Administration Documentation

SAP NetWeaver BI Accelerator High Availability

Target Audience

- System Administrators
- Technology Consultants
- SAP Hardware Partner

Document Version 1.0 – October 31, 2008

Copyright

Icons in Body Text

Icon	Meaning
⚠	Caution
❖	Example
💡	Note
⬆	Recommendation
‹›	Syntax

Additional icons are used in SAP Library documentation to help you identify different types of information at a glance. For more information, see *Help on Help* → *General Information Classes and Information Classes for Business Information Warehouse* on the first page of any version of *SAP Library*.

Typographic Conventions

Type Style	Description
Example text	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation.
`Example text`	Emphasized words or phrases in body text, graphic titles, and table titles.
EXAMPLE TEXT	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
`Example text`	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
`Example text`	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example text>`	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
`EXAMPLE TEXT`	Keys on the keyboard, for example, `F2` or `ENTER`.

BI Accelerator High Availability

- High Availability Concepts [Page 5]

  Describes the basic concepts for BIA high availability.

- Procedures for Enabling High Availability [Page 40]

  Describes detailed procedures on enabling specific BIA high availability features.

High Availability Concepts

Use

This documentation describes the concepts that the SAP NetWeaver BI Accelerator provides for minimizing downtimes and ensuring high availability. It explains how you can minimize or, if possible, avoid downtimes.

For the latest information about SAP NetWeaver BI Accelerator high availability and how to enable it, see SAP Note 984034 BIA 7.00: BI Accelerator High Availability.

Implementation Considerations

High availability is a technically complex area, and implementation considerations vary according to the nature of your system setup. This documentation is primarily intended to illustrate the available options. We cannot offer you a tailor-made high availability solution because you must develop this according to the individual requirements of your business. Therefore, for technical guidance when developing a high availability strategy or when implementing a specific product or feature, contact the appropriate source of information, such as your SAP consultant, your hardware supplier, or the SAP Competence Center.

Disaster-Tolerant Solutions

Disaster tolerance is the ability to restore applications and data within a reasonable period of time after a disaster. Such solutions and strategies are made for disaster situations like earthquake, fire, power blackout, hardware damage (storage, blades, and so on), or any event that unexpectedly interrupts services or corrupts data in an entire data center. A disaster tolerant solution is not a solution for the event of data loss or data corruption. A disaster tolerant solution is needed to cover catastrophes such as power-off or hardware damage. A disaster tolerant solution is also used to guarantee special Service Level Agreements (SLAs). The BIA disaster tolerance allows some data loss, where the BI system has to reindex the data of the indexing requests resulting in failure.

Hardware Dependent Disaster-Tolerant Solutions

The disaster-tolerant solutions for the SAP NetWeaver BI Accelerator consist of specific BI Accelerator blades (disaster tolerance by backup blades) or a complete BI Accelerator system (disaster tolerance by switchover) that are available as standby systems. If a disaster happens, a second BIA backup blade or a complete BIA system take over the tasks of the production system in a short time frame (normally less than one hour). These disaster-tolerant solutions do not cover logical errors or product issues.

Depending on the features of the hardware that the SAP hardware partners provide for the SAP NetWeaver BI Accelerator, there are different disaster-tolerant solutions available:

- Disaster-Tolerance by Switchover [Page 7]
- Disaster-Tolerance by BIA Backup Blades [Page 36]

Generic Settings for Disaster-Tolerant Solutions

Use

For each BIA disaster-tolerant solution you have to configure the following generic settings:

- TCP `keepalive` settings (IBM only)
- Disaster tolerant index writing using `fsync` system calls

  ⚠

  If you do not apply these changes, you would have a high probability of data loss (with the danger of corrupted indexes) in case of a switchover caused by a disaster.

Setting TCP ‘keepalive’ Settings (IBM only)

For the BIA disaster-tolerant switchover solution, we recommend to set the specific `keepalive` time values, despite the TCP connections inside TREX are used without `keepalive` socket option settings ordinarily. You add these values in `sapprofile.ini` on all blade hosts:

1. Start the TREX admin tool (standalone).
2. Navigate to the screen area *Landscape: Ini*.
3. Double-click one `sapprofile.ini` file to edit it.
4. Add the following `keepalive` time values as a new section **`#TCP KEEPALIVE`** in the `sapprofile.ini` file:

   ```
   #TCP KEEPALIVE
   TREX/NetComm/TCP_KEEPIDLE=60 #seconds, 0 = take tcp defaults
   TREX/NetComm/TCP_KEEPCNT=5 #retry counter, 0 = take tcp defaults
   TREX/NetComm/TCP_KEEPINTVL=60  #seconds, 0 = take tcp defaults
   ```

5. Choose the button *Save to All Hosts…* do deploy the changes to all `sapprofile.ini` files of all hosts of your BIA landscape.

Enabling Disaster-Tolerant Index Writing Using ‘fsync’ System Call

1. Edit the `sapprofile.ini` file:
    a. Start the TREX admin tool (standalone).
    b. Navigate to the screen area *Landscape: Ini*.
    c. Double-click one `sapprofile.ini` file to edit it.
    d. Add the following line to the `sapprofile.ini` file:

       `TREX/Persistence/syncOnClose=yes`

    e. Choose the button *Save to All Hosts…* do deploy the changes to all `sapprofile.ini` files of all hosts of your BIA landscape.
2. Edit all `TREXIndexServer.ini` files of your BIA blade host
    a. Double-click each `TREXIndexServer.ini` file of your BIA hosts to edit it.
    b. Add in the section `[delta]` the following line or create the section if it does not exist:

       `usefsync=yes`

    c. Choose the button *Save to All Hosts…* do deploy the changes to all hosts of your BIA landscape.
3. Restart the BIA landscape to assure that the settings take effect.

Background

The BIA failover solution is only defined to be disaster tolerant, and cannot guarantee to work without loss of some data. It assures only that all data, indexed and committed before the disaster, are preserved and available on mirror side after activating the BIA in the data center 2 (mirror). All data missing have to be reindexed by the BI system. Since the operating system buffers file data in kernel memory, BIA software has to use `fsync` system calls to activate flushing of these caches after indexing. The `fsync` system call copies all in-core parts of a file to disk, and waits until the device reports that all parts are on stable storage.

Disaster-Tolerance by Switchover

The BI Accelerator high availability solution *Disaster Tolerance by Switchover* is currently built upon Hewlett-Packard hardware.

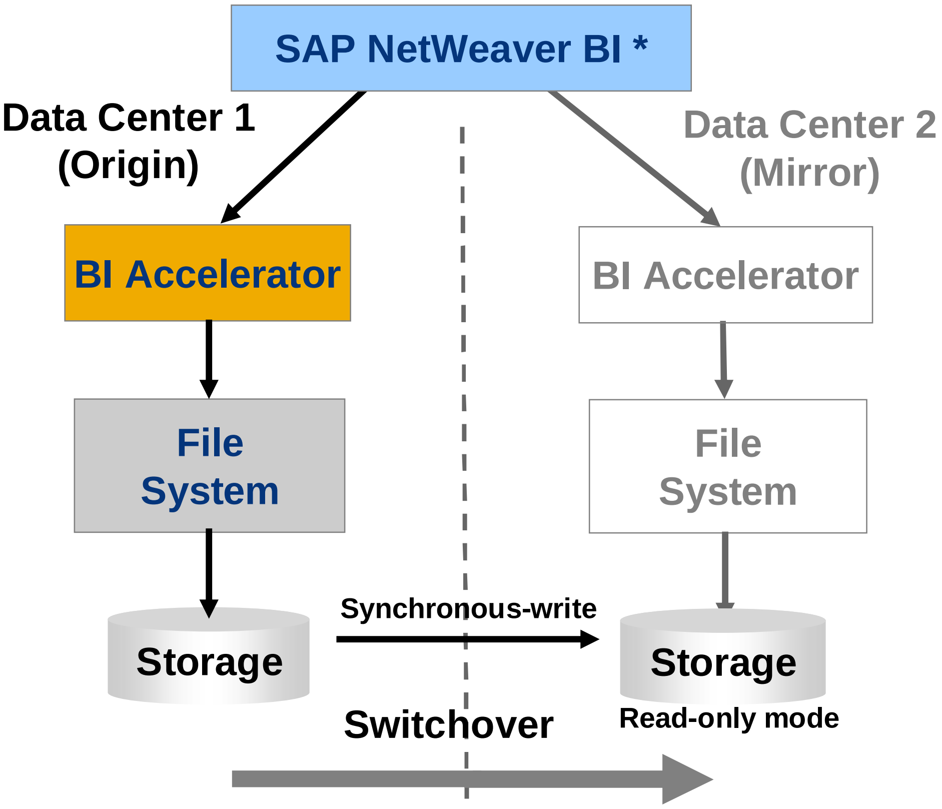
⚠

Note that the disaster-tolerant solutions for the SAP NetWeaver BI Accelerator are still in the pilot phase. Contact your SAP hardware partner if you are interested in implementing one of the named BI Accelerator disaster-tolerant switchover solutions.

The *Disaster Tolerance by Switchover* solution consists of two different data centers with duplicated installations of the BI system and of the connected BIA landscape. Two independent BIA landscapes need to be installed and configured on each data center for enabling this disaster-tolerant solution.

In this documentation the two data centers are referred to as follows:

- Data center 1 (**origin**)
- Data center 2 (**mirror**).



* Software component view without description of BI system high availability

The BI and BIA systems of one data center (origin) are running productive, whereas the other data center (mirror) is switched off and inactive (standby system). There is no phase in the switch-over process where both sides are active. The mass data storage of the active BIA system is continuously mirrored from the active data center to the inactive one by synchronous writing. Synchronous writing is a prerequisite of this solution, so that any data written on the active data center assures that in the other data center exists a consistent view of these data. This duplication or mirroring is done by the storage firmware used for data storing. This high availability solution relies on the Hewlett-Packard data replication software *Continuous Access (CA)*. A dedicated SAN network between the BIA storage systems of both data centers has to be installed to guarantee the needed transfer speed for the data mirroring. In case of disaster or failure a complete switchover from the production data center 1 (origin) to the inactive data center 2 (mirror) can be executed using a switchover script (`switchover.py`).

BI Accelerator Backup Blades

Each BIA installation of the disaster-tolerance by switchover solution has a configuration with BIA backup blades in *shared mode*. The BIA backup blades are an integrated high availability feature of the BI Accelerator. In case one blade fails inside one BIA installation there is no need to switch over. One or more BIA backup blades can take over the load of the BIA blades that are not available in the event of failure. The backup blade contains a standby TREX index server that takes over the tasks of the index server running on the production BIA blade if it is not available.

For more information, see Disaster-Tolerance by BIA Backup Blades [Page 36]

Roles of the Data Centers

The roles of the data centers are defined in the following way:

- Asymmetric: The data center 2 (mirror) is only used while the damaged data center 1 (origin) is being repaired. As soon as possible the customer switches back to data center 1 (origin).
- The customer uses the data center 2 (mirror) until a new problem occurs on data center 2 (mirror), then a switch back to data center 1 (origin) is triggered.
- Symmetric: The customer uses both sides symmetrically, and the roles are exchanged via switchover on regular schedule by the customer, for example, first of each month.

All those preferences are supported by BIA. Since the BIA installations on both data centers are identical, there is no preference for one side. There is no difference in the switch over and switch back actions, since switching back does the same as switching over.

Disk Storage Partitions

The BIA binaries and configuration files on the one hand and the BIA index data on the other hand are stored in two different file systems:

- BIA binaries and configuration files

  BIA binaries and configuration files are shared between the hosts of one blade enclosure in one data center, but **not** mirrored to the other data center. The directory name is `/filer`. This directory contains the BIA program files, the configuration files and the trace (logging) files of the BIA landscape. All files and directories below the path `/usr/sap/SAPSID` are located in this part. The switchover script `switchover.py` and its configuration file are located here also.

- BIA index data

  The BIA index data is mirrored by the Hewlett-Packard data replication software *Continuous Access (CA)* to the other data center. The BIA index data is also shared between all the hosts in one data center. The directory name is `/mirridx`. This part contains all index data of the BIA landscape.

  Continuous Access (CA) is a controller for Hewlett-Packard software for synchronous data mirroring on *Hewlett-Packard StorageWorks EVA (Enterprise Virtual Array)*, which is a family of Hewlett-Packard storage array solutions, providing high availability features.

This partitioned storage design has the following advantages:

- BIA can be installed independently on both sides of the disk storage. Distinct parameter settings can be defined on both sides, without the need to adapt these settings while the switch over is executed.
- The standby BIA on data center 2 (mirror) can be started up while the production BIA is running independently on data center 1 (origin). This can be used for tests of correct working of the standby BIA.

  ⚠

  Note that SAP does not support this specific usage of the partitioned storage design.

Installation and Configuration

Purpose

For enabling the disaster-tolerant switchover solution you install and configure two separate BI Accelerators, one in data center 1 (origin) and one in data center 2 (mirror). The installation and configuration steps have to be performed on both data centers in a similar way.

Prerequisites

- BI Accelerator installation
  - Installation of TREX BIA Revision 44 or higher on a Linux version recommended for this BIA release
  - BIA installation on both data centers with the same SAPSID, instance number and effective user and group ID (the user number of Linux)
- Installation and configuration of Hewlett-Packard software and scripts
  - Installation of an auxiliary script (`/opt/oem_bia/device_rescan.sh`) on all hosts of both BIA landscapes. This script needs to be provided by Hewlett Packard for this disaster tolerant solution supporting the activation of the mirrored storage devices.

    An integration of BIA switchover into a cluster management software with automatic switching over, or with switching over driven by scripts, activated by an administrator, should enforce that the BI system comes up before the BIA is switched over. If the switchover is done fully manually, the guidance documentation for the administrator should also conduct to this starting order.

  - Installation of the controller software *Continuous Access (CA)* by *Hewlett Packard* for synchronous data mirroring on HP *StorageWorks EVA (Enterprise Virtual Array*).
  - The *CA* software has to be configured to run in the *failsafe mode*. A mode-changing into *not failsafe mode* has to be visible in some monitoring environment, so that resetting to *failsafe mode* cannot be omitted.

    Only driving the *EVA* storage in *failsafe mode* guarantees a continuous and synchronous data replication without collecting larger amounts of data blocks waiting for replication. The disadvantage of the *failsafe mode* is that the mirror side cannot be switched off without bringing the origin side in a *hold-on* state. The consequence is than that the productive origin side hangs until the mirror is up again. If the EVA storage is configured to run in *not failsafe mode*, the data written on the origin side is collected in block queues waiting for replication, and a disaster (for example, storage system power outage in the origin side) causes a loss of these data.

    If the customer decides to run the EVA storage in *failsafe* mode, the customer has to switch off the mode manually, to do hardware maintenance on the mirror data center (if the mirrored EVA is affected). In such cases, the *non failsafe mode* has to be visible in monitoring tools (scripts bringing a report, or else a GUI showing system states). This should lead to switching back the mode as soon as possible.

Process Flow

For the sequence of the necessary installation and configuration steps, see Installation and Configuration Checklist [Page 11].

Installation and Configuration Checklist

Use the table below as checklists for the installation and configuration of the system. The checklist guides you through the installation and configuration steps one after another structured by log on as <SAPSID>adm and root user (**DC 1**: data center 1 (origin) / **DC 2**: data center 2 (mirror)).

✓	Activity	BIA hosts	DC 1	DC 2
	Log on as `root` user.			
	Install BI Accelerator	1. host + cloning	x	x
	Grant permissions on sudo to `<SAPSID>adm`	All hosts	x	x
	Check `/etc/sudoers` permissions (`--r--r---`)	All hosts	x	x
	Create `/etc/fstab` entries for `/mirridx`	All hosts	x	x
	Execute **`mkdir /mirridx`** - mount point for mirrored data	All hosts	x	x
	Activate EVA and **`mount /mirridx`** on one DC	1. host	x	
	Execute **`mkdir –p /mirridx/<SAPSID>/ TRX<instance_no>`**	1. host	x	
	Execute **`chown <SAPSID>adm:sapsys /mirridx/<SAPSID>/mirridx/<SAPSID>/ TRX<instance_no>`**	1. host	x	
	Try to mount on all other hosts	2 host …n	x	
	Log on as `<SAPSID>adm` user.			
	Start TREX (if not yet running)	All hosts	x	x
	SSH login preparation: **`python sshkeygen.py`**	1. host	x	x
	Copy `topology.ini` to mirrored data storage	1. host	x	
	Create symbolic link `/usr/sap/<SAPSID>/ TRX<instance_no>/baselink`	1. host	x	x
	Change `basepath` entries in `sapprofile.ini`	1. host + deploy	x	x
	Set `TCP keepalive time` values in all `sapprofile.ini` files	1. host + deploy	x	x
	Enable usage of `fsync` system call in all `sapprofile.ini` files	1. host + deploy	x	x
	Enable usage of `fsync` system call in all `TREXIndexServer.ini` files	1. host + deploy	x	x
	Configure `switchover.py` (as `<SAPSID>adm` user)			
	Execute **`switchover.py --configure`** on DC**1**	1. host	x	
	Transport `switchover.ini` to DC2	1. host	x	
	Execute **`switchover.py --configure`** on DC**2**	1. host		x
	Check `switchover.ini`, adapt settings in this file	1. host		x

	Execute **switchover.py --verify** on DC**2**	1. host		**x**
	Transport switchover.ini back to DC**1**	1. host		**x**
	Execute **switchover.py --verify** on DC**1**	1. host	**x**	

Installing BI Accelerator for Disaster Tolerance

Use

You install the BI Accelerator on both data centers (origin and mirror) with the same SAPSID, TREX (BIA) instance number and user ID (user number).

For the sequence of the installation and configuration steps, see Installation and Configuration Checklist [Page 11].

Procedure

1. Log on as **root** user to the first of your BIA blade hosts.

   For more information about the BIA installation procedure, see SAP Note 875400 BIA 7.00: Install SAPNetWeaver 7.0 BI accelerator.

2. Install one BIA instance on each of both data centers (origin and mirror) in the following way:

   - Location for BIA installation

     Use the storage parts on your *EVA (Enterprise Virtual Array)* that are not mirrored to the other data center. (example /filer) as location for the BIA installation. These locations need to be shared between all BIA installations in one blade enclosure on one data center side and should not be visible in the blade enclosure of the other data center.

   - Identical parameters for the BIA installation on both data centers (origin and mirror)

     You have to configure identical settings for the following parameters:

     - SAPSID
     - TREX (BIA) instance number
     - User ID

       To ensure identical user IDs on both data centers, you have to explicitly define the user ID (user number) in the file /etc/passwd, so that you have the same user ID on both data centers. File ownership on Linux depends on the user ID (user number), not the user name. Using different user IDs, causes permission denied problems after the switchover, and would require root permissions to change the ownership later on.

3. When asked for a mass storage location on a filer system, enter **/filer**.

   The BIA (TREX) servers start automatically, when the BIA installation has finished.

⚠

Do not execute indexing in the newly installed BIA, until the BIA installation has finished, since the index location changes during the subsequent BIA installation and configuration for disaster tolerance.

RFC Connection Between BI and BIA

The requests from the BI system are sent to the BI Accelerator via an RFC connection to the TREX Rfc Server. This server connects to the local gateways on the BI application servers, using registration mode. The linkage between the BI application and the BIA is defined by an entry in the BI transaction `RSCUSTA`, where the name of the RFC destination to the BIA has to be entered. This RFC destination is created and entered initially during the BIA installation.

The disaster tolerant switchover solution has to assure that if an active BIA falls into disaster or if an active side is deactivated, it cannot be accessed anymore by any request from BI side. Even if the deactivated BIA is started on the passive side for maintenance reasons, the deactivated BIA must not be accessible from the BI system.
This is assured by the following configuration:

Two different RFC destinations are established: one RFC destination is used for data center 1 (origin), and another RFC destination is used for data center 2 (mirror). While the switchover actions executed by the switchover script are performed, the switchover script enters the corresponding RFC destination name into the transaction `RSCUSTA`.

As a result on BIA side two different RFC connections exist connecting to the same BI system, and these RFC connections are visible in the TREX admin tools in both data centers. If both BIA installations are located in different disjoint networks, only the active side has access to the BI system. The RFC connectivity screen of this side can start RFC connection tests, whereas in the passive side, such tests fail.

If both sides have access to the BI system simultaneously, one can check the connections on both sides. It is possible to create these two RFC destinations while installing the BIA, one while installing BIA on origin side, and the other while installing BIA on the mirror side.

While the failover switch actions are performed, the `RSCUSTA` entry is changed. In the same step, the `server program id` defined in this RFC destination is also changed containing the actual date. The comment field of the RFC destination is changed also, so it reflects that it derives from a switchover action. This gives the opportunity to document the failover event also in transaction `SM59`.

Configuring Basic Settings on both Data Centers

To enable both BIA installations on data center 1 (origin) and data center 2 (mirror) for the disaster-tolerant switchover solution you have to configure specific basic settings.

- Granting Superuser Administration Rights to <SAPSID>adm User [Page 14]
- Preparing Remote Login with Secure Shell (ssh) [Page 15]
- Creating Mount Point to Mirrored Storage Part [Page 16]
- Define BIA Index Location on Mirrored Storage Part [Page 16]
- Copying Configuration File ‘topology.ini’ [Page 18]
- Creating Symbolic Link ‘baselink’ [Page 20]
- Change ‘basepath’ Locations in ‘sapprofile.ini’ [Page 22]

Granting Superuser Administration Rights to <SAPSID>adm User

The <SAPSID>adm user needs superuser (root) administration rights to execute specific Linux commands like mount, unmount an so on, without entering the root password. The switchover script switchover.py calls the Linux command sudo to grant those permissions to the <SAPSID>adm user temporarily. To enable the switchover script to execute the sudo command successfully, you have to do the following:

1. Log in as super user (root).
2. On every BIA host of each blade center (origin and mirror) of your BIA landscape add the following lines to the file /etc/sudoers:

```
<SAPSID>adm ALL=(ALL) NOPASSWD: /bin/mount
<SAPSID>adm ALL=(ALL) NOPASSWD: /bin/umount
<SAPSID>adm ALL=(ALL) NOPASSWD: /opt/hpclxeva/bin/clxevarun
<SAPSID>adm ALL=(ALL) NOPASSWD: /usr/sbin/lssd
<SAPSID>adm ALL=(ALL) NOPASSWD: /usr/sbin/hp_rescan
<SAPSID>adm ALL=(ALL) NOPASSWD: /sbin/fsck
<SAPSID>adm ALL=(ALL) NOPASSWD: /bin/fuser
<SAPSID>adm ALL=(ALL) NOPASSWD: /opt/oem_bia/device_rescan.sh
<SAPSID>adm ALL=(ALL) NOPASSWD: /etc/init.d/boot.lvm
```

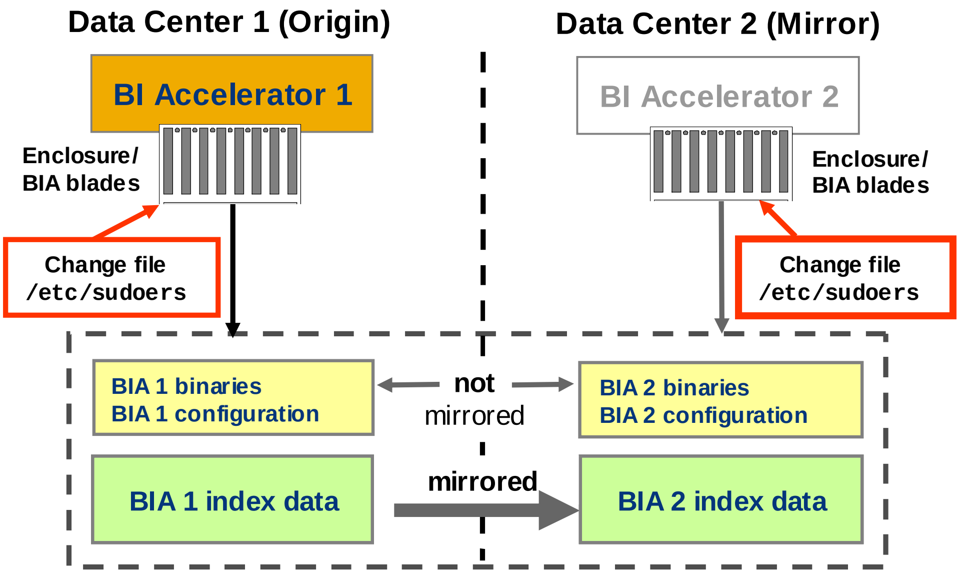
3. Check the permissions in one of the following ways:

   ○ Execute the command **ls –l /etc/sudoers**.

   The result must show the permissions: -r--r-----, otherwise the sudo calls in the switchover script fail. If there are other permissions, execute **chmod 0440 /etc/sudoers**.

   ○ Start the switchover script to check the correct sudoers settings by executing **python switchover.py --verify**.

For the sequence of the installation and configuration steps, see Installation and Configuration Checklist [Page 11].



Preparing Remote Login with Secure Shell (ssh)

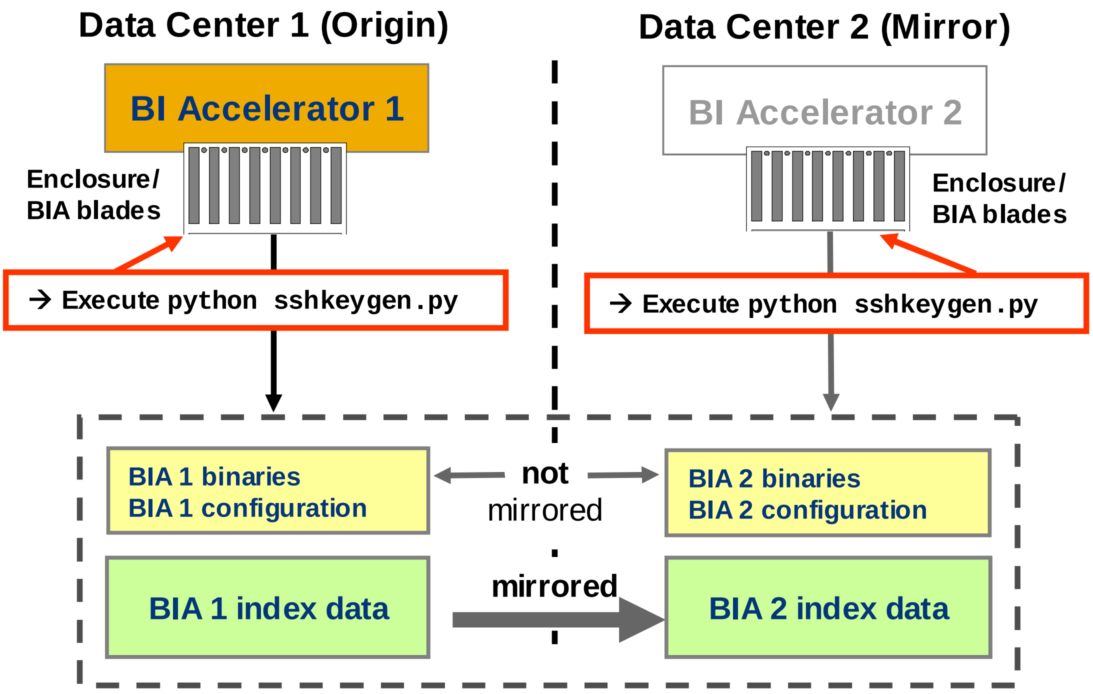
The secure shell (`ssh`) login must work without entering passwords. Otherwise calls of the switchover script `switchover.py` ends in password queries that can obstruct the switchover process.

1. Log in as `<SAPSID>adm` user (BIA administration user) on the first blade host of your BIA landscape.
2. Start the BIA on the whole landscape, if it is not already running since installation.
3. Navigate to the directory `/usr/sap/<SAPSID>/TRX00/exe/pythons_support`.
4. Execute the command **`python sshkeygen.py`** in this directory.
5. Check if the script had been executed successfully by entering **`ssh <TREX_hostname>`** on one of your BIA hosts in the same blade enclosure of one data center.

   The log in should work without questioning for a password.

   The `ssh` preparation allows for starting commands like **`mkdir`** using `ssh` without an interactive login and to use shell loops for facilitating administrative work.

   For the sequence of the installation and configuration steps, see Installation and Configuration Checklist [Page 11].

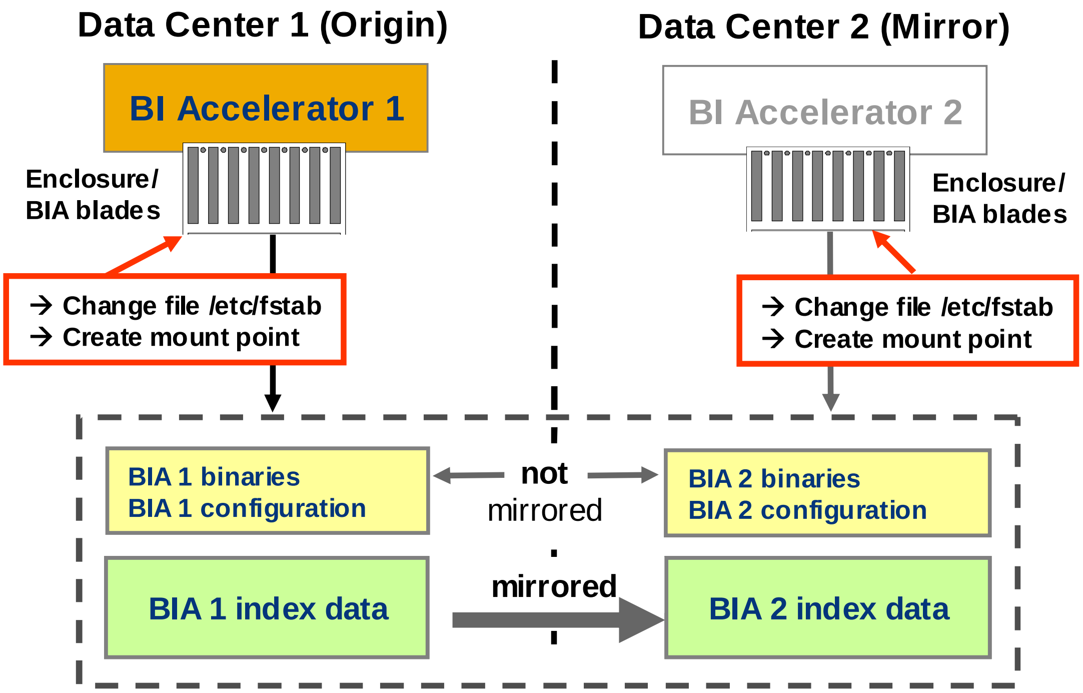


Creating Mount Point to Mirrored Storage Part

You have to create a mount point from all BIA hosts of each data center (data center 1/orign and data center 2/mirror) to the mirrored storage part, where the BIA index data is stored. You do this on all blade hosts of your BIA landscape.

1. Log on as super user (`root`).
2. Create the mount entry in the file `/etc/fstab` on all blade hosts by adding a line like the following: **`/dev/HPBIA/OCFS2    /mirridx    ocfs2    noauto,_netdev 0 0`**
3. Create the mount point for the mirrored storage part (directory `/mirridx`) on all blade hosts.

For the sequence of the installation and configuration steps, see Installation and Configuration Checklist [Page 11]



Define BIA Index Location on Mirrored Storage Part

You have to define the location, where the BIA index data is written on the mirrored disk storage part (disk storage part that is mirrored from data center 1 (origin) to data center 2 (mirror)).

1. Log on as super user (`root`).
2. The mirrored storage part must be active on the data center, where you want to do this step.

   If the EVA storage is not activated yet, activate it by the following command:

```
/opt/hpclxeva/bin/clxevarun CLXBIA
```

Replace the CLX application `CLXBIA` with your actual setting.

3. Mount the directory by executing the command **`mount /mirridx`** on one data center only.

   We recommend to use the directory `/mirridx/<SAPSID>/TRX00` (adapted to your individual BIA settings) for the storage location of the mirrored BIA index data.

4. Check if the directory is correctly mounted by executing the following commands:

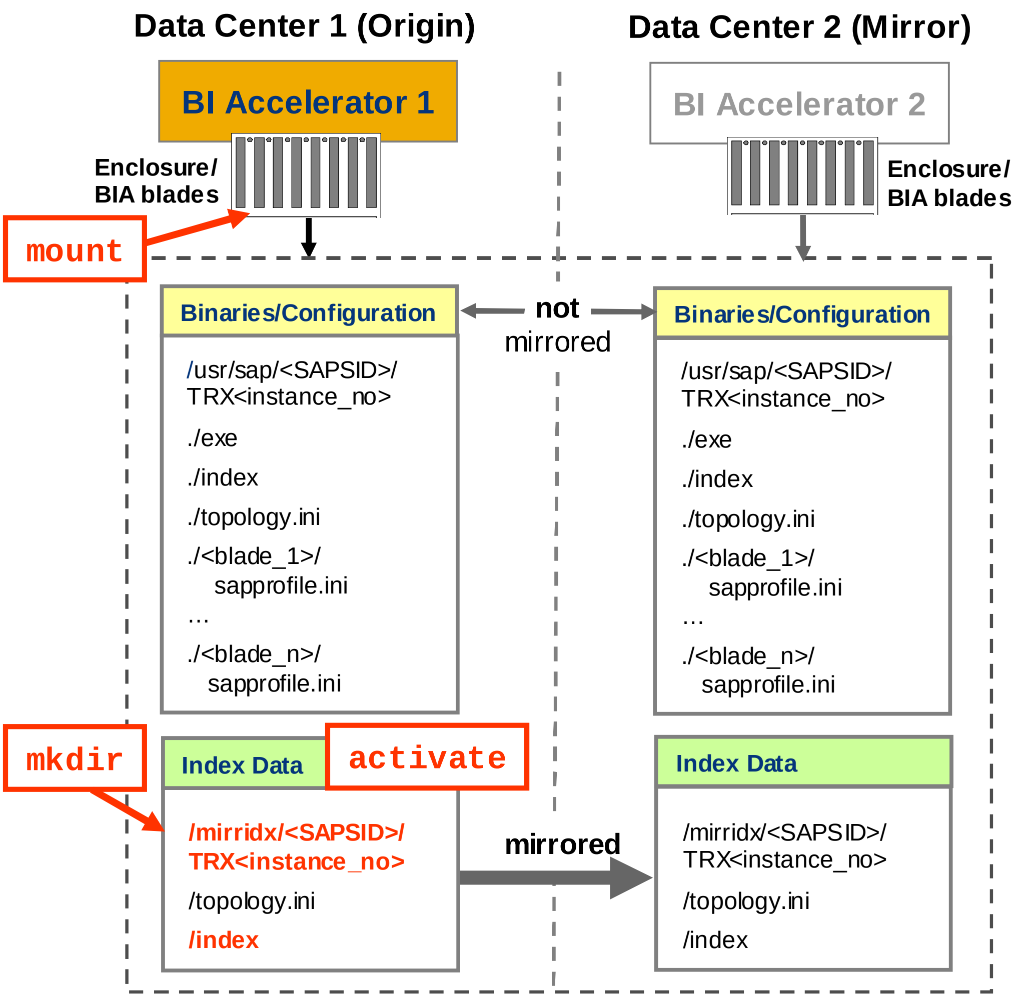
   `mount` and **`ls –l /mirridx`**

5. Create the directory and hand over the ownership of the directory to the `<SAPSID>adm` user by executing the following commands:

```
mkdir –p  /mirridx/<SAPSID>/TRX00

chown  <SAPSID>adm:sapsys  /mirridx/<SAPSID>
/mirridx/<SAPSID>/TRX00
```

For the sequence of the installation and configuration steps, see Installation and Configuration Checklist [Page 11]



Copying Configuration File ‘topology.ini’

The TREX configuration file `topology.ini` describes all BIA indexes and is shared between all blade hosts of one BIA landscape. For enabling the disaster-tolerant switchover solution the file needs to be copied from its regular location in the storage part with the BIA binaries and configuration (`/usr/sap/SID/TRX<instance_no>`) to the data storage part with the BIA index data (`/mirridx/<SAPSID>/TRX<instance_no>`). From this location the `topology.ini` file is mirrored automatically from data center 1 (origin) to data center 2 (mirror) together with the BIA index data.

Procedure

Copy the file `topology.ini` from its original location in `/usr/sap/<SAPSID>/TRX<instance_no>` to the directory chosen for the BIA indexes on the mirrored data storage by executing the following command:

```
cp –p /usr/sap/<SAPSID>/TRX<instance_no>/topology.ini
/mirridx/<SAPSID>SID/TRX<instance_no>
```

For the sequence of the installation and configuration steps, see Installation and Configuration Checklist [Page 11]

Fehler! Es ist nicht möglich, durch die Bearbeitung von Feldfunktionen Objekte zu erstellen.

Background

The configuration file `topology.ini` describes all indexes found in a BIA landscape and is shared between all hosts of the landscape. It contains important information, like master name server definitions and alert server configuration. All this info has to be available on data center 2 (mirror) directly after switchover. Since the BIA blades are not part of a cluster installation, they have different host names on both, the origin and the mirror sides. After switchover, the host names of the data center 1 (origin) mentioned in the `topology.ini` file have to be replaced by their counterparts of the mirror side. Especially the definitions for the master name servers must be correct.

For this purpose, the `topology.ini` file location is migrated from the normal location in the TREX instance directory (`/usr/sap/<SAPSID>/TRX<instance_no>`) into the mirrored data storage part. In our example, this location is `/mirridx/<SAPSID>/TRX<instance_no>`. During the BIA installation and configuration you need to copy the `topology.ini` file to that new location. From this location the `topology.ini` is mirrored automatically together with the index data to the mirror storage part of data center 2 (mirror).

You have installed and configured the BIA as described before and the mirroring between the data storage parts took place. After both procedures the following `topology.ini` files exist in the BIA landscape of data center 1 (origin) and data center 2 (mirror):

- Data center 1 (origin) with active BIA instance:
  - `/mirridx/<SAPSID>/TRX<instance_no>/topology.ini:` This file is used by the BIA running productively on data center 1 (origin).

    - `/usr/sap/<SAPSID>/TRX<instance_no>/topology.ini`: This file remains from the BIA installation process on data center 1 (origin) and is not used while the BIA on data center 1 (origin) is used productively.

- Data center 2(mirror) with inactive BIA instance:

    - `/mirridx/<SAPSID>/TRX<instance_no>/ topology.ini`: This file has been copied during the mirroring process and is not visible.

    - `/usr/sap/<SAPSID>/TRX<instance_no>/topology.ini:` This file remains from the BIA installation process on data center 2 (mirror) and can be used to start the BIA. This should be used only for testing purposes and to check if the binaries are running correctly. The access to this topology is provided by changing the symbolic `baselink.`

The location of the topology.ini file of the disaster-tolerant switchover solution is defined in the configuration file `sapprofile.ini`, with the following line:

```
TREX/NameServer/basepath/shared_topology=/usr/sap/<SAPSID>/TRX<instance_no>/baselink
```

How to enter the value `baselink` into all `sapprofile.ini` files is described here:

- Creating Symbolic Link ‘baselink’ [Page 20]

- Change ‘basepath’ Locations in ‘sapprofile.ini’ [Page 22].

Creating Symbolic Link ‘baselink’

For enabling the disaster-tolerant switchover solution you have to create two symbolic links differently on data center 1 (origin) and data center 2 (mirror).

1. Create a symbolic link **on data center 1 (origin/active)**, where the mirrored storage is visible, by executing the following command:

   ```
   ln –s /mirridx/SAPSID/TRX<instance_number>
   /usr/sap/SID/TRX00/baselink
   ```

   The symbolic link itself, and also the destination directory `/mirridx/SID/TRX00`, are shared and visible on all hosts of the BIA landscape. Hence it needs to be created or changed only once for all hosts of one blade rack.

   Do not change the name `baselink` itself since this name is needed by the script `switchover.py`. The location `/mirridx/SAPSID/TRX<instance_number>` is at your choice. The BI Accelerator on data center 1 (origin/active) can now be used for production indexing.

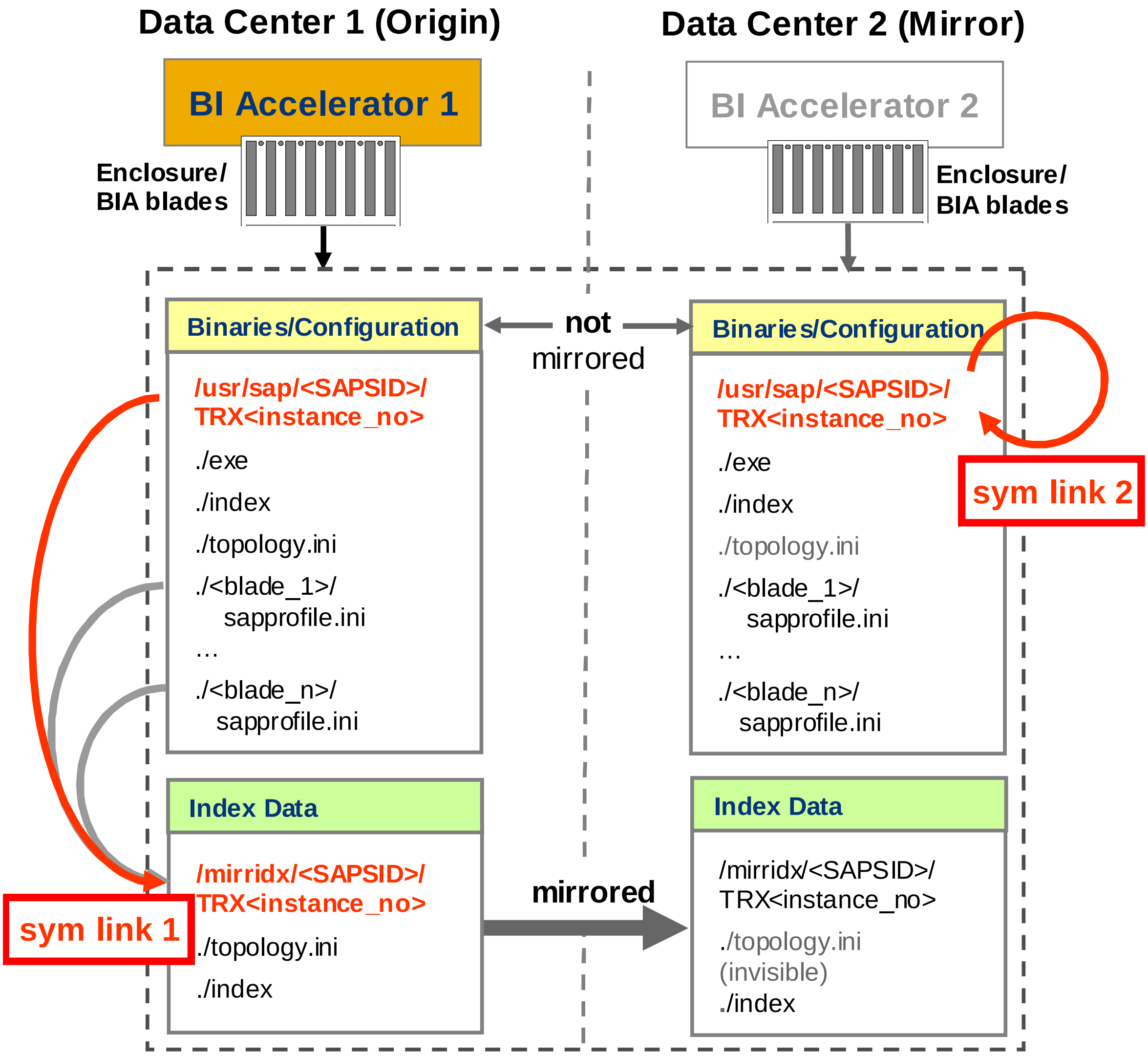
   Do not index production data until the correct functioning of the disaster tolerant switchover solution was tested.

2. Create a different symbolic link **on data center 2 (mirror/passive)**, where the location `/mirridx` cannot be mounted, by executing the following command:

   ```
   ln  –s  ‘.‘  /usr/sap/SAPSID/TRX<instance_number>/baselink.
   ```

   This symbolic link points self-referentially back to the location `/usr/sap/SID/TRX00`.

   For the sequence of the installation and configuration steps, see Installation and Configuration Checklist [Page 11]



Background

The `basepath` values in the configuration files `sapprofile.ini` define important paths to BIA data like the paths to the BIA indexes (`TREX/IndexServer/basepath/index`).

The special entry for the BIA variant of TREX is the location of the `shared topology` (`topology.ini`) defined as entry `TREX/NameServer/basepath/shared_topology`. In standard BIA installations, this location is defined to `/usr/sap/SID/TRX<instance_number>`. For the disaster-tolerant switchover solution you have to change this location so that it points to the mirrored data storage `/mirridx/SID/TRX00`. For this purpose we use a symbolic link located in `/usr/sap/SID/TRX<instance_number>` pointing to the mirror storage directory `/mirridx/SID/TRX00`.

The symbolic link `baselink` pointing `from /usr/sap/<SAPSID>/TRX<instance_no>` to `/mirridx/<SAPSID>/TRX<instance_no>` is added as path component `baselink` to the parameter `basepath` defined in the `sapprofile.ini` file in the configuration step Change 'basepath' Locations in 'sapprofile.ini' [Page 22]. The parameter `basepath` defines paths to directories where important TREX data like indexes, index snapshots, backups and so on are stored. The symbolic link `baselink` itself, and also the destination directory referred to, are visible on all hosts of the BIA landscape in one data center.

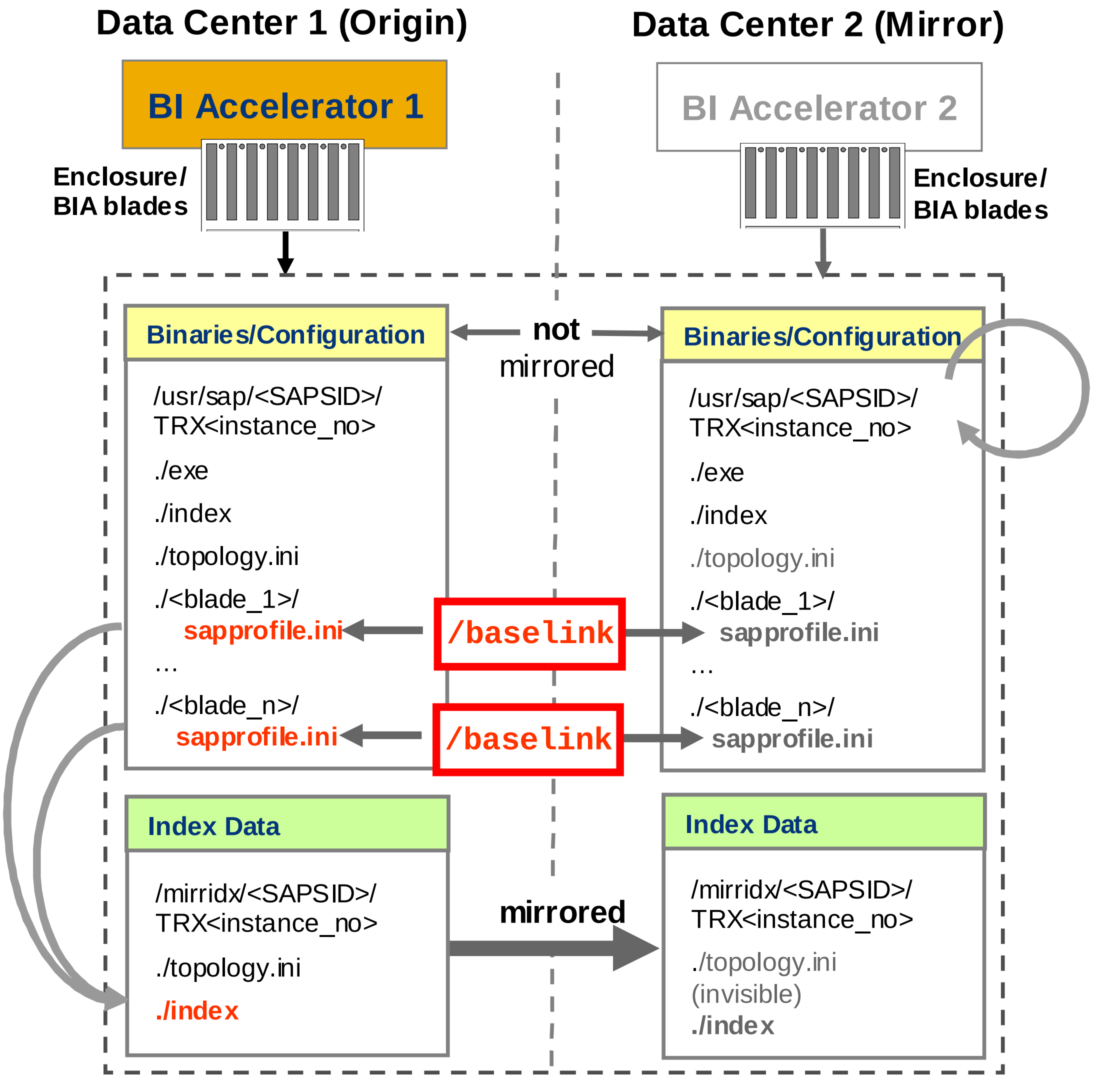
The symbolic link `baselink` allows the following trick: on the activated data center 1 (origin), the symbolic link points to the mirror storage part as described above. On the deactivated data center 2 (mirror), the symbolic link self-referentially points back to the location `/usr/sap/SID/TRX00` itself, so that there can be found a `topology.ini` that does not contain index definitions ('dummy' `topogy.ini`). This self-referential redirect is done automatically using `switchover --passivate` on the origin side. If this is not called, the symbolic link points to nowhere (broken link) and the TREX cannot start. Redirecting the symbolic link baselink self-referentially to `/usr/sap/<SAPSID>/TRX<instance_no>` on data center 2 (mirror) and providing a 'dummy' `topology.ini` there allows you to start the TREX on the deactivated side for test actions. All indexing done here in this state would become invisible after a switchover takes place.

Change 'basepath' Locations in 'sapprofile.ini'

You have to add the newly defined path component `baselink` in all `sapprofile.ini` files on all BIA blade host of the BIA landscape. You do the insertion of the path component `baselink` into the `basepath` values in all `sapprofile.ini` files on all BIA blade host of the BIA landscape using the TREX admin tool (standalone).

1. Log on as `<SAPSID>adm` on one first BIA blade host.
2. Start the TREX admin tool (standalone).
3. Navigate to the screen area *Landscape: Configuration* → tab *Storage.*
4. Select the first host in the list of hosts.
5. Navigate to the parameter *Base Path* in the upper right screen area.
6. In the input field right to the parameter *Base Path* add the path component `/baselink` to the path `/usr/sap/SAPSID/TRX<instance_number>` shown there.
7. Choose the *Set* button right of the input field, to add the path component `baselink` to all `basepath` variables defined in the `sapprofile.ini` of the selected host.
8. Choose the *Deploy* button at the bottom of the screen, to deploy the changes to all other BIA blade hosts of your BIA landscape.

For the sequence of the installation and configuration steps, see Installation and Configuration Checklist [Page 11]



Background

In the TREX configuration file `sapprofile.ini` the parameter `basepaths` is defined. The parameter `basepaths` defines paths to directories where important TREX data like indexes, index snapshots, backups and so on are stored. Each BIA blade host has its own `sapprofile.ini` file including `basepath` definitions. So that each BIA blade host knows where the BIA index data are stored, the path component `baselink` needs to be added to the `basepath` definitions of the `sappprofile.ini` files of each BIA host. The symbolic link `baselink` points to the directory in the mirrored data storage part where the BIA indexes are stored. You have created the path component `baselink` before as symbolic link in the step Creating Symbolic Link 'baselink' [Page 20] pointing exactly to the place where the BIA index data are stored.

Configuring the Switchover Script

Use

You have to configure the configuration file `switchover.ini` of the switchover script `switchover.py` differently for each data center.

For the sequence of the installation and configuration steps, see Installation and Configuration Checklist [Page 11]

Prerequisites

From the installation and configuration steps Installing BI Accelerator for Disaster-Tolerance [Page 12] and Configuring Basic Settings on both Data Centers [Page 13]) you have provided the following values:

- Mount point `/mirridx` to the mirrored disk storage part, where the BIA indexes and the TREX configuration file `topology.ini` are stored
- Location `/mirridx/<SAPSID>/TRX<instance_no>` inside this storage part used for the productive BIA index data
- You have assured that the *EVA (Enterprise Virtual Array)* storage is accessible on data center 2 (origin).
- You have created the subdirectories `/mirridx/<SAPSID>/TRX<instance_no>`.

  You need to have root user rights to create this directory.
- The directory `/mirridx/SID/TRX00` must be owned and be writeable by the `<SAPSID>adm` user.
- From the installation of the HP data replication software *Continuous Access (CA)*, you retain the CLX `application` name, for example `CLXBIA`. You can check the file `/etc/opt/hpclxeva/conf/UCF.cfg` for the entry `APPLICATION`.

Process Flow

1. Configuring Data Center 1 (Origin) [Page 25]
2. Configuring Data Center 2 (Mirror) [Page 25]
3. Checking Switchover Configuration File [Page 26]

See also

Content and Explanation of Switchover Configuration File [Page 26]

Configuring Data Center 1 (Origin)

1. Log on as `<SAPSID>adm` user on data center 2 (origin).
2. Deploy the switchover script `switchover.py` in the directory `/usr/sap/<SAPSID>/TRX<instance_no>`.

   We recommend this location since the switchover script `switchover.py` expects the configuration file `switchover.ini` in the TREX instance directory `/usr/sap/<SAPSID>/TRX<instance_no>`. The log output of the switchover script is written in the current working directory.
3. With the values provided from section *Prerequisites*, execute the switchover script `switchover.py` to create the configuration file `switchover.ini`:

   ```
   python switchover.py --configure --mountpoint=/mirridx
   --baselink=/mirridx/<SAPSID>/TRX<instance_no> --
   application=CLXBIA
   ```

   This execution writes the 'first half' of the configuration, with the host names of the origin side filled in, but the corresponding host names of the mirror side are still missing. The value settings provided with the options arguments above are not yet filled in for the mirror side. The file is located in the directory where the `switchover.py` file resides.
4. Copy the configuration file `switchover.ini` from the data center 1 (origin) to the data center 2 (mirror). The next call of `switchover.py --configure` issued on data center 2 (mirror) completes it there.

Configuring Data Center 2 (Mirror)

You have to repeat the same configuration steps on data center 2 (mirror) as you have done on data center 1 (origin). You have named your storage in a symmetrical manner (same names for CLX application and mount points).

1. Log in as `<SAPSID>adm` user on data center 2 (mirror).
2. Deploy the switchover script `switchover.py` in the directory `/usr/sap/<SAPSID>/TRX<instance_no>`.
3. Assure that the EVA is visible now on the data center 2 (mirror) by executing the following command: **`clxevarun CLXBIA`**.
4. Execute the switchover script `switchover.py` to complete the configuration file `switchover.ini`:

   ```
   python switchover.py --configure --mountpoint=/mirridx
   --baselink=/mirridx/<SAPSID>/TRX<instance_no> --
   application=CLXBIA
   ```

   This call creates the 'second half' of the configuration file switchover.ini so that it contains now the host name mapping. You may change the mapping by editing this file.

Checking Switchover Configuration File

1. Check the completeness of the configuration by executing the following command both on data center 1 (origin) and data center 2 (mirror): **python switchover.py --verify**

   The check should return with a ‘green’ result. Otherwise, the error messages are found in the log file `switchover.verify.<datum>.log`.

   When you call the switchover script `switchover.py` with the option `--verify` it checks the following:

   - Consistency of the configuration file `switchover.ini` in the existing landscape
   - Completeness of the `/etc/sudoers` entries
   - Switch off of switchover steps

     If you get some questions for a password while running this check, the execution of the `sshkeygen.py` step described above was not complete.

2. Copy the switchover configuration file `switchover.ini` from data center 2 (mirror) back to data center 2 (origin) and overwrite the already existing file `switchover.ini` you have created before. You now have two identical switchover configuration files on both data centers.

3. Execute the check again with **python switchover.py –-verify** on the data center 1 (origin).

Explanation of Switchover Configuration File

Example Content

```
[operation]
yellow=0
extensions=_origin,_mirror

[system]
bi=HAT
sid_origin=DEV
instance_origin=02
mountpoint_origin=/mirridx
baselink_origin=/mirridx/DEV/TRX02
application_origin=CLXBIA
rfcdest_origin=TREX_DEV_ORIGIN
sid_mirror=DEV
instance_mirror=02
mountpoint_mirror=/mirridx
baselink_mirror=/mirridx/DEV/TRX02
application_mirror=CLXBIA
rfcdest_mirror=TREX_DEV_MIRROR

[hosts_origin]
host001=hpbiac41
```

```
host002=hpbiac42
host003=hpbiac43
host004=hpbiac44
host005=hpbiac45
host006=hpbiac46
host007=hpbiac47
[hosts_mirror]
host001=hpbiac61
host002=hpbiac62
host003=hpbiac63
host004=hpbiac64
host005=hpbiac65
host006=hpbiac66
host007=hpbiac67
[parallel]
ping=0
stop=0
fuser=0
remount=0
umount=0
fsck=0
start=0
kill=0
mirror=0
[lazy]
ping=0
stop=7
fuser=9999
remount=0
umount=0
fsck=0
start=0
kill=9999
mirror=0
rfc=1
[run]
ping=1
stop=1
kill=1
fuser=1
prepare=1
mirror=1
hp_rescan=1
lssd=0
lvm=0
remount=1
umount=1
fsck=1
start=1
rfc=1
[timeout]
ping=0
stop=120
fuser=30
remount=0
umount=0
fsck=0
start=180
```

```
kill=30
mirror=0
rfc=300
```

Explanation of Sections and Entries

- **[operation]**

  This section defines, how the script `switchover,py` operates: entry `yellow=1` if the result `YELLOW` is allowed indicating a correctly working system, in the case of check operations, but where operator interaction is needed to bring it in a normal state. Return code `YELLOW` is mapped to `GREEN` if the entry is `yellow=0`. The entry `extensions=_origin,_mirror` defines the distinction of both data center sides in the rest of this configuration file. You could change it for example to `_source` and `_dest`, so the section and entry names may be renamed, for example `mountpoint_origin` to `mountpoint_source`.

- **[system]**

  The entry `bi` defines the `SAPSID` of the BI system, to which the BIA landscape is connected. This entry is needed, if the BIA has been connected to more than one BI system for testing purpose and to indicate the correct BI system, because a switchover makes changes to only one of the RFC destinations found.

- **[hosts_origin]/[hosts_mirror]**

  This sections define the host mapping between the data centers. The administrator may take another ordering than the alphabetical order, by changing the numbers in the `host_<number>` entries.

- **[run]**

  This section allows to switch off or switch on some of the actions executed by the switchover script.

  💡

  If you have switched off actions for testing purpose, switch on these actions again for regular production work.

- **[lazy]**

  This section defines, if failing of an action of some hosts may be tolerated, without creating a `RED` check result. The number here is the number of tolerated failing hosts.

- **[timeout]**

  This section defines some timeouts: if a timeout is given, the script waits only this time for the termination of an action (on each host), and than proceeds with counting this as an error (counting against `lazy` values).

- **[parallel]**

  This section defines for all actions performed on all hosts of the landscape, if those actions are done in parallel, or in a sequentially one by one. One can do the stop action in parallel, because it takes some time to finish it. We recommend to do all actions sequentially, so you need to set all actions in this section to the value `0`.

Testing Installation and Configuration

Use

For testing the installation and configuration of the disaster-tolerant switchover solution you execute all steps that are necessary for a real disaster event taking place on the active data center 1 (origin).

You execute test runs for the following scenarios:

- Switchover for disaster situations
- Scheduled switchover for standard working conditions

Test Run for Switchover for Disaster Situations

The test switches over from the active data center 1 (origin) to the passive data center 2 (mirror). If this switchover succeeded, you switch back from data center 2 (mirror) to data center 1 (origin).

After configuring the switchover script `switchover.py` the mirrored storage part is active and mounted in a read-write status on data center 1 (origin). BIA is running both on data center 1 (origin) and data center 2 (mirror). But on data center 2 (mirror) BIA runs based on the `topology.ini` file located in the directory `/usr/sap/<SAPSID>/TRX<instance_no>`.

Start Indexing on Data Center 1 (Origin)

1. Log on as `<SAPSID>adm` user on one of the blades on the active data center 1 (origin).
2. For testing purposes, start indexing from a command line prompt on data center 1 (origin).

   Let the indexing process work for some time.

Start Test Run for Switchover

1. Start the test run for switchover on the passive data center 2 (mirror):
   a. Log on as `<SAPSID>adm` user on the first blade on the passive data center 2 (mirror).
   b. Execute the following command: **`python switchover.py --switch`**

      Add the option **`---debug=4`** to get detailed debug output in the log files.

      If the switchover was executed successfully, the following checks should work:

      - The command **`python switchover.py --switch`** returns the result `GREEN`, which means that the BIA runs correctly.
      - A **`mount`** command shows the directory `/mirridx` as mounted.
      - You see the `topology.ini` file residing physically in the directory `/mirridx/<SAPSID>/TRX<instance_no>`.
      - The command **`ls –ld /usr/sap/<SAPSID>/TRX<instance_no>/baselink`** shows the symbolic link with green color.
      - In the TREX admin tool (standalone), all services in the screen area *Landscape: Services* and all indexes in the screen area *Index: Landscape* shows the status green.

- A search operation started on the index works without an error.
- The index server traces only show messages marked as `error` about rollback operations that have been completed successfully. The rollback messages are always marked as `error`, even if they succeed.

2. Execute a switch back from data center 2 (mirror) to data center 1 (origin) and make sure all checks work correctly on data center 1 (origin) again.

   a. Log on as `<SAPSID>adm` user on the first blade on the active data center 1 (origin).

   b. Execute the following command: **`python switchover.py --switch`**

   If the switch back was executed successfully, the same checks listed above for data center 2 (mirror) should work now for data center 1 (origin) also.

Test Run for Switchover Scheduled for Standard Working Conditions

The test simulates a scheduled switchover. By this test the active BIA on data center 1 (origin) is stopped before starting the switchover. Any indexing needs to be terminated before the BIA is passivated.

1. Log on as `<SAPSID>adm` user on the first blade on the active data center 1 (origin).
2. Passivate the active data center 1 (origin) by executing the following command:

   ```
   python switchover.py --passivate
   ```

   After starting the BIA on the mirror side, rollback actions must not take place, because all actions have been committed and terminated correctly on origin side.
3. Check the symbolic links for correct pointing:

   - On data center 1 (origin), the symbolic link `baselink` must point to the directory `/usr/sap/<SAPSID>/TRX<instance_no>`, so that the `topology.ini` file located there can be used.

     TREX should be able to start on data center 1 (origin) after switchover, started manually by the command **`startsap TRX<instance_no>`**.

   - On data center 2 (mirror), that is activated now, the symbolic link `baselink` must point to the directory `/mirridx/<SAPSID>/TRX<instance_no>`.

Switchover Processing and Details

Executing the Switchover

The following scenarios require switchover execution:

- Switchover in disaster situations
- Switchover scheduled for standard working conditions

Testing the Switchover Solution

After installation and configuration of the disaster-tolerance by switchover solution, you can execute switchover tests as dry runs before starting a switchover for production scenarios.

For more information, see Testing Installation and Configuration [Page 29].

Switchover in disaster situations

If the BI system is not more reactive in case of a disaster you have to start the BIA on data center 2 (mirror) side first by executing the switchover script there.

⚠

Do not stop the BIA on the origin side.

1. Log on as `<SAPSID>adm` user on data center 2 (mirror)
2. Execute the following command:

   ```
   python switchover.py --switch
   ```
3. Check the correct working of the BI Accelerator after the switchover has succeeded using TREX administration tool.

Background

Starting the BIA on data center 2 (mirror) brings the mirrored storage part of the shared storage part instantaneously in a read-only state on the origin side. In the consequence the Linux kernel of the hosts of data center 1 (origin) reboots. As a result all running operations of BIA cannot be committed and the BI system waiting of completion gets error messages for those requests.

All index data that are not completely written to the storage cause correctly rollback operations in the BIA while starting on the mirror side. The rollback operations can be issued for indexes, which are concerned in indexing calls broken by the switchover (or also by the disaster event itself). The behaviour to set the file system to read-only is called *IO-Fencing*.

Switchover scheduled for standard working conditions

In disaster situations the customer is urged to execute a switchover. But also for standard working conditions a switchover can be useful in the following scenarios:

- Functionality tests

  The customer can use this test to check if a switch over works correctly and therefore would also work correctly in a disaster situation. After the switch over, the customer can switch back to the old state performing a switch over in both directions.

- Scheduled change of the roles *origin* and *mirror* of both data centers

After switching over, the data centers remain in their new roles for some time. This is used to proof that the failover scenario is in a correct state and that both data centers are able to run the full workload. This approach uses both installations productively and enhances the workload on mirror side more than only running a test system there. This can help to detect hidden hardware problems.

For enabling a scheduled switchover the BI administrator has to perform the following steps:

1. Stop any indexing activity in the BI system.
2. Log in as `<SAPSID>adm` user on data center 1 (origin).
3. Execute the command **`python switchover.py --passivate`**
4. Check if the passivating has succeeded and if the whole BIA landscape is down.
5. Log in as `<SAPSID>adm` user on data center 2 (mirror).
6. Execute the command **`python switchover.py --switch`**
7. Check the correct working of the BIA after the switchover has succeeded using the TREX admin tool (standalone) or the TREX admin tool in the SAP system (transaction **`TREXADMIN`**).

Background

- For a scheduled switchover, the active running ‘origin’ side has to be stopped correctly before activating the mirror side. To do this, the BI system has to deprecate all indexing activity before stopping BIA. Before starting a switchover, the administrator has to proof that there are no more running indexing requests or similar BI activities concerning the BIA not yet completed. The administrator has to assure that the BI system does not issue any such request in the switching time frame. If there are running activities, the BI system would get error messages from broken requests, and it would issue rollback requests, if the BIA comes back again (on the data center 2).

- Only by stopping before switching, BIA can guarantee that all data written by BIA are correctly sent to the mirror storage. If correct stopping is omitted and there is data lost, the BIA itself can start rollback actions after coming up on the data center 2 (mirror side). This brings the risk that the BIA has another data status now, than the BI system assumes.

- The stopping actions also comprise the unmounting of the mirrored storage part, with the consequence that the file system gets no more writing activity. All system buffer cache is flushed before and no data loss occurs. The unmounting also prevents the operating system of the origin side from issuing a kernel panic with automatic reboot. The operating system continues to work properly.

- Both steps, to stop BIA and to unmount the file system, are performed automatically using the script `switchover.py` with option `--passivate`.

- After stopping the BIA on origin side, the mirror side can be activated.

- After switching over, the EVA storage is mounted on data center 2 (mirror side), and it cannot be mounted anymore on the origin side (data center 1), it is deactivated there.

Details of Switchover Script

The switchover script `switchover.py` performs all switchover and check activities for the disaster-tolerant switchover solution using different script options. During the installation you install the switchover script `switchover.py` and its configuration file `switchover.ini` on the BIA landscape.

Starting the Switchover Script

1. Log on as `<SAPSID>adm` user.
2. Start the switchover script in one of the following ways:
    - Execute **`python switchover.py`** with the appropriate options.
    - Call the switchover script remotely by `ssh`:

```
ssh <sapsid>adm@host  “python switchover.py –check”
```

Return Codes and Log

The script `switchover.py` has two return modes to designate its results:

- Traffic lights `green`, `yellow`, and `red`.
    - `Green`: All checks are OK
    - `Yellow`: The system can be used, but some checks failed without serious problems, like, for example, one host is unavailable.
    - `Red`: The system cannot be used. Checks indicating serious problems failed. For example, during the switch over the mirrored storage part becomes invisible on data center 2 (mirror); or too much BIA hosts are unavailable.
- Only `red` or `green:` all system states, in which the system even works with some constraints, are considered to be `green`. All system states, for which the return mode *traffic light* above would return `yellow`, the check returns `green` also. But warning messages in the log files of the script signify these states.

The return mode is controlled by the configuration file `switchover.ini`. By default only the two traffic lights `red/green` (`[operation]yellow=0/1`) are enabled. The return codes are: `green=0`, `yellow=1, red=2`. The script writes one line to the command prompt, containing the traffic light value in upper case and mentioning the log file name, where more information can be found. If a trace output for debugging is used (option `--debug=4`), all output is written to the standard output.

The result summary of the script `switchover.py` is written in one line on command standard output. This line and all other messages created while the script runs are written to the log file, including trace output. The log files written to are named differently depending on the action performed. On all actions except the action `check`, a log file named `switchover.<action>.<datum>.log` is created on each request, for example, `switchover.switch.2007-08-06-19_45_27.log`. On `check` actions, the messages are appended to the same log file with named `switchover.check.log`.

Switchover Actions

When you start the switchover script as **`python switchover.py --switch`** on data center 2 (mirror) the following actions are performed:

- `get landscape info`

  → detects actual hosts of the BIA installation (host names)

  An error in this step causes the script to fail (result `red`).

- Executes simple checks of configuration settings in `switchover.ini` file

  An error in this step causes the script to fail (result `red.`)

- Checks accessibility

  → `pings` all other hosts than the local host of the BIA landscape

- Stops all instances of a BIA landscape

  The instances may be running for test purposes (based on dummy `topology.ini` file).

  → Lazy entries will be set to the number of hosts: stopping may also fail, if the landscape is already stopped

- Kills remaining BIA processes on all hosts

  → cleans shared memory

  → gets rid of all processes surviving, so the `/mirridx` device can be mounted again

- Unmount mountpoint (`/mirridx`) on all hosts

- Executes main switching action

  → issues the system command `clxevarun`

  → makes mirrored storage part available on the actual side

  The mirrored storage part will become unreadable on the opposite data center.

- Prepare: calls the HP OEM-specific script `/opt/oem_bia/device_rescan.sh`

  → assures the BIA to be able to access the file system. The content of the script `device_rescan.sh` is in the responsibility of HP.

- `fsck on /mirridx`

  → performs an OCFS file system check, in case the opposite storage part has been left in an inconsistent state (this may be the case after a power outage or missing blocks of the Linux file cache).

- `mount /mirridx` on all BIA hosts (`mount` option `read-write`)

  → makes the mirrored data storage part available and usable for the BIA of the activate data center

- Redirects the symbolic link `baselink` to the mirrored data storage part system

  → does in on local host only, but it is visible on all hosts of the BIA landscape

- Adapts `topology.ini` to show host names of the activated side

  → does it on local host, but it is visible on all hosts of the landscape.

  This step transposes also the actual master name server definitions.

- Changes RFC destination settings (changes the program ID)

→ secure against using dangling active connections of data center 1 (origin).

This step needs access to a running BI system and changes all `TREXRfcServer.ini` files to reflect the new program ID.

- Starts the BIA instances on all blades of data center 2 (mirror) that has been activated now.

Result

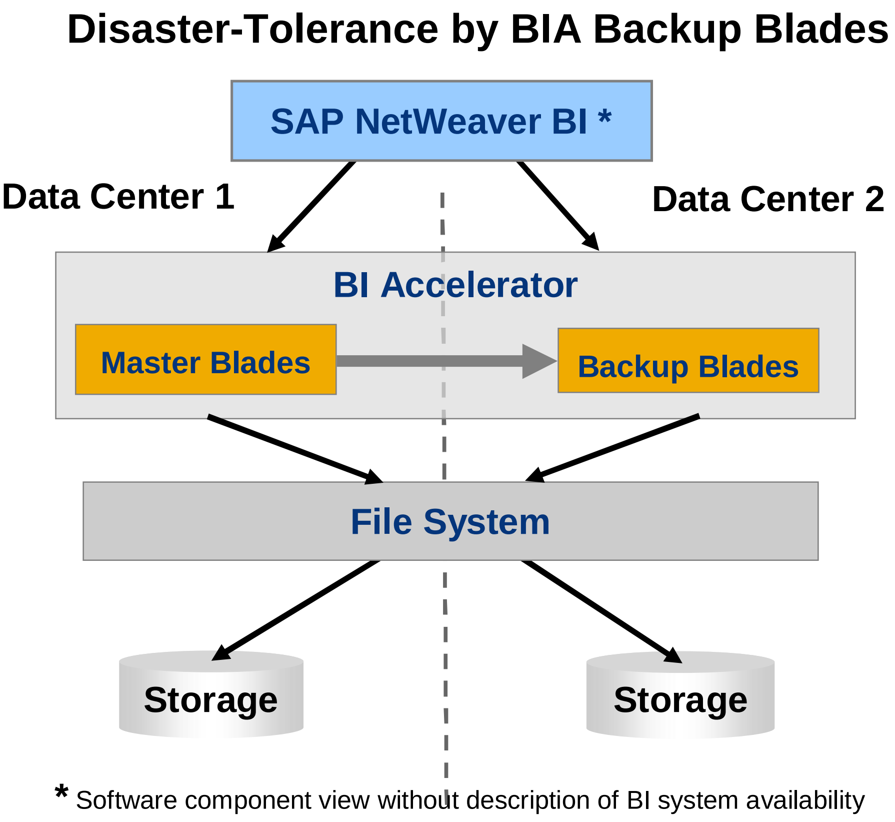
If all these switchover actions are performed successfully, the switchover script ends with the return code `0`, the output line begins with the text `GREEN`.

The script `switchover.py` can be called several times with the same option `--switch` without causing problems. The only impact is that an already started BIA will be stopped and after performing all switchover steps will be restarted again.

If the switchover process has been broken and something needs top be fixed on activate data center, then you should not start the BIA manually, but call `switchover.py --switch` again. The script completes the missing steps then. Especially the step that changed the `topology.ini` would be otherwise omitted with the consequence that your BIA could not start.

Disaster-Tolerance by BIA Backup Blades

The BI Accelerator high availability solution *Disaster-Tolerance by Backup Blades* uses the high availability feature of BIA backup blades. For enabling this feature you configure TREX index server as backup index server for one or more TREX master index servers. The backup index server – now working as BIA backup blade – takes over the indexes from the master index server in case the master server becomes inactive (for more information, see BI Accelerator Backup Blades [Page 37]).


Disaster-Tolerance by BIA Backup Blades
SAP NetWeaver BI *
Data Center 1
Data Center 2
BI Accelerator
Master Blades
Backup Blades
File System
Storage
Storage
* Software component view without description of BI system availability

The BI Accelerator high availability solution *Disaster-Tolerance by Backup Blades* is currently used on IBM hardware. For this solution the TREX index server is configured as dedicated backup index server for exactly one TREX master index server (for more information, see Configuring BIA Backup Blades [Page 39]).

Note that the disaster-tolerant solutions for the SAP NetWeaver BI Accelerator are still in the pilot phase. Contact your SAP hardware partner if you are interested in implementing one of the named BI Accelerator disaster-tolerant switchover solutions.

Prerequisites

- Need for access of all BIA blades from both sides to a sufficient network connection to the BI system
- All blades on both sides configured in one single IP network

- Sufficient network bandwidth needed between the two blade centers

For more information about the IBM solution, see SAP note 1178661 High Availability for BI Accelerator on IBM Hardware.

Features

- Based on IBM hardware and IBM General Parallel File System (GPFS) via Storage Area Network (SAN)
- Redundant 1:1 hot standby of BIA hardware (one side contains BIA master blades - one side contains BIA backup blades)
- No switchover script and no cluster manager integration necessary
- BI and BIA landscape with one SAPSID
- Due to GPFS data replication, continuous data availability without disruption or manual intervention provided
- Clustering of BI system and virtualization to SAP BI Accelerator through usage of virtual host names and IP addresses

Limitations

- SAN and IP network latency should be low between the data centers
- IP network bandwidth between the blade centers needs to match approximately the cumulative bandwidth of half the blades of one side

BI Accelerator Backup Blades

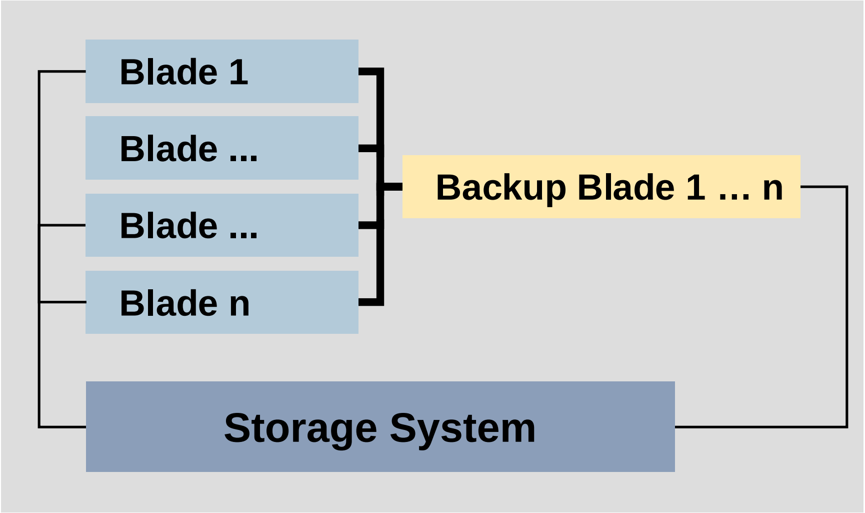
To avoid or minimize downtime, SAP has developed the concept of the BIA backup blade. The idea is to have one or more BIA backup blades that can take over the load of the BIA blades that are not available in the event of failure. The backup blade contains a standby TREX index server that takes over the tasks of the index server running on the production BIA blade if it is not available.

BI Accelerator Backup Modes

BIA supports several backup modes that differ with regard to the assignment between the production BIA blades and the BIA backup blades. The following BIA backup modes are available to ensure the high availability of the BI Accelerator:

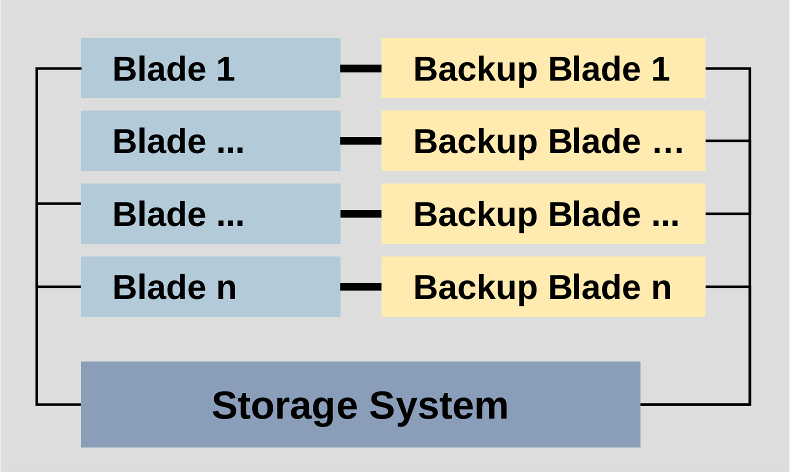
- **Backup Mode Shared: 1:n** assignment, one backup blade is defined as a single backup for several production BIA blades. If one or more blades fail, the backup blades take over their tasks.
  - Benefits and drawbacks:
    - Fully-automated protection against the failure of any blade
    - Load on backup blade increases if more that on blade fails

Backup Mode Shared



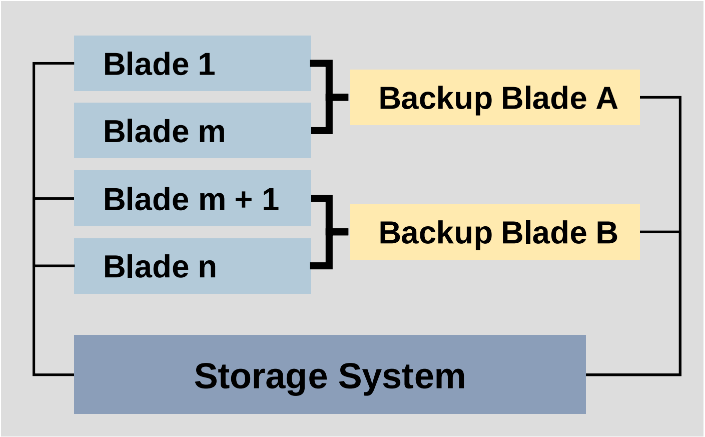
- **Backup Mode Dedicated**: **`1:1`** relationship, exactly one backup index server is assigned to each production BIA blade. There is a dedicated backup blade for each master blade.
  - Benefits and drawbacks:
    - Full protection against failure of any or all blades with no loss of performance
    - Twice as many blades needed

Backup Mode Dedicated



- **Backup Mode Multiplexed**: Two blades, A and B, each back up some of the blades 1 ... n
  - If a blade **`1 ... m`** fails, backup blade **`A`** takes over its workload.
  - If a blade **`(m + 1) ... n`** fails, backup blade **`B`** takes over its workload.

Backup Mode Multiplexed



Configuring BIA Backup Blades

Use

BIA supports several backup modes that differ with regard to the assignment between the production BIA blades (master index server) and the BIA backup blades (backup index server). You use the TREX admin tool (standalone) to configure the backup modes.

Procedure

1. Start the TREX admin tool (standalone) [Page 43].
2. Navigate to the screen *Landscape: Configuration.*
3. Choose the tab *Backup*.
4. Activate the checkbox for *Use Backup Index/Queue Servers i*n the screen area *Scenario* (left on screen).
5. In the screen area *Hosts* a list of all available BIA blade hosts is displayed.

   From that list choose the BIA blade host you want to define as backup index server for one or several master index servers.
6. In the dropdown menu *Backup Mode* (right on screen) you can choose between the following *Backup Modes* for the BIA backup blades (backup index servers):

   - Shared: one backup index server for all master index server (1 → n)
   - Dedicated: one backup index server for each master index server (1 → 1)

     The solution Disaster-Tolerance by Backup Blades is currently used on IBM hardware. For this solution the TREX index server is configured as dedicated backup index server for exactly one TREX master index server.
   - Multiplexed: one backup for several master index server (1 to m → a and m+1 → b)
   - Mutual: each master index server is backup index server for another master index server. So a master index server acts both as master index server and as backup index server.

⚠

The backup mode *mutual* is **not recommended** for BIA landscapes.

Choose the backup mode you want to use for your BIA landscape and choose *OK*.

7. In the screen area *Hosts* with the list of available BIA blade hosts → column *Backup Index/Queue Server* activate the checkbox. Depending on the backup mode you have chosen a list of possible master index server you can assign the backup index server to is displayed.
8. Choose *Deploy* (bottom of screen) to activate the settings.

See also:

BI Accelerator Backup Blades [Page 37]

Procedures for Enabling High Availability

You can enable the following specific high availability features:

- Adding and Removing Hosts
- Cloning a BIA Instance to a New Blade Server [Page 43]

Adding and Removing Hosts

Features

You can use the TREX admin tool (stand-alone) to add or remove a host (server or blade server) to/from a TREX landscape. You do this if you have configured a distributed TREX landscape.

Prerequisites

Make sure that you will still have enough CPU capacity and memory for your TREX landscape after removing a host.

Process Flow

- Removing a Host
  - Removing a Host Temporarily
  - Removing a Host Permanently
- Adding a Host

Removing a Host

You can use the TREX admin tool (standalone) to remove a host from a TREX landscape temporarily or permanently.

Removing a Host Temporarily

1. Go to the *Landscape* → *Configuration* window in the TREX admin tool (standalone).

2. Remove the *Master Index/Queue Server* indicator for the host that you want to remove from your TREX landscape temporarily.

3. Choose *Check* and then *Deploy* to save your change.

4. In the *Landscape Reorg* window, go to the *Plan* tab page.

5. Choose *Start Reorg* to start the required reorganization of your TREX landscape.

   The reorganization process distributes indexes that are located on the removed host to other hosts. When the reorganization is finished, there are no more indexes on the host in question.

   Note

   If you select the *Split/Merge Indexes* checkbox before performing the reorganization, the system not only reorganizes the indexes but also distributes splits the logical indexes again. During this type of reorganization, the system also recalculates the number of parts of which a logical index consists.

   End of the note.

   Caution

   Note that this reorganization can cause a complete reindexing process that can last as long as the initial indexing run. During this period, the system cannot perform indexing runs and searching is limited.

   End of the caution.

Removing a Host Permanently

1. Stop TREX on the host that you want to remove from your landscape.

   The host is highlighted in red as soon as you have stopped it.

2. Go to the *Landscape* → *Configuration* window in the TREX admin tool (standalone).

3. Select the host that you want to remove permanently.

4. Choose *Remove Host*.

   You are asked whether you want the indexes located on this host to be moved automatically.

5. Choose *Move* if you want this to happen.

   The system removes all the indexes from the host in question.

   Note

   After permanently removing a host, do not simply carry out an organization. For performance reasons, you should completely redistribute the indexes. To do so, select the *Split/Merge Index* checkbox in the *Landscape Reorg* window of the TREX admin tool (standalone) and then start the reorganization. During this type of reorganization, the system also recalculates the number of parts of which a logical index consists.

End of the note.

Caution

Note that this reorganization can cause a complete reindexing process that can last as long as the initial indexing run. During this period, the system cannot perform indexing runs and searching is limited.

End of the caution.

Adding a Host

You use the TREX admin tool (standalone) to add a new host (server or server blade) to your TREX landscape.

Procedure

1. Start TREX on the host that you want to add to your TREX system landscape.

   o Install a TREX instance on the server

     If you have not yet installed a TREX instance on the host that you want to add to your TREX landscape, do so before continuing with the procedure.

     For more information about the installation of TREX, see the *SAP NetWeaver Standalone Engine Search and Classification (TREX) Single Host* installation guide. The guide is located on SAP Service Marketplace at `service.sap.com/instguidesnw`.

   o Install a TREX instance on the server blade

     For a distributed TREX installation with server blades, use the `cloneInst.py` script to generate a new TREX instance on the server blade.

     See: Activating the Configuration Clones for Server Blades Go to the ▶ *Landscape* → *Configuration* ◀ window in the TREX admin tool (standalone).

2. Add the server or server blade to your TREX landscape as follows:

   o Following the installation of an additional TREX instance on a server, execute the *Add host* command (see Adding a Host) The `cloneInst.py` script automatically adds the server blade to the landscape

3. Select the *Master Index/Queue Server* indicator for the host that you want to add to your TREX landscape.

4. Choose *Check* and then *Deploy* to save your change.

5. In the ▶ *Landscape* → *Reorg* ◀ window, go to the *Plan* tab page.

6. Choose *Start Reorg* to start the required reorganization of your TREX landscape.

   Note

After adding a host (server or server blade) to your TREX landscape, do not simply carry out a reorganization. For performance reasons, you should completely redistribute the indexes. To do so, select the *Split/Merge Index* checkbox in the *Landscape* → *Reorg* window of the TREX admin tool (standalone) and then start the reorganization. During this type of reorganization, the system also recalculates the number of parts of which a logical index consists.

End of the note.

Caution

Note that this reorganization can cause a complete reindexing process that can last as long as the initial indexing run. During this period, the system cannot perform indexing runs and searching is limited.

End of the caution.

Cloning a BIA Instance to a New Blade Server

Procedure

To clone a BIA instance from an existing blade host to a new blade host, proceed as follows:

1. Check that the filer mount point exists on the new host.

   In a standard BIA installation, the mount point is called `/import`.

2. Execute the `/import/<SAPSID>/SYS/global/cloner.sh` script.

   `<SAPSID>` is the SAP system ID for the BIA instance.

Result

To check that cloning was successful, start the TREX admin tool (stand-alone) and choose *Hosts*

You should see the new host listed with the others.

Starting the TREX Admin Tool (Standalone) for BIA

Prerequisites

On UNIX/Linux: The TREX admin tool has a graphical interface, therefore you need an X server. You cannot use a terminal program that only supports text mode, such as `telnet`.

Procedure

1. Log on with the user `<sapsid>adm`.
2. Perform the following step:

Operating System	Procedure

Linux	Enter the following:
	`cd <TREX_DIR>`
	`./TREXAdmin.sh`

`<TREX_DIR> = /usr/sap/<SAPSID>/TRX<instance_number>`