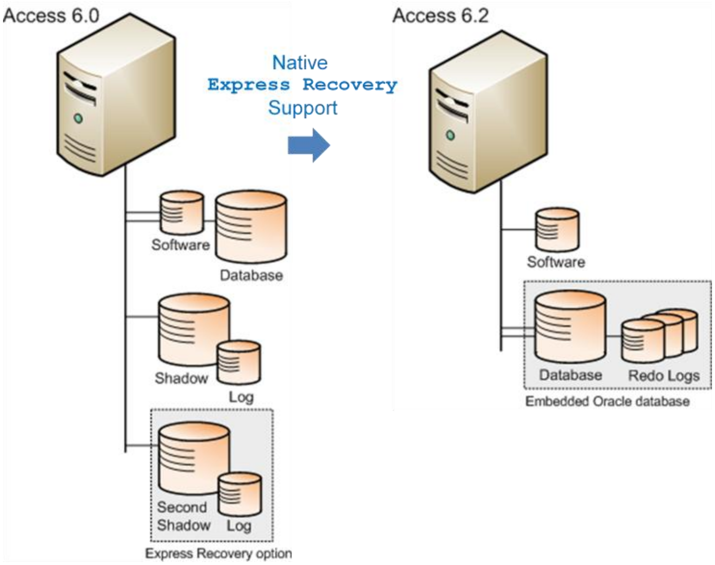
This section details the different resiliency mechanisms available in Access.

2.2.1 Native Express Recovery

A core function of any database engine is to ensure, in case of an abnormal shutdown of the system (meaning that the system may not have had time to properly close all its files), that the database integrity and consistency are preserved. This core function also ensures that no database updates are lost.

With the C-ISAM based architecture, Access 6.0 implemented this core function using a built-in recovery (shadow database and log trail). However, for higher volume customers, the potential large size of the log trail could lead to a long recovery time, resulting in extended unavailability of Alliance Access at start-up time. In response, the ‘Express Recovery’ option was proposed to guarantee a limited time delay when restarting the database.

Since release 6.2, Access internal database is based on Oracle. The Oracle database natively provides the core functionality to ensure integrity, consistency and guaranteed updates. Oracle also guarantees a fast recovery of the database in case of abnormal shutdown.



In summary, since Access 6.2, the Express Recovery function is natively supported by the system and no longer requires to be licensed additionally.

It is still important to note that this function protects against an abnormal closure of the database but does not protect against a database loss.

2.2.2 Access Database Backup & Restore tools

Access provides the Database Backup & Restore tools, allowing to take regular backups of the live system and to possibly restore these backups in case of unavailability of the Access database:

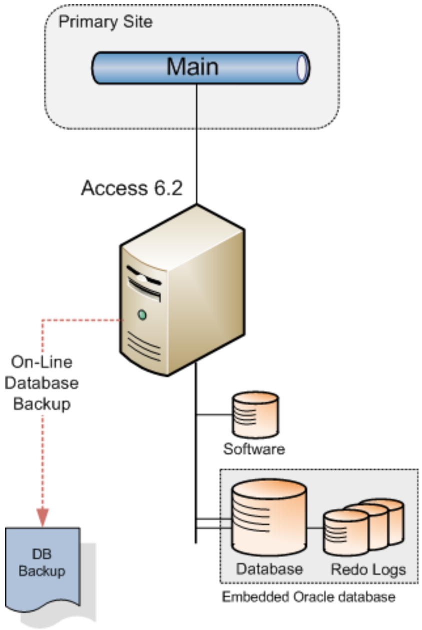
- The Backup tool backs up all Access configuration data into an external backup file.
- The Restore tool uses this external backup file to restore Access configuration.

  The Restore tool can also restore subsets of Access configuration (referred to as 'Selective Restore')

The restore tool will allow restoring the configuration information, but the restore function **will not allow** restoring operational data, i.e. messages and events.

This operation is sometimes referred to a 'cold recovery' concept, as it ensures that Access can be recovered with the latest configuration information, but with an empty operational database.

Note Since Access 6.2, with the embedded Oracle engine, the Backup of Access configuration can be done on-line while the system is running (In Access 6.0, the system still needs to be stopped before taking a backup to ensure that the most updated configuration is backed up).



In summary, the Access Database Restore tool allows Access configuration to be recovered but it does not protect against the loss of operational data (messages and events).

2.2.3 Database Recovery tool

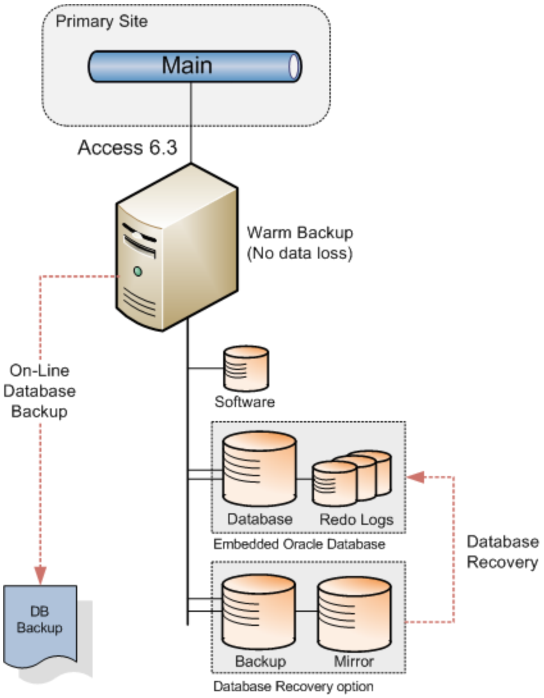
Database Recovery is a new licensable feature since Access 6.3 that can be used to restore the whole database content.

Database Recovery is fundamentally different from the Database Backup/Restore tool, as it allows restoring messages and events.

Database Recovery is operated much differently from the Database Backup & Restore tool.

As explained in more details in section 5.1, Basic principles, Database Recovery first requires to be activated to start maintaining on additional file systems (referred to as mirror and backup disks) the updates done on the live database (referred to 'Recovery Data').

In case of a major incident resulting in the loss of the Access database, this recovery data is used to restore the database to its last committed state. For that purpose, Database Recovery provides a 'recover' command, which must be launched by the Access administrator, to initiate the recovery of the database.



In summary, Database Recovery is the solution to protect against a failure resulting in the loss of the Access database. With a single command, it can restore the database up to its last status. It is the only tool that guarantees that business operations can be resumed without any loss of operational data.

3 Manual Recovery Method

Prior to the availability of the new Database Recovery feature, customers had to implement the following steps to recover the lost business transactions (i.e. the live FromSWIFT and ToSWIFT messages that were present in the database).

- FromSWIFT manual recovery

  The customer must perform an OSN retrieval (based on the MT020 and MT021 system messages) to request SWIFT to resend the messages that were lost. This operation is usually complex, as it requires identifying the OSN gap per each Logical Terminal (i.e. the gap between the OSN of the last message received versus the last OSN known by SWIFT), and to generate the corresponding MT020 OSN retrieval message.
- ToSWIFT manual recovery

  The customer back office systems must identify all the messages sent to Access but still waiting for the SWIFT acknowledgment, and re-send all these messages to Access, with a PDE trailer.

  This PDE trailer is mandatory as the back office cannot distinguish between the messages that were actually sent (but where the ACK has not been received by the back office) from the messages that were in Access but not yet sent.

In summary, it is possible for the customer to implement procedures recovering lost operational data. This recovery process is however complex to implement and can potentially impact all back office applications. The time required to execute and complete this manual recovery procedure may also be significant, possibly increasing the Access downtime.

The Database Recovery option, available since Access 6.3, and further enhanced with Access 7.0, provides all the features to support a full recovery of the database content, removing the need for the customer to implement these complex manual recovery procedures.

4 Resiliency Configurations

This section explains the resiliency configurations that are usually implemented by customers. The configuration chosen by a customer depends on the resiliency needs and the infrastructure the customer is ready to invest in to achieve a certain level of resiliency.

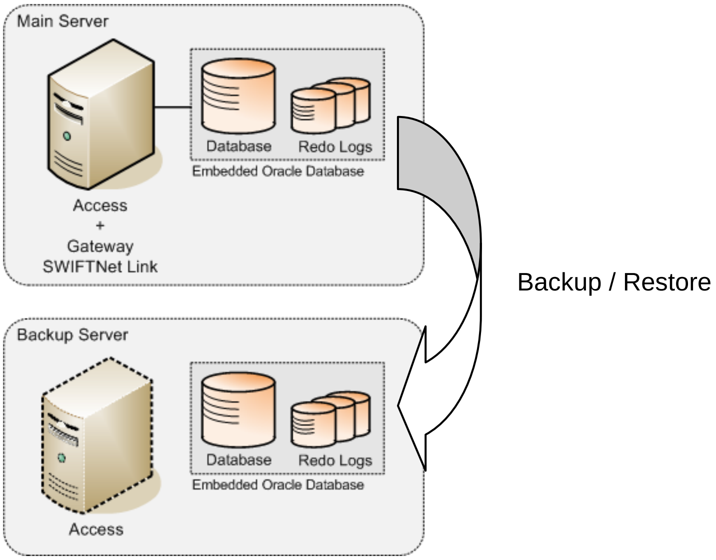
This section also explains how the Database Recovery option can be used in each of these configurations to further enhance the system resiliency.

4.1 Main / Backup Server Configuration

Overview

A main / backup server configuration consists usually of a single server system running the whole Alliance software (SWIFTNet Link, Gateway and Access combined on one system). The Access database is also installed locally on this server.

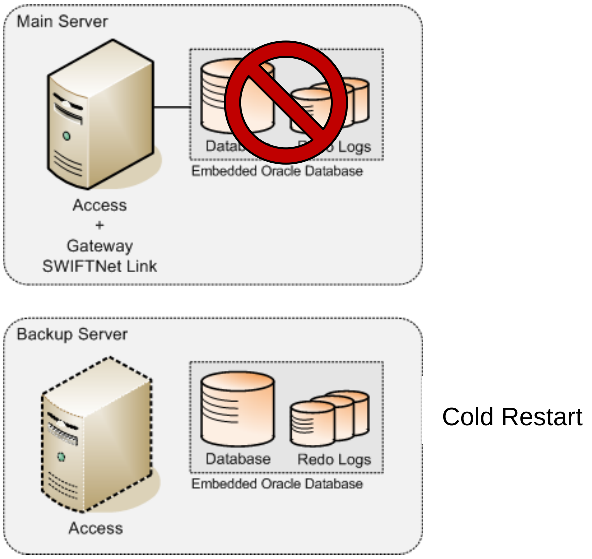
A backup system, with an identical configuration, is available to be able to take over operations in case of failure of the main server. Access Database Backup / Restore tool is used to keep the backup server system configuration synchronized with the main server.



Main Server Failure

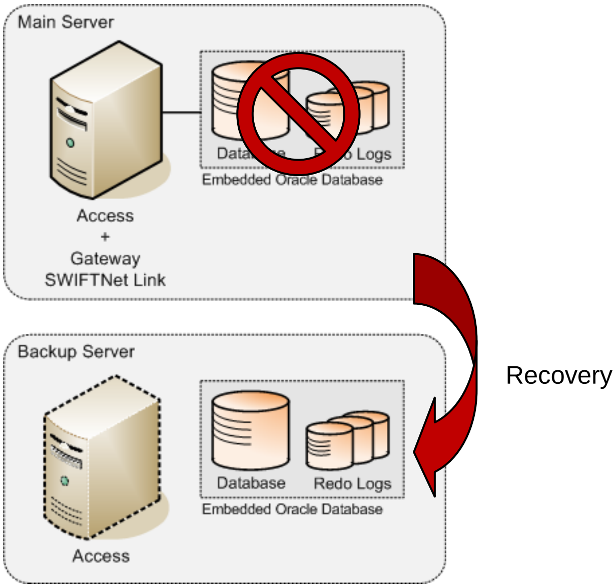
In case of a failure of the main server, this backup server is manually started and back office applications redirect their traffic flows to backup server.

The back office communication is interrupted until the backup server is restarted. All operational data is lost, as the backup server database is maintained using Access Database Backup/Restore tool, which does not include messages and events.



With Database Recovery

If Database Recovery is activated on the main server, it can be used to restore the whole database content on to the backup server, after it has been restarted. In this configuration, it is important that the recovery data is maintained outside the main server. This is to ensure that the data will still be available even when the main server is inaccessible.



The Back Office communication is interrupted until the backup server is restarted and the database is recovered.

In this configuration, operations are resumed on the backup server without any data loss and without any need for PDE's or retrievals.

4.2 Active / Standby Configuration

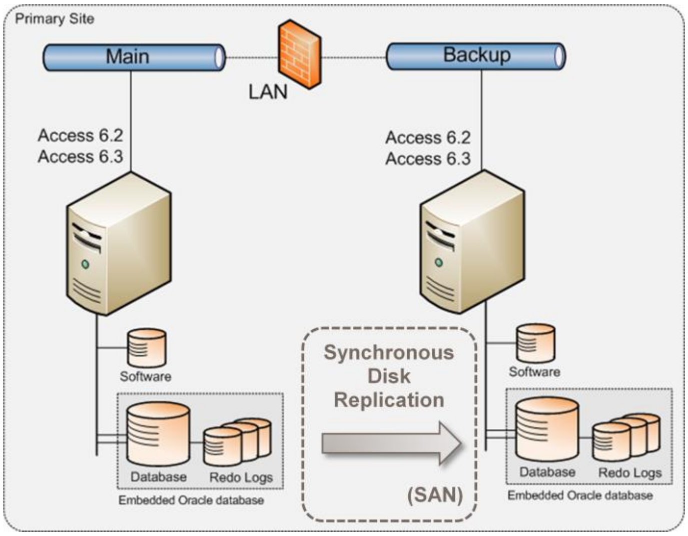
Overview

In an active / standby configuration, the Access system is running on the active site. Its database (and optionally its software) is replicated to a backup site.

The replication is transparent to Access. It is implemented by the underlying file system used by Access. This replication is often provided by a Storage Area Network (SAN) infrastructure.

In this configuration, the active site data is **synchronously** replicated to the standby site, guaranteeing that the data maintained on the active and the standby sites is always identical.

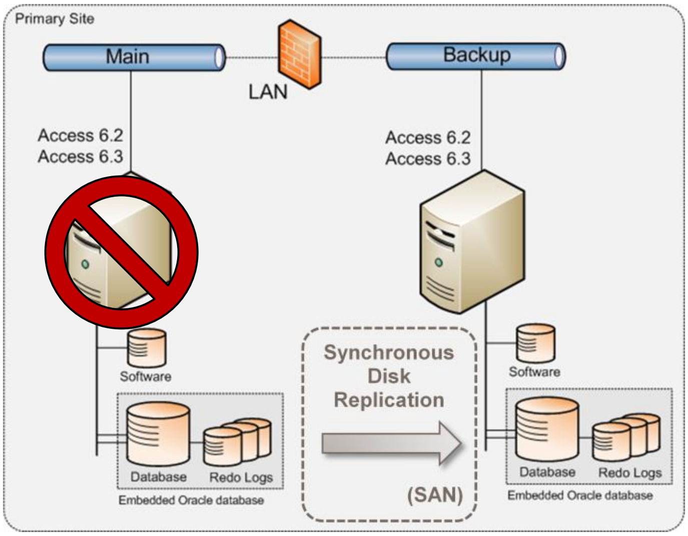
The SAN replication must not affect the overall file system performance and is therefore only possible when the distance between the two sites is limited: typically below 300 kilometres. When the distance is too large, a synchronous replication is not possible, as it would degrade the disk performance too much, and possibly affect the availability and reliability of the system.



Active Site / Access failure

In case of a loss of the Access in the primary site (either due to an Access failure or to a general site failure), operations can be resumed in the standby site.

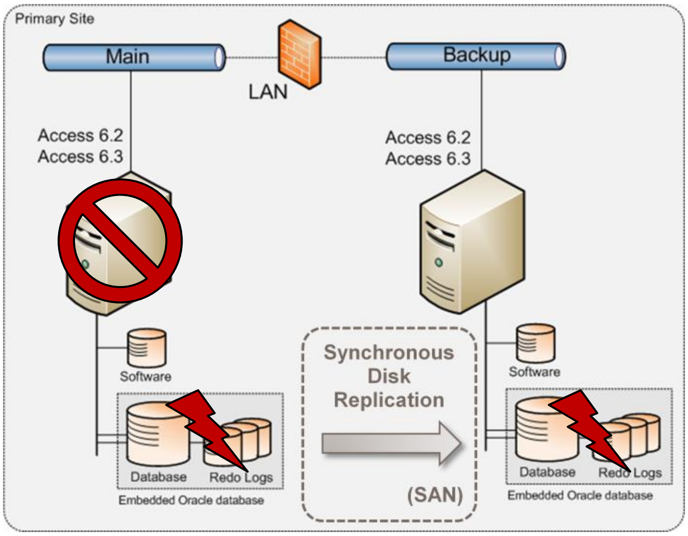
If the issue causing the unavailability of the primary site did not impact the Access database, the Access in the backup site can be activated and will be able to resume operations from the replicated database. In that scenario, operations are resumed on the standby site without any data loss. The back office communication is interrupted until the standby site has been activated and Access has been restarted.



However, if the primary site failure also caused a corruption of the Access database, this corruption will most likely be replicated to the standby site.

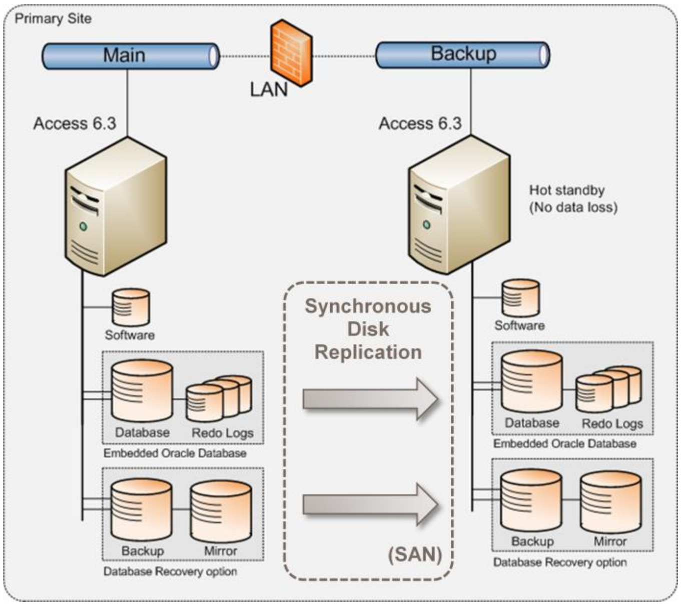
In that situation, the Access database on the standby site is unavailable. A cold restart is therefore necessary (i;e. using the Access Database Backup/Restore tool) to repair the database.

In that scenario, the back office communication is interrupted until the standby site has been activated and the database backup has been restored. All operational data is lost and a manual recovery intervention is needed, resulting in retrievals and resending of messages with PDE trailer.



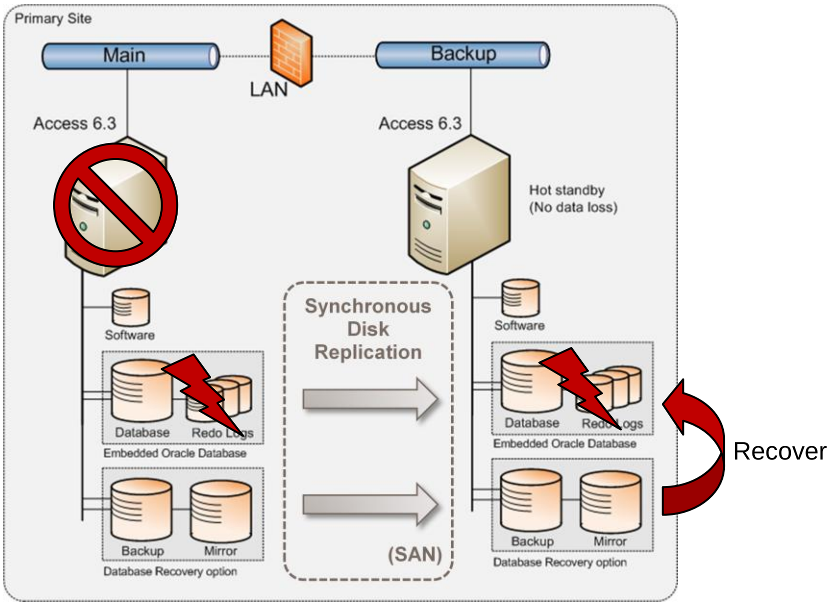
With Database Recovery

To protect against that risk, Database Recovery can be activated on the primary site. The recovery data that it maintains is also replicated to the standby site, relying on the same SAN technology as for the database files.



In case of an incident on the primary site, leading also to the corruption of the Access database on the standby site, the replicated recovery data available on the standby site can be used during the site activation to recover the database content.

In that scenario, back office communication is interrupted until the standby site is activated and the database recovery procedure is completed. Operations are resumed without any data loss or the need for retrievals, nor the need to re-send messages with PDE trailer.



Note The methods used by the Oracle database to maintain recovery data are different from the methods used to maintain the database files.
It is therefore possible that the database becomes corrupted while the recovery data is still un-corrupted and consequently usable for recovery.

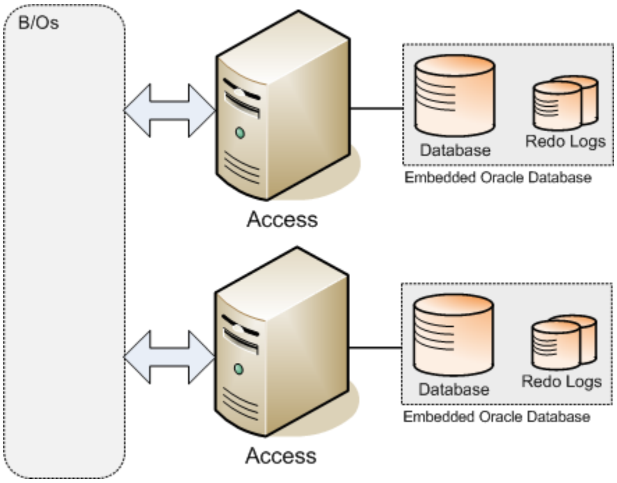
The recovery data available on the primary site can also be used as well to recover a local loss of the Access database (i.e.; the primary site is still available, Access is still operational but the database has been lost).

4.3 Active / Active Configuration

Overview

The active / active configuration is usually the most sophisticated resiliency configuration, used by high volume customers. It consists of multiple active Access systems, running concurrently.

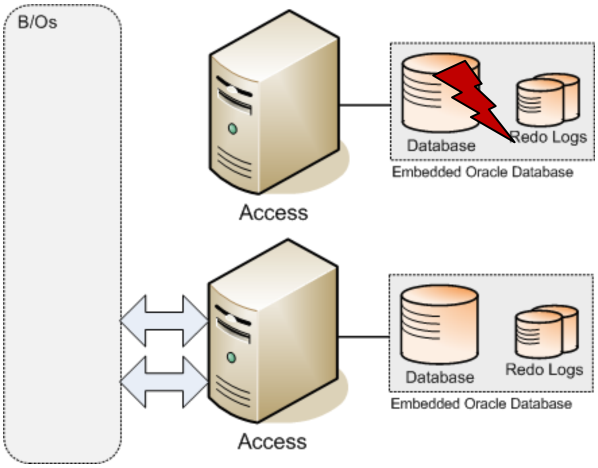
This configuration requires additional logic, in the back office applications communicating with these Access systems, to distribute the message traffic across these Access instances. This logic usually ensures that the traffic load is equally distributed across all available Access instances.



Access Failure

In an Active / Active configuration, the failure of an Access instance does not impact the back office communication. The traffic is dynamically rerouted to the other Access instances still available.

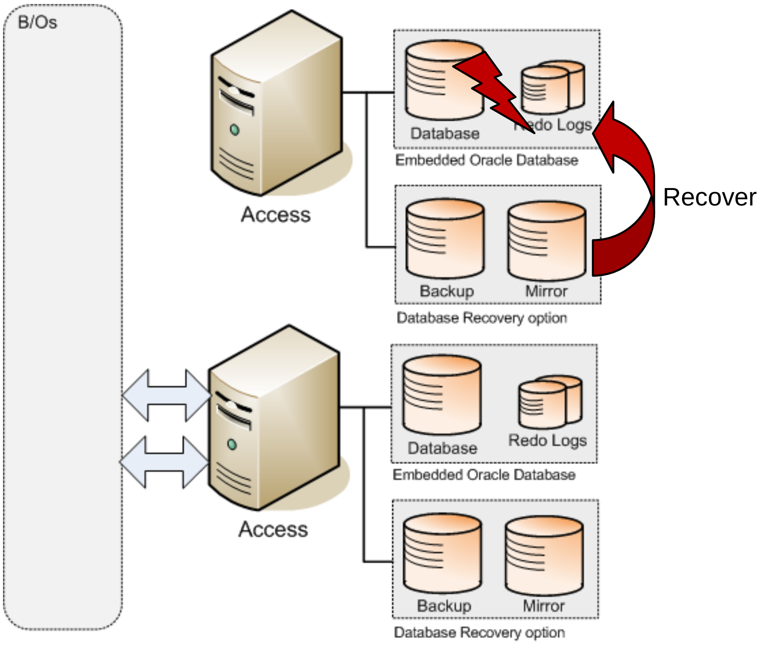
The operational data present in the specific Access instance that has become unavailable is however lost.



With Database Recovery

If Database Recovery is running in each Access instance, it can be used to recover the lost operational data in the failed Access instance, while the new traffic is re-routed to the other available Access instances.

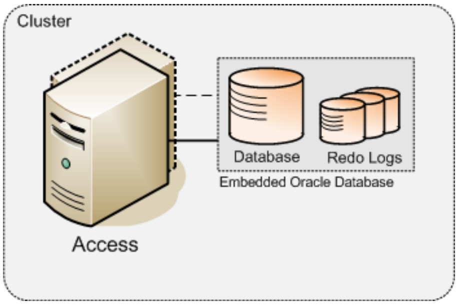
This configuration is the most resilient, as the back office communication is never interrupted and no operational data is lost, nor is there a need for retrievals or resending of messages with PDE trailer.



4.4 Cluster Configuration

Overview

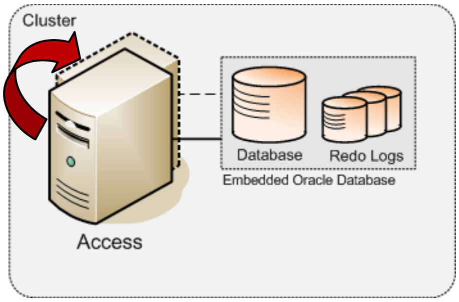
The cluster configuration consists of two Access systems (cluster nodes), sharing the same database. One node is running, while the other node is inactive but can detect a failure of the active node and automatically start to take over operations. As the two nodes share the same database, there is no information loss during the node take over process.



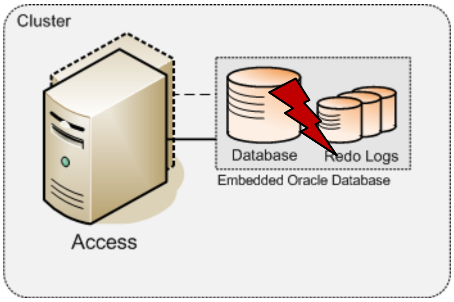
Cluster Node Failure

A cluster configuration is primarily meant to protect against a hardware failure of the server running Access software. The cluster technology will ensure a fast takeover of the other Access server. The back office communication will be interrupted for a short time (the time for the cluster to detect the Access failure and start the other Access system).

As the Access database is shared between the 2 systems, a takeover due to a hardware failure will not cause any data loss.

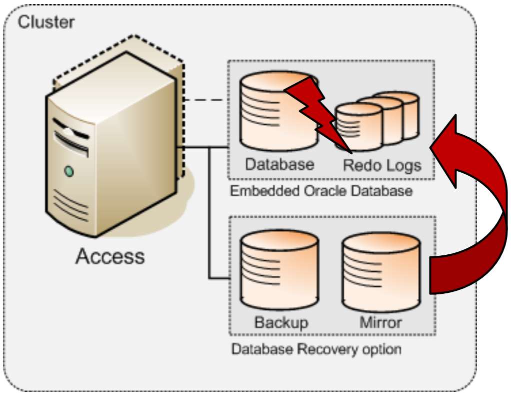


The cluster configuration is however of no use in case of a loss of the Access database (as the database is shared between the 2 nodes). In such an event, a cold restart is necessary.



With Database Recovery

The Database Recovery option can therefore complement the hardware resiliency feature provided by the cluster, by also protecting the cluster against a database loss. The recovery data, also shared by the two nodes of the cluster, can be used to recover the database in that situation.



4.5 Disaster Site Configuration

Overview

The disaster site configuration is often complementary to the other configurations described previously.

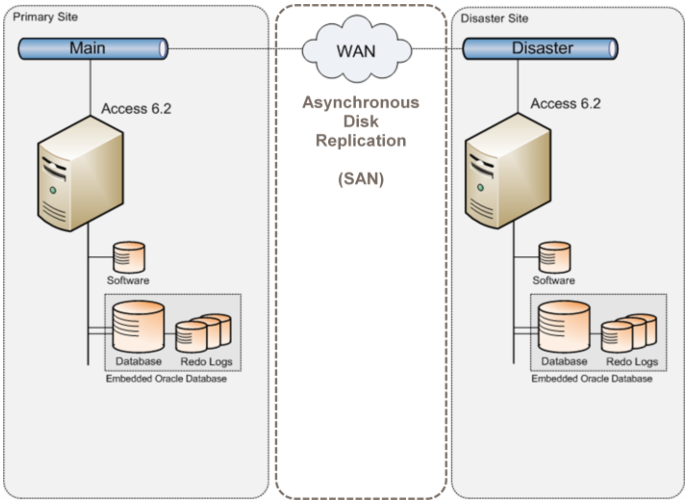
It is implemented by customers willing to maintain a disaster site, located far enough from the primary site. The far distance between the two sites guarantees the availability of at least one, even in the case of major disruptions.

The disaster site will ensure that communication can be resumed even in case of a major incident resulting in the complete loss of the primary site. In such scenario, the disaster site must be activated.

This configuration is somewhat similar to the active / standby configuration, with a major difference linked to the remote location of the disaster site. In this configuration, the distance between the two sites does not allow for a synchronous replication of data from the primary site to the disaster site.

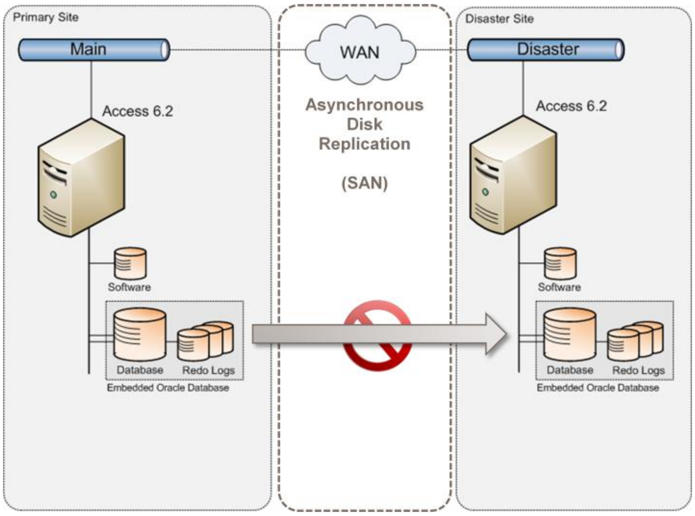
The only acceptable replication must be based on an asynchronous copy of primary site data. It is a compromise between maintaining acceptable disk performance on the primary site, while managing to replicate disk data on the remote site.

With asynchronous replication, the data is not identical between the two sites. There is an inherent time delay before the information generated on the primary site is available on the disaster site. The delay is mainly linked to the quality and speed of the connection between the two sites. This delay can vary a lot, from a few seconds for the most sophisticated infrastructures to a few minutes for less advanced configuration. The delay usually never exceeds 30 minutes.



No Asynchronous Replication of Access database

A core function of the Oracle engine is its database integrity mechanism, which validates at all times that all database files are perfectly consistent. Due to the asynchronous nature of the replication, an attempt to replicate the Oracle database files on the disaster site will technically work, but Oracle integrity mechanism will refuse to use this database, as there is a quasi certitude that the asynchronously replicated database files on the disaster site will be inconsistent.

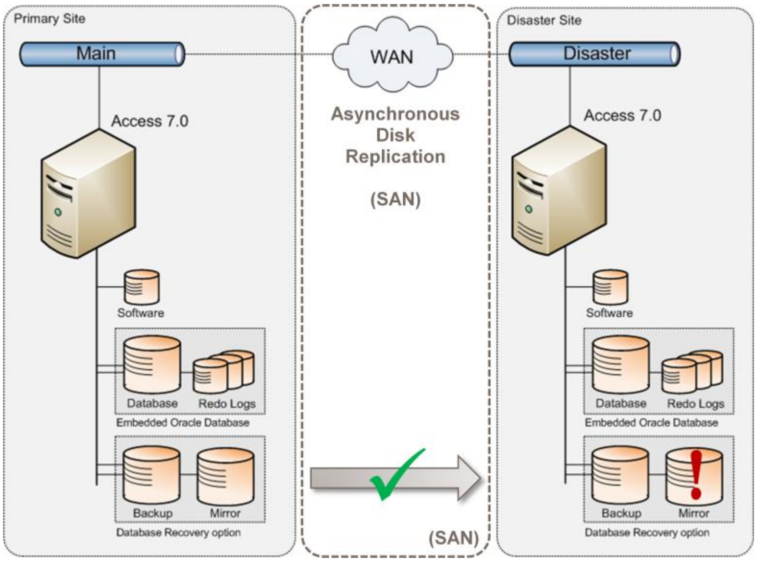


In summary, it is not possible to maintain a replica of the Access database on a disaster site, when using asynchronous replication.

Asynchronous Replication of Recovery Data

To maintain a nearly up to date disaster site, the data in the primary site must be asynchronously replicated to the disaster site.

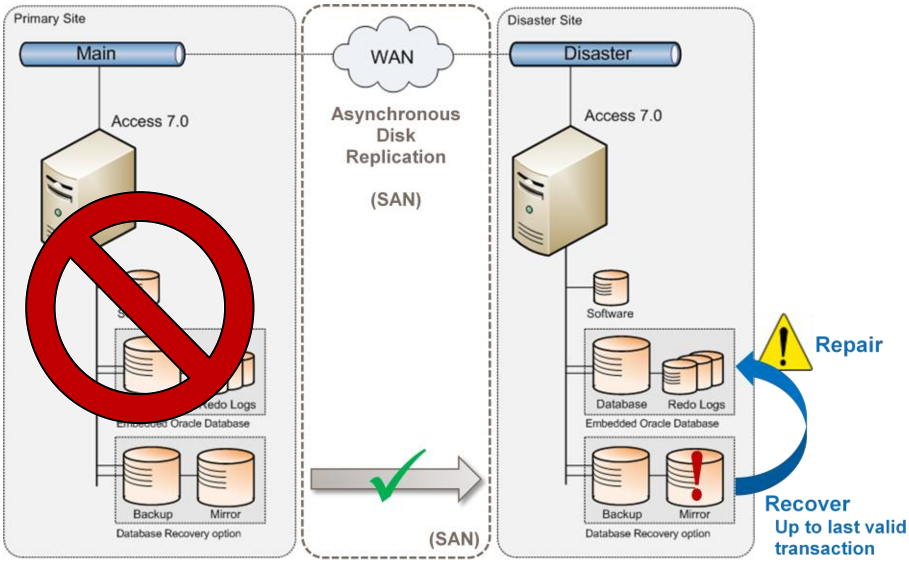
It will be inconsistent, as the last updates done on the primary site will not be available on the disaster site. The amount of information lost will correspondent to the database updates done during the replication delay.



Warning The use of Database Recovery for Disaster site support is available as of Access 7.0 (using the 7.0 new feature to restore a database 'up to the last valid transaction'). On Access 6.3, Database Recovery can only operate with a complete set of recovery data, and is therefore not suitable for a Disaster Site configuration.

Primary Site Failure

In case of a complete loss of the primary site, the disaster site is activated. During the activation procedure, Database Recovery 7.0 new option 'Recover Up to the last valid transaction' is used to restore the Access database.



Database Repair Function

Database Recovery enables the restoration of the database in a consistent state, but missing the last updates done on the primary site. This 'outdated' situation is likely to generate duplicate transactions as messages just completed before the incident, may re-appears as 'live' in the disaster database. If not addressed, the 'live' messages will either be sent again to SWIFT or to the back office applications, leading to duplicate transactions.

To ensure a safe recovery of the database on the disaster site, Database Recovery, when using the 'up to the last valid transaction option' automatically executes a 'repair' process on the restored database. The repair process essentially consists of two steps:

- Add a PDE trailer to all live messages in the database
- Optionally take an additional action on the message:
  - Either move these live messages to an investigation queue to enable an operator to identify amongst these messages the ones that must be sent because they were not completed before the incident versus the ones that have already been executed and must be completed.
  - Or complete all the live messages

During the incident, the back office communication is not available until the disaster site has been activated, the database has been recovered and the repair process is finished.

Operational data will be recovered, but with a small loss of information, linked to the time delay caused by the asynchronous replication of recovery data.

5 Database Recovery Features

This section describes the technical features of the Access Database Recovery function.

5.1 Basic principles

In summary, Database Recovery provides the possibility to maintain ready-to-use on-line backups of database updates on separate disks. Once it is activated and configured, no further action is required to maintain recovery data. Database Recovery provides content recovery, including configuration data and live messages/events.

Database Recovery main functions are as follows:

- Activation of database recovery is done by defining a mirror and backup disk on which information is kept in synch with the live database at all times. Both disks are managed by the Oracle database engine. Oracle keeps on the mirror disk an image of the transaction updates performed on the redo log disk (hence its name, the database engine takes care of mirroring transaction updates on two disks). The updates are not done at the byte level (as a hardware disk mirroring is doing). The transaction versus byte level update guarantees that if the live database becomes corrupted, the probability that this corruption will be propagated to the mirror disk is extremely low.
- Automatic scheduling of recovery backups ensures that regular backups are taken containing all the data from the primary database. These backups can also be generated on request, typically to be included in an external scheduler.
- In case of database corruption, a simple command allows to recover the database to its last committed transaction without any data loss. The information on the mirror and backup disk must be available locally to restore the database content. A remote recovery is possible if the recovery data is also available remotely.

5.2 Activation of database recovery

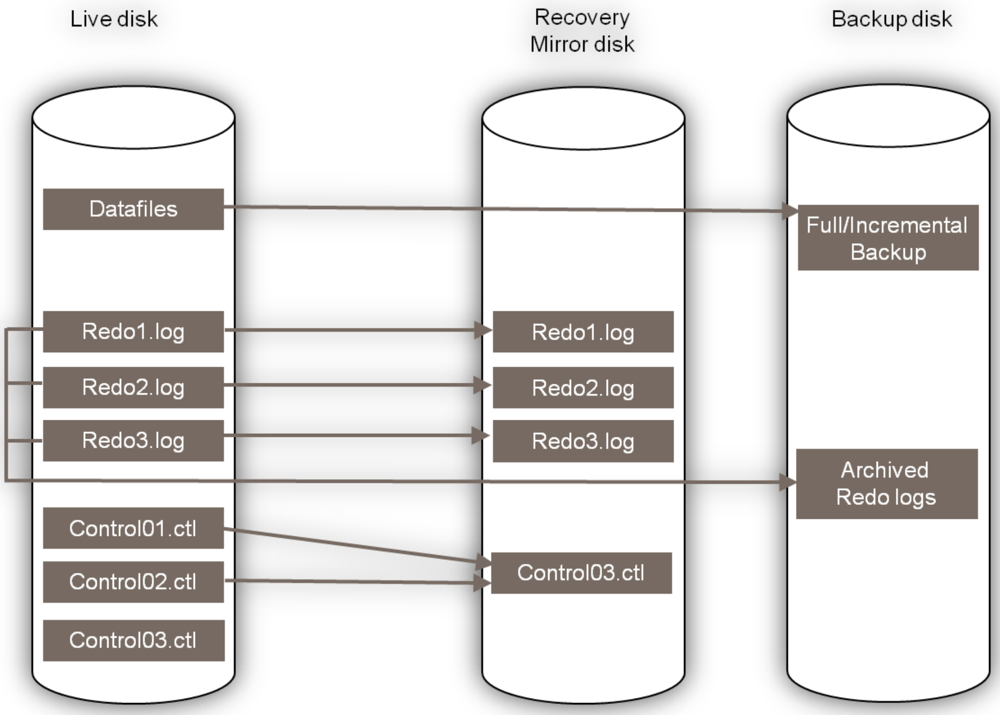
The Oracle database in Alliance Access physically consists of three main elements:

- Data files, containing all the database data.
- Redo log files, which record all the changes made to the database data. In case the datafiles are temporarily unavailable they are used to resynchronize the datafiles.
- Control files, containing information specifying the physical structure of the database.

Database Recovery is activated using a simple Access command (saa_dbrecovery). The path to the mirror and backup disks must be provided at that time. In the background, this means the following mechanisms are activated on the database engine (see graph below):

- The redo logs are mirrored by the database engine on the mirror disk (i.e. the database engine updates simultaneously the active redo log on the live and on the mirror disk). The redo logs contain a log trail of every database transaction that was committed in the Alliance Access database. This information is crucial to guarantee that the database can be restored to its last committed state in a recovery situation.
- The redo logs are archived, i.e. written to the backup disk before being overwritten on the live disk. Redo logs are limited (usually 3) and cyclically re-used, i.e. when a redo log is full, another redo log is used, on the content if the redo log will be at some point overwritten and lost. In order not to lose this data needed for recovery, each redo log is archived on the recovery backup disk when it has become full and another log is used on the primary disk.

- A backup mechanism allows the scheduling of regular incremental and full backups to be taken. As these backups include the information already contained in the archived redo log, the archived redo logs are deleted from the backup disk when the backup is successfully completed on the backup disk.



5.3 Configuration of the recovery backups

After activation, it is possible to schedule incremental and full backups, either on a specific scheduled timing or at a predefined size threshold.

When using the time schedule, the incremental and full backups are taken on a user-defined point in time.

When using the size thresholds, two thresholds are defined:

- The size threshold of archived redo logs, at which point an incremental backup is taken. The incremental backup contains all changes done on the live disk since the last full backup was taken.
- The size threshold of incremental backups, at which point a full backup is taken. This full backup contains all information from the live system and replaces the previous full backup.

It is also possible to take incremental and full backups via scripting, using the command line tool saa_dbrecovery.

5.4 Recovery

Recovery data is available locally, i.e. on the system where the backup and mirror disks are maintained.

5.4.1 Full Recovery

The full recovery is initiated by launching the command line tool saa_dbrecovery, using the 'recover' option.

For the command to succeed, it is mandatory that the recovery data is complete. This will always be the case when using the local recovery data.

The full recovery command will be rejected if it is executed against recovery data that has been replicated, but is not complete (as is the case with asynchronous replication).

During a full recovery, Database Recovery will transparently perform all the following steps required to recover the database up to the last committed transaction:

- Restore of the last full recovery backup
- Restore of the incremental recovery backups, if there are any
- Restore and replay of the archived redo logs, if there are any
- Replay of the redo logs available on the mirror disk

Note that Database Recovery provides the possibility to exclude messages that have been backed up and restored in Alliance Access. As the message information is already available in a separate backup file, and can be restored on an Alliance Access system if needed, there is no operational need to include this information in the on-line recovery data. The main advantage of excluding these messages is to limit the size of recovery data, limiting it to on-line, live information only. The ultimate benefit is to reduce the time needed to perform the recovery operation.

Another option supported by Database Recovery is the compression of the generated recovery backups. This option limits the size of the recovery backup, but will increase the time needed to perform the recovery. It should be considered only when there is a disk space utilisation constraint for recovery data, and it is acceptable to extend the time needed for a recovery operation.

5.4.2 Partial Recovery

The partial recovery command is initiated by launching the command line tool saa_dbrecovery, using the 'recover -v' option.

This recovery mode should be used when the recovery data is not complete. It is therefore the only option available when executing a recovery from a disaster site, using recovery data replicated asynchronously from the primary site.

In this mode, as in the full recovery mode, Database Recovery will start by restoring the available recovery backups (full and incremental). In partial recovery mode, Database Recovery will then locate the last valid transaction available in the redo logs and restore the database up to that transaction.

Database Recovery will indicate the timestamp of the last restored transaction.

After successful completion of the partial recovery, the database will be in a consistent state, but will miss some of the last updates done on the primary database.

Legal Notices

Copyright

Confidentiality

This publication contains SWIFT or third-party confidential information. Do not disclose this publication outside your organisation without the prior written consent of SWIFT.

Disclaimer

The information in this publication may change from time to time. You must always refer to the latest available version on www.swift.com.

Translations

The English version of SWIFT documentation is the only official and binding version.

Trademarks

SWIFT is the trade name of S.W.I.F.T. SCRL. The following are registered trademarks of SWIFT: SWIFT, the SWIFT logo, Sibos, SWIFTNet, SWIFTReady, and Accord. Other product, service, or company names in this publication are trade names, trademarks, or registered trademarks of their respective owners.