Evidian

SafeKit Release Notes

High Availability Software for Critical Applications

Overview

Subject	This document provides information about SafeKit releases: major changes, restrictions and known problems, migration instructions.
Content	
Version	SafeKit 7.5
Operating Systems	Windows and Linux; for a detailed list of supported OS refer to 1.1 page 5.
Web Site	Evidian marketing site: https://www.evidian.com/safekit Evidian support site: https://support.evidian.com
Ref	39 A2 19LT 26

If you have any comments or questions related to this documentation, please mail us at
institute@evidian.com

1. Before Starting

This document describes the latest releases of SafeKit. We encourage users of all previous releases to upgrade to the latest release when it is possible.

1.1 Supported Operating Systems

SafeKit **7.5.2** is available for:

- Red Hat Entreprise Linux 9 (Intel x86 64-bit kernel)
- Red Hat Entreprise Linux 8 (Intel x86 64-bit kernel)
- Red Hat Entreprise Linux 7 at least 7.3 (Intel x86 64-bit kernel)
- CentOS 7 at least 7.3 (Intel x86 64-bit kernel)
- Ubuntu 20.04 (Intel x86 64-bit kernel)

In Red Hat and CentOS, the following packages are required:

- ✓ coreutils, sed, gawk, bind-utils for SafeKit core
- ✓ nfs-utils for SafeKit file replication
- ✓ make, gcc, kernel-devel for SafeKit load balancing (and elfutils-libelf-devel if necessary)

- Windows Server 2022 (Intel x86 64-bit kernel)
- Windows Server 2019 (Intel x86 64-bit kernel)
- Windows Server 2016 (Intel x86 64-bit kernel)
- Windows 11 Enterprise (Intel x86 64-bit kernel)
- Windows 10 Enterprise (Intel x86 64-bit kernel)
- Windows 11 Pro (Intel x86 64-bit kernel)
- Windows 10 Pro (Intel x86 64-bit kernel)

The up-to-date list of supported operating systems can be found in the *Software Release Bulletin* and at https://support.evidian.com/supported_versions/#SK. Old SafeKit packages and documentations can be found at https://support.evidian.com/safekit_old.

1.2 Documentation

The latest version of the SafeKit 7.5 documentation can be found at https://support.evidian.com/safekit under Version 7.5/Documentation.

Name	Description
SafeKit Solution	SafeKit solution is fully detailed at https://www.evidian.com/safekit
SafeKit Release Notes	It describes new features of major SafeKit 7,5, 7.4, … releases and provides migration instructions.
Software Release Bulletin	Technical release bulletin for all SafeKit 7.5 packages with the description of changes and problems that are fixed.
SafeKit Knowledge Base	List of known problems and restrictions on SafeKit 7.5, 7.4, 7.3, 7.2, 7.1 and 7.0.
SafeKit User's Guide (english version) *Guide de l'utilisateur SafeKit (french version)*	It covers all phases of SafeKit implementation: architecture, initial use, installation, configuration, administration, troubleshooting, testing and support.  Important The links refer the latest version of the SafeKit User's Guide. For previous versions, refer to the one delivered into the SafeKit package.
SafeKit Training (english version) *Formation SafeKit (french version)*	Refer to this online training for a quick start in using SafeKit.

2.Major Changes

This section gives the list of new features introduced in SafeKit since release 6.2. Go to section 3 page 51 and carefully read known problems about SafeKit releases and to section 4 page 63 for migration instructions.

2.1 Major Changes between SafeKit 7.5.2 and SafeKit 7.5.1

Version 7.5.2 is a consolidation of version 7.5.1 and includes the following changes.

2.1.1 Virtual IP

In Linux, load-balancing for farm module is now based on two kernel modules, instead of one in preceding releases (`vip` and `tcpseq`) . The two kernels' modules must be signed when used with SecureBoot (See Q009176).

Re-introduction of the possibility to designate the interface, on which to configure the virtual IP, by its `name` (necessary in some use cases). The interface name must be identical on all nodes.Supported in Linux **and** Windows.

Add the netmask definition for the virtual IP in Windows, in addition to Linux.

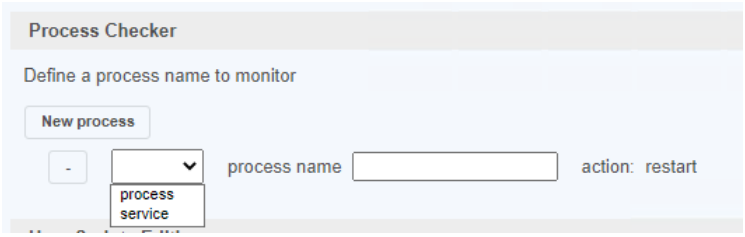
```
<vip [tcpreset="off"|"on"]>
 <interface_list>
  <interface
    [name="interface name"]
  >
  …
  <virtual_addr
```

```
      addr="virtual_IP_name"|"virtual_IP_address"
       [netmask="netmask for the virtual address"]
  …
```

<vip	
<interface_list>	
<interface	
[name="interface name"]	You can specify the name of the network interface on which the virtual IP addresses will be set. Ex.: name="bond0" on Linux. Default: no value, SafeKit detects the network interface with virtual IP addresses set on it.
…	
<virtual_interface	Definition of virtual IP addresses configured on an Ethernet interface.
netmask="defaultnetmask"	IPV4 only. By default, the netmask of the network interface on which the virtual IP address is set. Set a netmask if there are several netmasks on the interface.

2.1.2 SafeKit web console

The module configuration wizard with the SafeKit console now allows you to select the type of monitoring to be performed (process or service) when configuring the process checker



2.1.3 Japanese language support

The Japanese translation has been updated.

2.1.4 Miscellaneous

In Windows, add printcert utility to display the module certificate subject and expiration date. Usage: safekit.exe -r printcert -m AM, where AM is the name of a configured module.

2.2 Major Changes between SafeKit 7.5.1 and SafeKit 7.4.0

SafeKit 7.5 comes with major changes for storing internal data and for authenticating users of the web console and `safekit` distributed command (`safekit -H …`). For migrating to SafeKit 7.5, you must follow the procedure described in 4.2 page 63.

2.2.1 SafeKit install procedure

Since SafeKit 7.5, by default, the web service requires user authentication to access the service to improve security when using the SafeKit web console and `safekit` distributed command (`safekit -H …`). To make the web console and the distributed command operational quickly and easily, now the install procedure requires a new step for initializing the web service with an `admin` user.

For details, see "SafeKit install" in in the *SafeKit User's Guide*. Below is the quick install and setup procedure:

Windows	Linux

a. Log as administrator
b. Double click on the package
`safekitwindows_7_5_y_z.msi`
c. Open a PowerShell console
d. To setup the Windows firewall, run:
`cd SAFE\private\bin\`
`.\firewallcfg add`
e. To initialize the web service with the `admin` user and its password, for instance, `pwd`, run
`cd SAFE\private\bin\`
`.\webservercfg.ps1 -passwd pwd`
	a. Log as root
	b. Open a system console
	c. Run `chmod +x safekitlinux_7_x_y_z.bin`
	d. Run `./safekitlinux_7_x_y_z.bin`
	It extracts the package and the `safekitinstall` script
	e. Run `./safekitinstall`
	⇨ Reply `yes` for firewall automatic configuration (with `firewalld` or `iptables`)
	⇨ Reply with the password, for instance, `pwd` to initialize the web service with the `admin` user

The password assigned during initialization must be identical on all nodes that belong to the same SafeKit cluster. Otherwise, web console and distributed commands will fail with authentication errors.

Once this initialization is done:

⇨ you can authenticate in the web console with the name `admin` and the password you provided. The role is Admin by default.

⇨ you can run distributed command `safekit -H …`

This default configuration can be extended:

✓ to add users and assign them a role

✓ to switch to HTTPS

The default configuration can still be replaced by another predefined setup with HTTP/HTTPS ; no authentication ; authentication based on LDAP/AD server or client certificates.

For details on the default setup and all predefined setups, see section "Securing the SafeKit web service" in the *SafeKit User's Guide*.

2.2.2 Module resources and web console enhancement

Since SafeKit 7.5, text files storage for internal data has been replaced by SQLite database. This evolution offers increased resiliency, protections against corruption while providing scalability and performance. It has also permitted to add new module resources for:

- making visible the internal state of the module (such as failover rules, ..)
- provide performance/usage indicators (such as synchronization indicators, ..)

And finally, it allowed to keep a history of the state of the resources to be able to make an analysis of the evolution of the states over time.

The web console has been revised to provide a lighter and more ergonomic layout. Resource display has been improved to better represent new resources and their history. For details, see section "The SafeKit web console" in the *SafeKit User's Guide*.

2.2.2.1 Mirror module resources

- In Control tab
- Click on the node to display the detailed status of the module on this node
- Select the Resources tab to view the status of resources

Since SafeKit 7.5, the date displayed is the last date the resource was assigned. Before SafeKit 7.5, this is the first time the resource has been assigned to the current value.

- Module state

  Local and remote state ; replication state ; boot and encryption configured or not ; checkers, errd, failover active/inactive (result of commands in the Admin submenu of a node)

- Checkers

  heartbeats, errd, intf, ip, ping, tcp, custom, …

- File replication (since SafeKit 7.5)
  - ✓ Incoming and outgoing bandwidth

    There are 2 new resources that reflect the network bandwidth (in KBytes/sec) used between nfsbox processes:

    - `rfs.netout_bandwidth` is the network output bandwidth
    - `rfs.netin_bandwidth` is the network input bandwidth

You can observe the value of `rfs.netout_bandwidth` on the primary or `rfs.netin_bandwidth` on the secondary to know the modification rate at the time of observation (write, create, delete, …). The history of the resource values gives an overview of its evolution over time.

The value of the bandwidth depends on the application, system, and network activity. Its measurement is available for information purposes only.

✓ Data synchronization metrics

New resources have been added to have metrics on the synchronization (number of files, copied size and time, …). These measurements are for information purposes only and may be inaccurate in some cases. In addition, some are not updated in real time.

⇨ Others

Failover rules (including the active one) ; internal resources

Note

Resources named `rfs_bandwidth.replication` and `rfs_bandwidth.reintegration` have been renamed `rfs.rep_bandwidth` and `rfs.rei_bandwidth`

2.2.2.2 Farm module resources

⇨ In Control tab

- Click on the node to display the detailed status of the module on this node
- Select the Resources tab to view the status of resources

Since SafeKit 7.5, the date displayed is the last date the resource was assigned. Before SafeKit 7.5, this is the first time the resource has been assigned to the current value.

- Module state

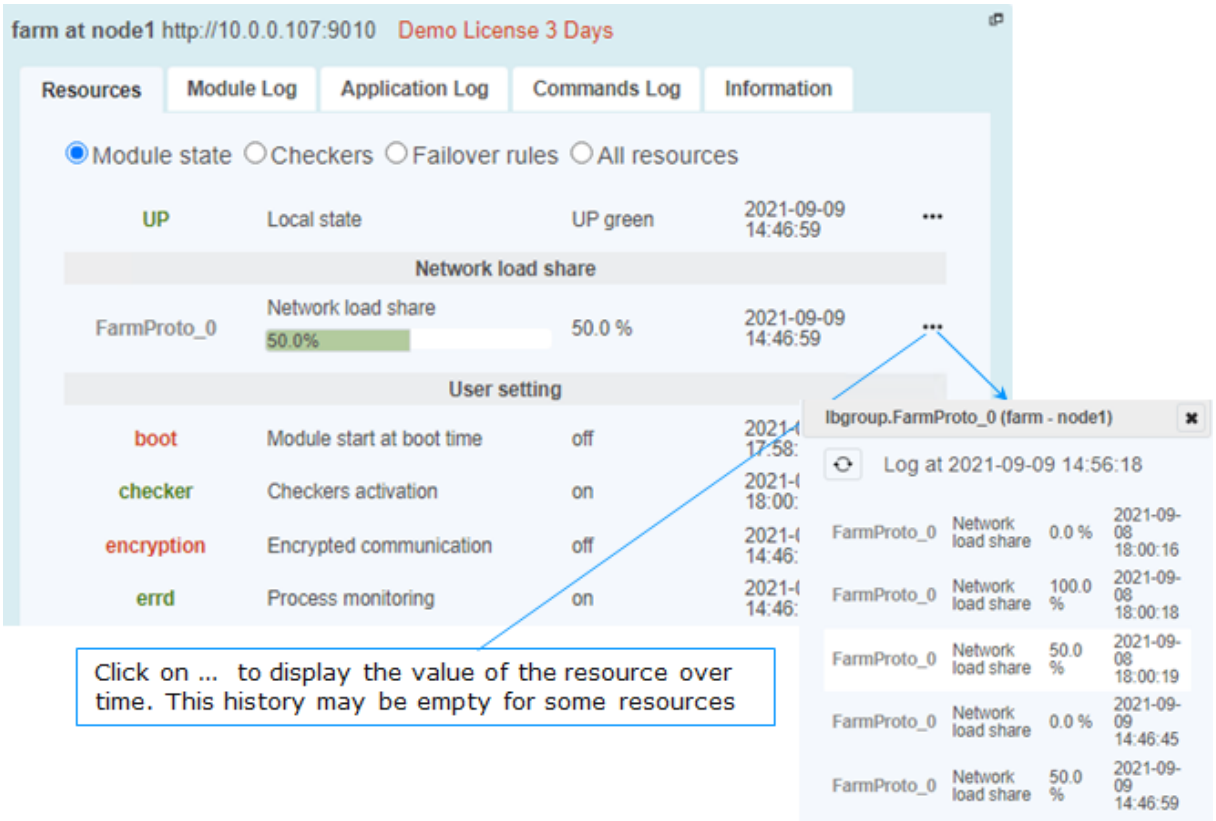
  Local state ; network load share for the node; boot and encryption configured or not; checkers, errd (result of commands in the Admin submenu of a node)

- Checkers

  heartbeats, errd, intf, ip, ping, tcp, custom, …

- Others

  Failover rules (including the active one), internal resources



2.2.3 Module templates

2.2.3.1 Lightweight Kubernetes

Evidian SafeKit brings high availability to Kubernetes between two redundant servers. For details see Kubernetes: The Simplest High Availability Cluster with Synchronous Replication and Failover between Two Redundant Servers – Evidian

2.2.3.2 Custom checker module template

SafeKit 7.5 delivers a new module template, `customchecker.safe`, that is a basic example of a custom checker in a mirror module. Install it with the web console (in Advanced modules) or with the `safekit` command. For details, see section "Custom checker example with customchecker.safe" in the *SafeKit User's Guide*.

2.2.4 New attributes for the module configuration

2.2.4.1 Module boot configuration

A new attribute permits to integrate the boot configuration of the module into its configuration. It can be set with the web console into the configuration wizard or in the XML configuration file of the module:

```
<service mode="mirror"|"farm"|"light"
         [boot="off"|"on"|"auto"|"ignore"]
         [boot_delay="0"]
```

`<service`	Top level section of `userconfig.xml`
`[boot= "on"| "off"| "auto"| "ignore"]`	If set to `on`, the module is automatically started at boot time. If set to `off`, the module is not started at boot time. If set to `auto`, the module is automatically started at boot time, if it was started before the reboot. Before SafeKit 7.5, the configuration to start the module at boot was done with the command `safekit boot -m AM on | off` (which had to be executed on each node). If you prefer to continue using this command, remove the `boot` attribute or set it to `ignore` (the default). The module will not be started at boot time unless the `safekit boot -m AM on` command is executed. The state of the boot configuration is visible in the `usersetting.boot` resource. The status of resources is visible in web console/ Control/Select the node/Resources tab/; with the command `safekit state -m AM -v` Default value: `ignore`
`[boot_delay="0"]`	The delay, in seconds, before starting the module at boot. Default value: `0` (no delay)

2.2.4.2 Counter of active connections on the virtual IP

The `connections` attribute enable metric on the virtual IP. It must be set in the XML configuration file of the module:

```
<virtual_addr addr="virtual_IP_name"|"virtual_IP_address"
[connections="off"|"on"]
```

<virtual_addr	Definition of one Virtual IP address
[connections="off"|"on"]	Enables counting of the number of active connections on the virtual address. This count is stored in the resource named connections.<virtual addr value> (for example: connections.192.168.1.99) which is assigned every 10 seconds. This value is provided as a guideline only. Default value: off

2.2.5 Scripts for the test, debug, or support

Since SafeKit 7.5, when configuring the module, 2 scripts are generated under SAFE/private/modules/AM/bin (where AM is the name of the module):

- AM_start_wrapper (.ps1 in Windows, .sh in Linux)

  It configures the virtual IP address if one is defined into the module's configuration and runs the script start_prim or start_both with all required environment variables

- AM_stop_wrapper (.ps1 in Windows, .sh in Linux)

  It runs the script stop_prim or stop_both, with all required environment variables, and deconfigures the virtual IP address if one is defined into the module's configuration

These scripts can be executed, as administrator/root, when the module is stopped:

- ✓ to test or debug the application start/stop scripts in the module (start_prim/stop_prim, start_both/stop_both)
- ✓ to run the application for support or maintenance purpose

If the start/stop scripts execute a SafeKit command, it may have a different behavior when executed while the module is stopped.

Be aware that starting the application outside of the module may cause application files on that node to change. If these files are replicated by a mirror module, the next time you start the module, please start as primary, the node that has the most up-to-date data from your point of view.

2.2.6 Miscellaneous

- Commands log of the SafeKit node

  Since SafeKit 7.5, this log is stored in SQLite3 format. For viewing the commands log, run the command safekit cmdlog or click on the commands log tab into the web console.

  For more details, refer to section "Commands log of the SafeKit server" in the *SafeKit User's Guide*.

- Module snapshot

  The structure and content of the snapshot has changed in SafeKit 7.5. For a full description, see section "Analysis from snapshots of the module" in the *SafeKit User's Guide*. See also the new SafeKit training resources "Support tools".

- Proxy mode for the web console

  In previous versions of the SafeKit web console, it connected to each cluster node to retrieve their state and run commands. In this new version, the console connects only to the node specified in the URL, which acts as a proxy for the other nodes. This implementation is called `proxy` mode (set the query ?`proxy=false` on the URL, to revert to the previous implementation).

  The proxy mode means that the console becomes inaccessible if the connecting node is unreachable. It is then necessary to change the URL to get the cluster state from another cluster node.

- Command to clean the HTTPS setup

  If necessary, you can use the new command `rmcerts` under `SAFEBIN` to clean your HTTPS setup. It removes all certificates and switch to HTTP mode for the web service

- SafeMonitor

  SafeMonitor, the legacy java console for SafeKit, is no more delivered with the SafeKit package and no more supported.

2.3 Major Changes between SafeKit 7.4.0 and SafeKit 7.3.0

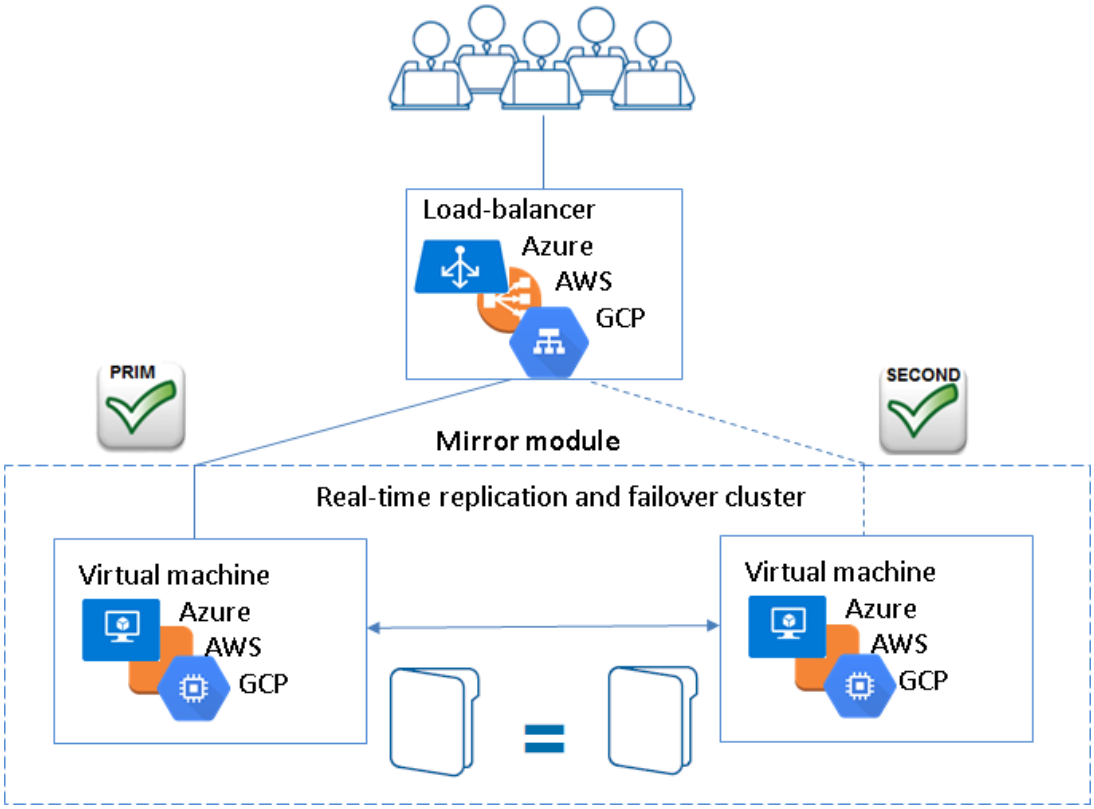
2.3.1 SafeKit cluster in Microsoft Azure, Amazon Aws, and Google GCP clouds

SafeKit 7.4 core and console have been improved for providing the simplest solution for a high availability cluster in the Microsoft Azure, Amazon AWS and Google GCP clouds. It can be implemented on existing virtual machines or on a new virtual infrastructure, that you create by simply clicking on a button that deploys and configures everything for you in Azure or AWS clouds.

For a full description, see in section 17 SafeKit Cluster in the Cloud in the *SafeKit User's Guide*.

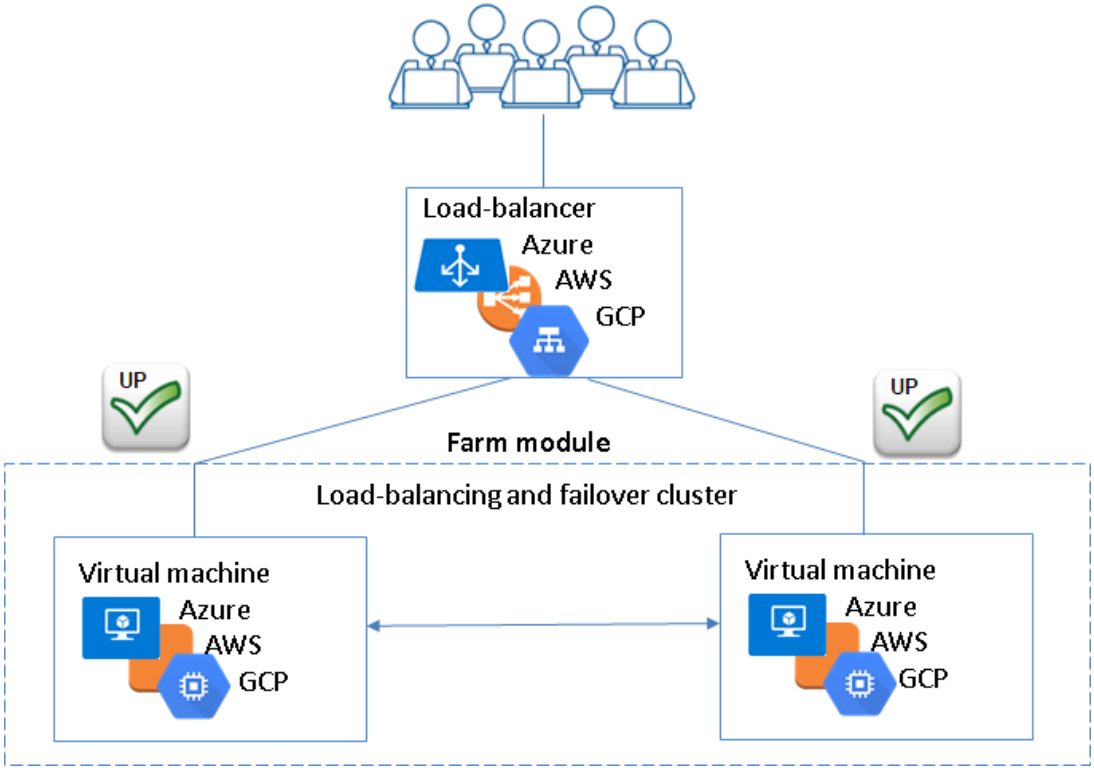
- high availability cluster with real-time replication and failover (mirror cluster)

  For a quick start, refer to mirror cluster in Azure, mirror cluster in AWS or mirror cluster in GCP.



⇨ high availability cluster with load balancing and failover (farm cluster)

For a quick start, refer to farm cluster in Azure, farm cluster in AWS or farm cluster in GCP.



2.3.2 File replication

File replication offers 3 new configuration attributes.

<rfs	
[allocthreshold="0"]	Windows only. Size in Gb to apply the allocation policy before reintegration. When allocthreshold> 0, enable fast allocation of disk space for files to be synchronized on the secondary node. This feature avoids a timeout when the primary writes at the end of the file, when the file is very large (> 200 Gb) and not yet completely copied. The allocation is applied only: ⇨ for new files (files that do not exist on the secondary when reintegration starts) ⇨ for a full synchronization (for example, during the first reintegration or when the secondary is started with `safekit second fullsync`) ⇨ when the file size on the primary is >= allocthreshold (size in Gb) Default value: 0 (that disables the feature)
[nbremconn="1"]	Number of TCP connections between the primary and the secondary nodes. This value may be increased to improve the replication and synchronization throughput when the network has high latency (in cloud for instance). Default value: 1
[sendtimeout="30"]	For SafeKit > 7.4.0.13 Timeout, in seconds, for sending packets to the remote node. This value may be increased when replication or reintegration fails on low latency networks. Default value: 30

Moreover, the file replication component:

- ⇨ now uses only one port for communication between mirror nodes
- ⇨ data synchronization has been improved for low latency networks:
  - ✓ automatic reconnection on communication failure
  - ✓ optimisation for the first synchronization retries (with `namespacepolicy="4"` that is the default value since SafeKit 7.4.0.13)
- ⇨ has been improved and fixed for hard links managements. But some operations are still not supported
- ⇨ checks on the secondary node that files are not opened before starting the synchronization

- since SafeKit 7.4.0.31, in Windows, it now supports a new value for `roflags=0x10000` attribute; when set, the secondary is stopped if a process other than system tries to modify a replicated file

2.3.3 Process death detection

Process death detection now offers a new monitoring class, `class="pre"` for monitoring processes started/stopped into user scripts `prestart`/`poststop`.

Since SafeKit > 7.4.0.19, in Linux, process death detection can monitor a Linux system service in addition to processes.

2.3.4 SafeKit web console and web server

Since SafeKit 7.4.0.13:

- The web server configuration provides built-in configuration files for login to the web console with roles based on basic authentication (ldap or file)
- On the first start of the web console, automatic insertion of the connected server as the cluster named `cluster1`, into the cluster inventory

2.3.5 DNS name resolution

Some fixes and changes have been made for a better management of the DNS resolution of names contained into the cluster configuration.

These changes also impact the way to force a DNS resolution when a DNS entry is modified. Since SafeKit 7.4.0.58, to consider the new IP address (by SafeKit services and modules), you must re-apply the cluster configuration on all nodes with the web console or the commands:

```
safekit cluster config ; safekit -H "*" -G
```

The new name resolution is not automatically considered by the running modules. To do this, you must either:

- Stop and start the module
- Run the command `safekit update -m AM`

  This is allowed only if the module is in the states UP (farm module) or ALONE (mirror module)

2.3.6 Miscellaneous

- Since SafeKit 7.4.0.19, the extension for the application log file has changed. The file name is now `userlog.ulog` and it was `userlog.AM` (where AM is the module name)
- Since 7.4.0.20, module log display (`safekit logview -m AM`) and save (`safekit logsave -m AM`) has been changed to display/save only E(vent) messages. Use `-I` option for displaying/save also I(nformation) messages, or `-A` for displaying all messages (including debug ones)
- Since SafeKit 7.4.0.27 in Linux, modification of firewall rules management
- Since SafeKit 7.4.0.27, improvement of the section "Securing the SafeKit web console" into the SafeKit User's Guide

2.4 Major Changes between SafeKit 7.3.0 and SafeKit 7.2.0

2.4.1 Service monitoring in Windows

Since SafeKit 7.3, in Windows, the process death detection has been enhancement with Windows service monitoring. For instance, to monitor the service named `myservice` edit the module configuration as follow:

```
<errd>
<proc name="myservice" service="yes" atleast="1" action="restart" class="prim" />
</errd>
```

2.4.2 External synchronization for replicated directories

On the first synchronization, all replicated files are fully copied from the primary node to the secondary node. During the following synchronizations, necessary when the secondary node comes back, only zones modified, during the secondary downtime, of files that have been modified on the primary node during the secondary node downtime. When the replicated directories are voluminous, the first synchronization can take a lot of time especially if the network is slow. For this reason, since SafeKit> 7.3.0.11, SafeKit provides a new feature to synchronize a large amount of data that must be used in conjunction with a backup tool.

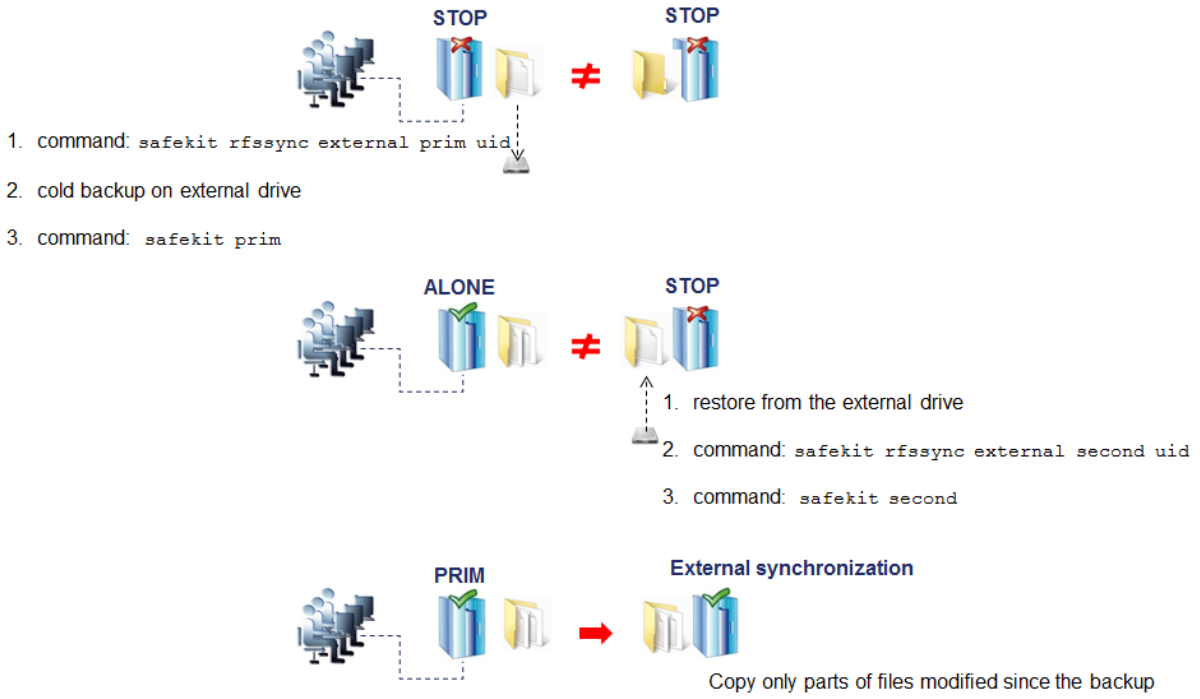
On the primary node, simply back up the replicated directories and pass the synchronization policy to the external mode. The backup is transported (using an external drive for instance) and restored to the secondary node, which is also configured to perform external synchronization. When the module is started on the secondary node, it copies only the file areas that were modified on the primary node since the backup

The external synchronization relies on a new SafeKit command `safekit rfssync` that must be applied on both nodes to set the synchronization policy to `external`. This command requires as arguments:

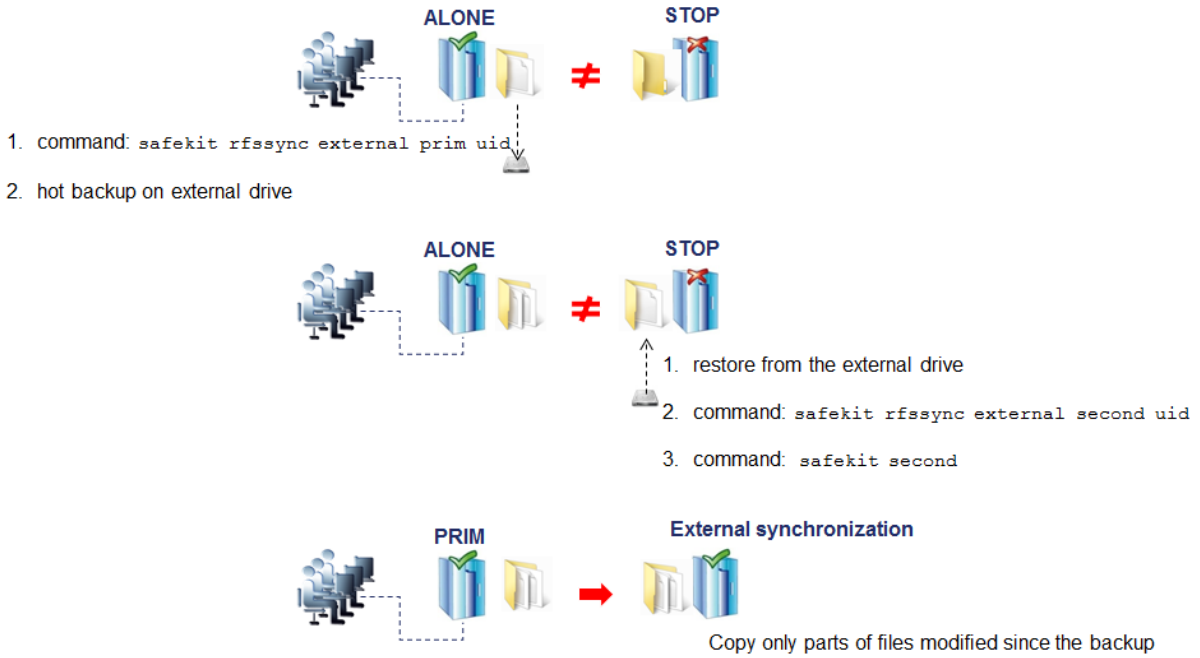
- the role of the node (`prim | second`)
- a unique identifier (`uid`)

For more details on the external synchronization refer to "File Replication (<rfs>, <replicated> tags" section in *SafeKit User's Guide.*

The external synchronization procedure, described below, is the procedure to be followed in the case of a cold backup of the replicated directories. In this case, the application must be stopped, and any modification of the replicated directories is prohibited until the module and the application are started, in ALONE - green. The order of operations must be strictly adhered to.



The external synchronization procedure, described below, is the procedure to be followed in the case of a hot backup of replicated directories. In this case, the module is ALONE - green; the application is started and changes to the contents of the replicated directories are allowed. The order of operations must be strictly adhered to.



2.4.3 File replication

2.4.3.1 File replication internals enhancement in Linux

Since SafeKit 7.3, the file replication architecture has changed and has been aligned with Windows. This will permit the implementation of 3 nodes replication on Linux in a future release of SafeKit.

In SafeKit 7.3, the NFS server is no longer used and all file operations are directly implemented by the `nfsbox` process.

2.4.3.2 Scheduled synchronization since 7.3.0.22

By default, SafeKit provides real-time file replication and automatic synchronization. On heavy loaded server or high latency network, you may want to let the secondary node weakly synchronized. For this, since SafeKit 7.3.0.22, you can use the `syncat` attribute for scheduling replicated directories synchronization on the secondary node. The module must be started for enabling this feature. Once synchronized, the module blocks in the WAIT (red) state until the next synchronization schedule.

With `syncat`, you just must configure the scheduled time for the synchronization with the syntax of the native job scheduler: `crontab` in Linux and `schtasks.exe` in Windows. For more details, refer to 13.6.4.10 <rfs> Scheduled synchronization in the section in the *SafeKit User's Guide.*

2.4.4 3 Nodes Replication Module in Linux since 7.3.0.14

Since SafeKit 7.3.0.14, the 3 nodes replication module template, `3nodesrepli.safe`, is available on Linux. It is the same feature as the one delivered in Windows since SafeKit 7.2.0 (see 2.5.4 page 26).

The 3-nodes replication feature is an extension of a mirror application module with the replication of data on a third server supposed to be a Disaster Recovery site (DR Site):

- the mirror application module between server1 and server2 works as a standard mirror module
- another module (named the spare module) makes the synchronous real-time replication, of the mirror application module data, to the DR site
- when server1 and server2 fail, the application failover to the DR site is possible with a manual procedure. And when server1 and server2 come back, the application failback from the DR site to server1 and server2 is also possible with a manual procedure.

For a full description of 3 Nodes replication, see 3 Nodes Cluster with Replication and Failover or the 3-Nodes Replication section in *SafeKit User's Guide*

2.4.5 Safekit commands

2.4.5.1 safekit prim and second for mirror modules

Before SafeKit 7.3, the `safekit prim -m AM` and `safekit second -m AM` commands used to force the start of the module `AM` as primary (`safekit prim`) or secondary (`safekit second`). Moreover, when synchronizing the replicated directories, the secondary node used to force the full copy of all files. Since SafeKit 7.3, the default synchronization policy is applied, that is reintegration optimization is applied when possible.

Since SafeKit 7.3, use the command `safekit second -m AM fullsync` for starting the module as secondary after the full synchronization of the replicated directories.

2.4.5.2 safekit swap for mirror modules

The `safekit swap` command swaps the roles of primary and secondary nodes. Before SafeKit 7.3, this command used to swap without synchronizing the replicated directories. Since this command must be used with cautions, its semantic has been changed since SafeKit 7.3:

- `safekit swap` command swaps the roles of primary and secondary nodes but now leads to the synchronization of the replicated directories on the future secondary node
- `safekit swap nosync` command swaps the roles of primary and secondary nodes without synchronizing the replicated directories.

2.4.5.3 safekit module getports

Since SafeKit 7.3.0.14, the command `safekit module getports -m AM` has changed:

- it works only if the module AM is configured
- it lists now only ports used by the module depending on its mode (mirror or farm)

You can run the command `safekit module getports -i ID` to list the ports that could be used by the module that got the id value ID. This command can be run even if the module is not yet installed neither configured, but it will return a superset of the really used port by the module.

2.4.6 Firewall settings in Linux

In Linux, since SafeKit 7.3.0.14, during the SafeKit package installation, you can select the automatic configuration of the local firewall. It automatically inserts (or remove) the firewall rules required by the SafeKit core processes (safeadmin and safewebserver services) and the modules processes to communicate with their peers in the cluster. If you opted-out, you must configure the firewall manually or you may use the `firewalldcfg` command.

2.4.7 Application modules delivery

Since SafeKit 7.3.0.14, modules customized for a specific business application are no more delivered with the SafeKit package. These HA modules can now be downloaded from Evidian site.

2.4.8 SafeKit cluster definition change since 7.3.0.22

The SafeKit cluster defines all the servers that make up the cluster as well as the IP address (or name) of these servers on the networks used to communicate with the cluster nodes. Since SafeKit 7.3.0.22, it also allows specifying the use of networks:

- a framework network (`framework="on"`) is a network used for internal communications within the SafeKit framework. These are global cluster and module internal communications, as well as communications for executing global commands. You must define at least one framework network that includes all nodes in the cluster. It is recommended to define several framework networks to tolerate at least one network failure.
- a console network (`console="on"`) is a network on which the SafeKit web console can connect for cluster and module configuration and administration. This type of network

must include all the nodes that make up the SafeKit cluster. You can define multiple console networks according to administrative requirements and network topology.

By default, a network is a network for the console and the framework communications. The SafeKit web console and cluster.xml format have changed to allow the setting of the network type. This permits to administer SafeKit cluster nodes using different administration network.

For more details refer to the *SafeKit User's Guide.*

Since SafeKit 7.3.0.22, the cluster.xml syntax accepts two new attributes for setting the network type into the `<lan>` section: `console` and `framework`.

In SafeKit < 7.3.0.22, by default, the console could connect only on the network with the attribute `admin="on"`. Since SafeKit 7.3.0.22, this attribute is obsolete (but still supported) and replaced by `console` and `framework` attributes. The web console can connect on any network with attribute `console="on"`.

2.5 Major Changes between SafeKit 7.2.0 and SafeKit 7.1.3

2.5.1 SafeKit cluster definition

Since SafeKit 7.2.0, the Safekit cluster is defined by a set of networks and the addresses, on these networks, of a group of SafeKit servers. These servers implement one or more modules. The same server can not belong to many SafeKit clusters.

The SafeKit cluster definition allows checking the consistency of the configuration of installed modules and facilitates the dynamic reconfiguration of modules. Moreover, it provides the abstraction of the network topology by naming networks. Thus, when configuring a module, the network name can be set instead of the list of nodes on the network used by the module. See the *SafeKit User's Guide* for details.

2.5.2 SafeKit web console and web server

2.5.2.1 SafeKit cluster management

The new SafeKit web console offers the ability to define the SafeKit cluster and to configure, control and monitor, modules installed on this cluster. The web console provides also the ability to administer one or more SafeKit clusters. See *The SafeKit Web Console* section in the *SafeKit User's Guide* for details.

2.5.2.2 Monitoring on smartphone and tablet

The SafeKit web console has been improved for smartphone and tablet usage. You just must connect your web browser to the SafeKit servers as usual. Two videos demonstrate it:

- Configuration of a High-Availability cluster with a mobile phone
  https://youtu.be/S-ruad4IgLI
- Monitoring of a High-Availability cluster with a mobile phone
  https://youtu.be/yZxLKx7BIGU

You can use the SafeKit web console on:

- Android smartphone and tablet with Google Chrome and Firefox. There are some restrictions with the default Samsung browser
- Windows phone with Internet Explorer
- iPhone and iPad with Google Chrome. The web console does not work properly with the default browser, Safari.

2.5.2.3 Configuration wizard for legacy modules

Legacy modules using lua (`SAFE/modules/AM/index.lua`) for displaying a user-friendly page into the "Edit the Configuration" tab of the module configuration wizard is no more supported. The raw edition of userconfig.xml of the module is proposed instead. If you prefer, you can upgrade your installed modules as described in 4.6.1.3 page 70.

2.5.2.4 SafeKit web server

For security reasons, since 7.2.0.18 release of SafeKit, Apache server has been upgraded to 2.4.10 and the openssl library to 1.0.2 H.

You can setup the SafeKit web server for using an externally built Apache server instead of the SafeKit built-in server. This permit to easily upgrade the Apache server if required for security reasons. If you intend to use the secured web console with HTTPS, you must ensure that the mod_ssl and openssl packages are also installed. For using an external Apache server, follow the procedure described in SK-0068 for Windows and in SK-0069 for Linux.

In Linux, for SafeKit <= 7.2.0.29, by default, the SafeKit web server was set for using the installed Apache server if present instead of the Apache server delivered with SafeKit package.

2.5.3 Security management

Since SafeKit 7.2.0, security issues, concerning SafeKit internal and administration communications, has been introduced (see the *SafeKit User's Guide* for details).

2.5.3.1 Security of SafeKit internal communications

The network traffic encryption can now be applied to all SafeKit internal communications:

- the global runtime network traffic, between SafeKit cluster nodes for exchanging state on installed modules, is always encrypted.
- an application module can be configured to encrypt all network communication between nodes that implement the module. With the SafeKit web console, go to the "Cryptography" section into the "Edit the Configuration" tab of the module configuration wizard.

2.5.3.2 HTTPS quick configuration wizard

A configuration wizard has been developed to make it easier to setup HTTPS for securing communications between the SafeKit web console and the SafeKit servers. With this wizard, you can build X509 certificates and switch from HTTP to HTTPS.
At the bottom of the HTTPS quick configuration wizard, you can find some links for advanced configurations: text-mode certificate request edition to allow server certificate content customization before signature has been added; interfacing with an external PKI.

Erratum in the SafeKit User's Guide in *HTTP Basic Configuration* section:

Http configuration is enabled by default after installation.

To re-enable the HTTP protocol for the unsecure SafeKit web console after another configuration disabled it, remove the file SAFE/web/conf/ssl/httpd.webconsolessl.conf and restart the web server.

2.5.3.3 Firewall Setting in Linux

Since SafeKit > 7.0.2.29, the `firewalldcfg` command has been improved for managing Linux firewall based on iptables in addition to firewalld.

The command described in *Linux with firewalld (Red Hat >= 7.0)* in *SafeKit User's Guide* can thus also been applied in RedHat 6 and Debian 8.

2.5.4 3 Nodes replication module

On Windows (not yet on Unix), SafeKit offers the 3-nodes replication feature. The 3-nodes replication feature is an extension of a mirror application module with the replication of data on a third server supposed to be a Disaster Recovery site (DR Site):

- the mirror application module between server1 and server2 works as a standard mirror module
- another module (named the spare module) makes the synchronous real-time replication, of the mirror application module data, to the DR site
- when server1 and server2 fail, the application failover to the DR site is possible with a manual procedure. And when server1 and server2 come back, the application failback from the DR site to server1 and server2 is also possible with a manual procedure.

For a full description of 3 Nodes replication, see 3 Nodes Cluster with Replication and Failover.

In Windows 2008 R2, refer to 3.5.2 page 53.

The *3 Nodes Replication* section in *SafeKit User's Guide* describes the 3 nodes replication module for SafeKit < 7.2.0.29. Since SafeKit 7.0.2.29, 3 nodes replication has been improved and the directory dir_spare is no more required.

2.5.5 File synchronization

2.5.5.1 File synchronization by date

SafeKit 7.2 offers a new command `safekit secondforce -d date -m AM` that forces the module AM to start as secondary after copying only files modified after the specified date.

This command must be used with cautions since the synchronization will not copy files modified before the specified date. It is the administrator's responsibility to ensure that these files are consistent and up-to-date.

The date is in the format of YYYY-MM-DD[Z] or "YYYY-MM-DD hh:mm:ss[Z]" or YYYY-MM-DDThh:mm:ss[Z], where:

- YYYY-MM-DD indicates the year, month and day
- hh:mm:ss indicates the hours, minutes and seconds
- Z indicates that the time is in UTC time zone; when not set the time is in local time zone

For instance:

- `safekit secondforce -d 2016-03-01 -m AM` copy only files modified after the 1st March 2016
- `safekit secondforce -d "2016-03-01 12:00:00" -m AM` copy only files modified after the 1st March 2016 at 12h, local time zone
- `safekit secondforce -d 2016-03-01T12:00:00Z -m AM` copy only files modified after the 1st March 2016 at 12h, UTC time zone

This command may be useful in the following case:

- the module is stopped on the primary server and a backup of the replicated data is done (on a removable drive for instance)
- the module is stopped on the secondary server and the replicated data is restored from the backup. It may be the first start-up or the repair of the secondary server.
- the module is started on the primary server that becomes ALONE
- the module is started on the secondary with the command `safekit secondforce -d date -m AM` where the date is the backup date

In this case, only the files modified since the backup date will be copied (full copy), instead of the full copy of all files.

In Windows, the file modification date on the secondary server is changed when the file is copied by the synchronization process. Therefore, `safekit secondforce -d date -m AM`, where date is prior to the last reintegration on this server, has no interest.

2.5.5.2 File changes since the last synchronization

Before starting a secondary server, it may be useful to evaluate the number of files and data that have been changed on the primary server since the secondary server has stopped. This feature is provided by running the following commands on the ALONE server:

1. `safekit rfsdiff -m AM > path_for_the_xml_log`

   This command runs on-line checks of regular files content of the module AM. It scans the entire replicated tree and logs the number of files that have been modified as well as the size that need to be copied. The log is in XML format and can be saved into a file (`path_for_the_xml_log` in the example).

2. The log is an XML file that is translated into a readable text log on standard output or into a file with the command:

   On Unix

   ```
   cat path_for_the_xml_log | safekit -r flatlogdump -O [ stdout | path_for_the_txt_log ]
   ```

   On Windows

   ```
   type path_for_the_xml_log | safekit -r flatlogdump -O [ stdout | path_for_the_txt_log ]
   ```

The result of the command is only an evaluation since only regular files are scanned and some other modifications may occur until the synchronization is run by the secondary server.

This command must be used with caution on a production server since it leads to an overhead on the server (for reading trees and files with locking). On Windows, rename of files can fail during the evaluation.

2.5.5.3 Configuration changes

Since SafeKit > 7.2.0.23, the attributes for the file replication configuration have changed as described below:

<rfs	
[packetsize]	Unix only. Maximum size in bytes for NFS replication packets. It must be lower than the maximum size allowed by the NFS server of both servers. When it is set into the configuration, it is used as mount options for rsize and wsize. By default, the size is the one of the NFS server.
[reipacketsize="131072"]	Maximum size in bytes of reintegration packets. In Unix, this value must be less or equal to packetsize. Default value in Unix: value of packetsize if it is set into the configuration and is lower than 131072; else 131072 Default value in Windows: 131072
[ruzone_blocksize="131072"]	Size of a zone for the modification bitmap of a file. It must be a multiple of reipacketsize attribute. Default value: value of reipacketsize if it is set into the configuration; else 131072

In SafeKit <= 7.0.2.29, the default value is 65536 instead of 131072.

When upgrading to SafeKit 7.2, refer to 4.6.2.3 File replication configuration changespage 71 for migration instructions if your configuration contains packetsize or ruzone_blocksize settings.

2.5.6 Incompatibility of SafeMonitor with SafeKit 7.2.0 and SafeKit web console

SafeMonitor, the legacy java console for SafeKit, does not support the SafeKit 7.2.0 new features. Moreover, since 7.2, users must not use at the same time SafeMonitor and the SafeKit web console. This could lead to unpredictable behavior. Therefore, since SafeKit 7.2.0, SafeMonitor is not anymore operational by default. Although not recommended, you may keep using SafeMonitor under some conditions described in 4.6.3.3 page 72.

2.5.7 Software Error Detection

Since SafeKit > 7.0.2.29, the software error detector component has been improved in Unix for monitoring processes selected using a regular expression on the command name (`nameregex` attribute). This feature is required when the command name of the monitored process is not the binary name.

Find below the changes and add-on to the SafeKit User's Guide.

2.5.7.1 <errd> Example

Unix and Windows

```
<errd>
  <proc name="myproc" atleast="1" action="restart" class="prim"/>
</errd>
```

Unix only

```
<errd>
  <proc name="oracle" nameregex="oracle_.*" atleast="1" action="restart"
class="prim"/>
</errd>
```

2.5.7.2 <errd> Syntax

```
<errd
  [polltimer="10"]
>
  <proc name="command name and/or resource name for the monitored process"
        [nameregex=="regular expression on the command name"]
        [argregex="regular expression on process arguments, including command
name"]
        atleast="1"
        action="stopstart"|"restart"|"stop"|"executable_name"
        class="prim"|"both|"second"|"sec"|"othername"]
        [start_after="nb polling cycles"]
        [atmax="-1"]
  />
  …
</errd>
```

The <errd> tag and full subtree can be changed with a dynamic configuration.

A SafeKit resource is associated with each monitored process set into the <errd> tag. The resource name is proc.<value of the attribute name> (e.g., proc.myproc) and is up when the monitoring condition is true; else down if false.

2.5.7.3 <errd>, <proc> Attributes

<errd	
polltimer="10"	Time delay between two polls of the list of processes.
<proc	Definition of a process to monitor. Set as many proc sections as there are processes.
name="process_name"	Name of the resource associated with the process to monitor. When nameregex is not set (Unix only), name is also the command name for the process. Example: on Unix, name="vi" and on Windows name="notepad.exe".  Windows only. The name is automatically converted to lower case.
[nameregex="regular expression on command name"]	Unix only Regular expression matching the command name of the process to monitor. Optional parameter. When set, the attribute name is only used for naming the resource associated with the monitored process. Example for monitoring oracle processes nameregex="oracle_.*" name="oracle" The associated resource is proc.oracle  As regular expressions are defined inside the XML file userconfig.xml, special characters interpreted by XML like '<' or '>' cannot be used in regular expressions.

2.5.8 Farm module in Debian

On Linux for a farm with load balancing on virtual IP address, the compilation of vip kernel module is required before configuring the module. In Debian, make, gcc and linux-headers-amd64 packages are required for this compilation.

In RedHat, the vip kernel compilation is automatically done on the first configuration of a farm module.

In Debian, this compilation must be done by the administrator before configuring the farm module. For this, run the following commands:

- `cd /opt/safekit/kernel`
- `make -f Makefile.debian`

Vip kernel module compilation is required only once.

2.6 Major Changes between SafeKit 7.1.3 and SafeKit 7.1.2

2.6.1 Ergonomic SafeKit web console

The SafeKit 7.1.3 web console has been improved compared to the previous versions. Effort has been made to make it more ergonomic: the console is easier to learn and use; it has got more feedbacks to guide the user and indicate the effect of a control, it provides controls to avoid human errors... Moreover, tutorials on the web console are now available for helping to use SafeKit for the first time and also to set advanced configurations: see SafeKit console tutorials. See also the SafeKit training on the web console: Management Console – Web (English version) and Console web (French version). The web console still displays 4 tabs, but they have been renamed for a better user understanding:

- Configuration tab: module quick installation and configuration

  This tab is intended to be used for installing and configuring new modules; re-configuring and monitoring installed modules. It has been improved for helping the user to take the right decision when running an action. Moreover, the configuration wizard has been enhanced to accelerate the configuration and avoid errors.

- Control tab: module runtime administration (start/stop…)

  This tab provides a finer control and detailed status of installed modules.

- Monitoring tab: module status monitoring

  This tab offers modules monitoring for administrators.

- Advanced Configuration tab: expert module configuration and management

  This tab can be used for advanced configuration and management of modules. It has been enhanced to provide the entire configuration and control functionalities on modules.

See *SafeKit User's Guide* for using the new web console. Backward compatibility has not been fully preserved. Thus, a SafeKit 7.1.3 web console cannot fully administer a SafeKit server installed with a different version.

2.6.2 Replication and reintegration bandwidth

Since 7.1.3, the replication component monitors, on the PRIM server, the bandwidth used by replication and reintegration write requests.

Two resources (`rfs_bandwidth.replication` and `rfs_bandwidth.reintegration`) reflect the average bandwidth used by replication and reintegration respectively, expressed in kilo bytes per second (KB/s).

If the replication load is IO intensive, the reintegration phase may saturate the network link and slow down the application significatively. In such a case, the attribute `reiallowedbw` of the `<rfs>` tag into the configuration file, may be used to limit the bandwidth taken by the reintegration phase (see *SafeKit User's Guide*). Please note that limiting the reintegration bandwidth will make the reintegration phase longer.

The replication bandwidth depends on the application and system activity and is reported for monitoring only.

2.6.3 Module templates

2.6.3.1 New module template hyperv2012R2.safe

`hyperv2012R2.safe` is a module template that implements a Hyper-V R3 (Windows 2012 R2 Hyper-V role) cluster with SafeKit. It provides real-time replication of VMs data, HA (High Availability) and automatic failover. See on http://www.evidian.com/safekit, the tutorial for setting up the hyperv module.

2.6.3.2 New module template iscsimirror.safe

`iscsimirror.safe` is a module template that provides HA (High Availability) and automatic failover of filesystems stored on iSCSI disk partitions. It is based on the Linux RAID subsystem *md* and an *iscsi* link between the two servers. See the file Readme.txt delivered with the `iscsimirror.safe` for setting up the iscsimirror module.

2.6.3.3 Module templates layout

The modules templates are now installed under (where `SAFE` is the root installation path):

- `SAFE/Application_Modules/generic`

  generic modules, `mirror.safe` and `farm.safe`, for integrating a new application based on a mirror or a farm architecture for generic

- `SAFE/Application_Modules/demo`

  module templates that are customized for a business application

- `SAFE/Application_Modules/other`

  advanced module templates for advanced integration

2.6.4 Japanese language support

Since SafeKit 7.1.3.7, support of Japanese language for the Safekit commands and module logs; for the SafeKit Web console and for the Windows package installer.

2.7 Major Changes between SafeKit 7.1.2 and SafeKit 7.1.1

2.7.1 SafeKit web console and web server

The SafeKit 7.1.2 web console has been improved compared to the previous versions. See *SafeKit User's Guide* for using the new web console. Backward compatibility has not been fully preserved. Thus, a SafeKit 7.1.2 web console cannot fully administer a SafeKit server installed with a different version.

The SafeKit web server has been upgraded to apache 2.2.25 and openssl 1.0.1e. Moreover, the SafeKit web console does not rely any more on the SafeKit apache module (mod_safekit). This allows to run the SafeKit web console with any web server (if this one comes with all the html pages, JavaScript code and cgi-bin binaries used by the console). The SafeKit web server configuration has been changed to provide basic user authentication when using the SafeKit web console. See *SafeKit User's Guide* for details.

2.7.2 SafeKit logs

2.7.2.1 Module logs

The log of a module has been split into 2 circular logs: the log with only main messages (I and E messages) and the verbose log (I, E, and debug messages). By default, log commands (`logview`, `logsave`, `log`) applies on the short log (use the `-A` option for the verbose log).

2.7.2.2 Commands log

SafeKit 7.1.2 comes with a log of the safekit commands ran on the server. It allows auditing the actions performed on the server to help support for instance. The log records all the safekit commands that are run and that modify the system such as a module install and configuration, a module start/stop, the safekit web server start/stop, … The commands may be displayed using:

- ✓ the web console/ Control/Select the node/Commands Log tab (it displays safekit commands applied on the selected module and all global commands)
- ✓ the web console/ Manage/ Commands Log (it displays all the commands logged on this server)
- ✓ a command for reading the file SAFEVAR/commandlog on the server side

2.7.3 SafeKit dynamic configuration

With SafeKit 7.1.2, it's now possible to change some configuration parameters when safekit is running in ALONE (green) or WAIT (red) states. This feature is called dynamic configuration.

Only a restricted subset of parameters could be changed dynamically:

- Local and remote heartbeat addresses
- Local and remote rfs flow addresses
- User, errd and check configuration
- Some minor global service and rfs parameters (See *SafeKit User's Guide* for details).

Heartbeat and rfs flow dynamic capabilities allow to dynamically change the pairing of servers in a cluster without service interruption and so to have configuration with 2 active and multiple spare machines (see the *SafeKit User's Guide* for details).

2.7.4 Asiatic language support

Use the last 7.1.2 package, for support of Asiatic languages: Japanese and Chinese language support for the Safekit commands and module logs; for the SafeKit Web console and for the Windows package installer.

2.7.5 Miscellaneous

See *SafeKit User's Guide* for details.

- Add call of the user script `confcheck`, if one, when running the command `safekit confcheck`.
- New command `safekit waitstate -m AM STOP | ALONE | UP | PRIM | SECOND` for waiting a stable state for a module.
- New command `safekit clean [all | log | process | resource] [-m AM]` for cleaning module status for support only.
- In Windows, 64 bits SafeKit package delivers 64 bits binaries. This permits to work with a larger data sets (for instance, the number of replicated files can be increased).
- In Windows 2008 R2 and above, the `safewebserver` service virtual account is used to run the safewebserver service and to secure associated files in the SafeKit file tree. In Windows versions preceding Windows 2008 R2, the `SafekitUser` account is used as in previous SafeKit versions.
- In Windows, a new script `addStartupShutdown.cmd` adds SafeKit startup and shutdown scripts as part of the computer group policy startup/shutdown scripts. Administrators that want to automatically install the SafeKit startup and shutdown scripts may run this script instead of using the Group Policy Object Management console snap-in.
- Add Japanese language support for the Safekit commands and module logs; for the SafeKit Web console and for the Windows package installer.

2.8 Major Changes between SafeKit 7.1.1 and SafeKit 7.0.11

2.8.1 SafeKit web console

2.8.1.1 Summary

The SafeKit 7.1.1 web console has changed compared to the previous versions. See *SafeKit User's Guide* for using the new web console. Backward compatibility has not been fully preserved. Thus, a SafeKit 7.1.1 web console cannot fully administer a SafeKit server installed with a release lower than 7.1.1.

For accessing the SafeKit web console, open the URL: `http://servername:9010` where servername is the network name or Ip address in dot notation of the server you want to administer.

The legacy SafeKit java console, SafeMonitor, is still delivered with the SafeKit package but we encourage users to use the new SafeKit web console. You can mix the use of the web console and SafeMonitor. For users that want to keep using SafeMonitor, you can download it from the URL `http://servername:9000`.

The mirror.safe and farm.safe delivered since SafeKit 7.0.11 have changed, compared to previous releases, for nice deployment with the web console. Modules installed in previous releases can anyway be deployed with the web console.

2.8.1.2 Detailed description

The new SafeKit web console can manage mirror modules as well as farm modules on Windows and Linux servers. From your workstation and with a standard browser, you can deploy modules on clusters, monitor them, control their logs, and even integrate a new application module.

In the Deploy tab, you can deploy an application module on a cluster. You just give the names of the servers and some additional parameters as the virtual IP address of the cluster. Once deployed, your application module is ready to be started in a high availability mode.

In the Manage tab, you can build a new module dedicated for your application. You can edit the start and stop scripts of your application. And you can customize checkers, replication, and load balancing rules in the userconfig.xml file. A colored editor helps you in this work. And a xml editor gives you all configuration options for each SafeKit component. Once your module is working, you can package it and it will be deployed by people with no skill in cluster technology.

In the Monitor tab, you have a very simple view of application modules states. You can quickly detect a failure and you have simple buttons to start or stop a module on a cluster. This interface has been designed to be as simple as possible for people administrating critical applications. The goal is to avoid human errors. Note that you can configure a global map of several clusters and several application modules in the Monitor tab.

In the Control tab, you will find an event log per server and per module and also a log of application start and stop scripts. You will see the state of checkers integrated in the application module. And advanced control commands will be also available.

Administrator roles have been also defined and some users can view only Monitor and Control tabs, while others can view all the tabs.

The code of the Monitor and the Control are locked in the browser of the workstation (if the browser supports HTML5 with local storage and Application Cache technology; else it will not be locked). Thus, once the web console is loaded, the administration workstation is independent of the failure of the SafeKit server from which the web console has been loaded.

2.8.2 New failover mode

The process monitoring, the checkers (ip, tcp, custom) and the automatic failover on a backup server can now be deactivated (`safekit errd suspend/resume -m module`, `safekit checker on/off -m module`, `safekit failover on/off`). These commands are very useful during maintenance of the critical application. The application can be stopped for maintenance without false detection and automatic failover.

2.8.3 New feature: 3 servers in a mirror cluster

A mirror cluster is based on 2 servers. If one server fails or is isolated in a remote site that cannot be reachable, it can be very useful to re-associate the alone primary server with a new secondary server. Thus, the solution returns very quickly to a high availability mode.

For that, 3 servers must be deployed with the same module configuration and with DNS names. 2 servers will be active and 1 stopped. If one active server fails, the goal is to reassociate the alone active server with the stopped one, without stopping the application on the alone active server. The procedure is simple: reconfigure the DNS to point to the stopped server, pass the command `safekit update -m module` on the active alone server, and start the stopped server.

2.8.4 New load-balancing implementation

The previous load balancing technology was based on flooding of switches when a MAC address is unknown and on setting the Ethernet card to promiscuous. In virtualized environment like VMware, prerequisites on network configuration were necessary to implement the previous load balancing technology.

A new load balancing technology has been designed to avoid these prerequisites. This technology keeps the same simple load balancing rules configuration. And the load balancing is still made efficiently by packet filtering. But there is no more dependency on the switch flooding and the promiscuous mode. The solution can be deployed in virtualized environment as VMware without specific network configuration.

2.8.5 Mail notification on failover

The SafeKit User's Guide describes how to send an e-mail on the module start, stop or failover. It says to use the `mailsend` binary delivered with the SafeKit package. Since this binary is no longer delivered with the SafeKit 7.1 package (see SK-0038), you can:

- for Windows

  download the windows binary from the `mailsend` download area

- for Unix

  use the `mail` command instead of `mailsend`. For instance, the following line, inserted in `poststop` script of a module, notifies about the stop of the module:
  ```
  echo "Running poststop" | mail -s "Stop module $SAFEMODULE on `hostname`" admin@mydomain.com
  ```

where "Running poststop" is the mail's body and "Stop module $SAFEMODULE on `hostname`" is the mail's subject.

2.8.6 Miscellaneous

The value one_side of the attribute where in the <virtual_addr> tag is no longer supported. Use instead the one_side_alias value.

2.9 Major Changes between SafeKit 7.0.10 and SafeKit 7.0.11

2.9.1 IPv6 support

SafeKit now supports using IPv6 addresses into configuration file. But there are some restrictions described in Section 3.9.1 page 58.

2.9.2 Load-balancing rules configuration change

For load-balancing rules set in farm architecture, the configuration values for proto and filter attributes have been restricted: proto="forward" and filter="on_route" are no longer supported.

2.9.3 IP address checker

Since SafeKit 7.0.9, the package delivers a custom checker for detecting IP address conflicts. This custom checker has been replaced in 7.0.11 by an IP checker to simplify user configuration. In UNIX and Windows, it checks that the IP address is locally defined; in Windows it also detects IP conflicts. See *SafeKit User's Guide* for configuring an IP checker. The IP checker configuration is automatically generated for testing the virtual IP address when writing check="on" into the <virtual_addr> tag.

2.9.4 Split brain checker

SafeKit provides a new split-brain checker that is suited for mirror architectures. Split brain is a situation where, due to temporary failure of all network links between SafeKit nodes, and possibly due to software or human error, both nodes switched to the primary role while disconnected. This is a potentially harmful state, as it implies that the application is running on both nodes. Moreover, when file replication is enabled, modifications to the data might have been made on either node.

The split-brain checker detects the loss of all connectivity between nodes and select only one node to become the primary. See *SafeKit User's Guide* for configuring a split-brain checker.

2.9.5 SafeKit package install

In Windows, since 7.0.10 you can choose the prefix for the root installation path of SafeKit (see Section 2.10.4 SafeKit package install and upgrade page 42). SafeKit 7.0.11 handles embedded spaces in the prefix path name.

2.9.6 Security fix in Windows

In Windows, access to the SafeKit installation tree has been restricted to all users except the administrator.

2.9.7 File replication configuration in Unix for Oracle Direct NFS

Since SafeKit 7.0.11, you can configure SafeKit file mirroring with Oracle 11g Direct NFS.

You have first to configure oracle for Direct NFS while SafeKit and Oracle are stopped. For this refer to the Oracle documentation. It consists in changing the ODM library by running:

```
cd $ORACLE_HOME/lib
cp libodm11.so libodm11.so_stub
ln -s libnfsodm11.so libodm11.so
```

Then you can start Oracle and check that Direct NFS is enabled. Oracle records the use of Direct NFS in `alert.log` and in internal catalog `v$dnfs` tables. For instance, you can check the table of servers accessed using Direct NFS by running:

```
su - oracle
sqlplus
system (login)
system (password)
select * from v$dnfs_servers;
```

When Oracle is properly configured for Direct NFS, you can configure SafeKit file mirroring for enabling Oracle NFS connections with SafeKit *nfsbox* process. Edit the module configuration file *userconfig.xml* and insert into the <rfs> tag the attribute: `pmapset="on"`. This option can be applied only on one module. Then apply the new configuration and start the module. You can check that Oracle uses Direct NFS and connects to the *nfsbox* port instead of the default standard *nfsd* port 2049.

The *nfsbox* port is the *nfs_port* listed by the command `safekit module getports -m AM`. To check connections, you can read the `alert.log` and `v$dnfs` tables. You can also run the command `lsof -Pnl +M -i4` (for IPv4) or `lsof -Pnl +M -i6` (for IPv6) that lists all processes connections. You should have oracle processes that connect to *nfs_port*.

To roll back to the standard Oracle configuration, stops the module, reconfigure it with the attribute: `pmapset="on"` removed and revert Oracle configuration for Direct NFS.

2.9.8 SafeKit web console (under development)

With SafeKit 7.0.11, you can administer application modules on SafeKit servers with a JavaScript capable web browser (tested browsers are Internet Explorer 8 and 9; Firefox 12; Chrome 18). For accessing the SafeKit web console, open the URL: `http://servername:9010` where servername is the network name or Ip address in dot notation of the server you want to administer. See *SafeKit User's Guide* for using the web console.

Old SafeKit java console, SafeMonitor, is still delivered with the SafeKit package. You can download it from the URL `http://servername:9000`.

The mirror.safe and farm.safe delivered with SafeKit 7.0.11 have changed, compared to previous releases, for nice deployment with the web console. Modules installed in previous releases can anyway be deployed with the web console.

2.10 Major Changes between SafeKit 7.0.9 and SafeKit 7.0.10

2.10.1 File reintegration changes

In previous SafeKit releases, when one file has been modified on the server being into ALONE state, it is fully copied on the other server during the reintegration phase. This can be over killing when the file is big and only some parts have been modified. File reintegration has been improved in 7.0.10 to copy only modified zones of the file instead of the full content. This is called zone reintegration and is enabled with `smartreintegration="8"` in `<rfs>` tag of module configuration file `userconfig.xml`. This is the default value that can be set to `"5"` to disable zone reintegration.

The attribute `ruzone_blocksize` in `<rfs>` tag defines the size in bytes of zones. The default value is 65536 and it must be multiple of `packetsize` attribute value. In UNIX, `packetsize` is dynamically got from the NFS server. An error will be logged if it is not compatible with the default value and you will have to change `ruzone_blocksize` value in `userconfig.xml`.

To implement zone reintegration, rfs component must track all changes on the replicated files. These changes are automatically saved on SafeKit nice stop into `SAFEVAR` directory (`/var/safekit` in Unix and `c:\safekit\var` in Windows if `SystemDrive=c:`). For each replicated file it saves (1300 + n*32) bytes where n is the number of modified zones.

Zone reintegration is automatically disabled:

- on server or SafeKit crash since changes have not been saved
- on replicated directories or files changes while SafeKit is stopped since changes have not been tracked
- on SafeKit upgrade since changes are not kept (for compatibility reasons).
- on module re-configuration in Windows; on replicated directory configuration change in Unix.

In Windows, changes are tracked by the `rfs` driver. Thus, any time the `rfs` driver is stopped, zone reintegration is also disabled. This is the case:

- on `rfs` driver manual restart
- on module configuration.

In Windows, to enable zone reintegration after reboot when the module has been properly stopped, `rfs` component use the NTFS USN change journal to check that saved information on zones are still valid after reboot. When the check succeeds, zone reintegration can be applied on the file; otherwise, full reintegration must be used. To enable the use of USN change journal, set `namespacepolicy="3"` in `<rfs>` tag (default value is 1). Check based on USN change journal can fail when:

- the NTFS volume does not have a USN change journal (see `fsutil usn` command for creating USN change journal on a volume)
- the USN change journal associated to the volume has been deleted/recreated for administration reasons
- a discontinuity in the USN journal is detected

In Windows and UNIX, all replicated files are now fully copied on special module start: `safekit prim`, `safekit primforce` and `safekit second`. This policy applies for any `smartreintegration` value configuration.

Special care must be taken when restoring replicated directory content from a backup. For performance reason, it is better to restore data while SafeKit is stopped. Then, the server with the up-to-date data must be started with `safekit prim` and the other one with `safekit second`.

2.10.2 File replication configuration changes in Windows

From SafeKit 7.0.9.30, the attribute `nfsbox_options` has been added to `<rfs>`. It is used to specify the policy to apply when a reparse point of type `MOUNT_POINT` is present in the replicated directory tree. This policy applies to all replicated directories.

`MOUNT_POINT` reparse points in NTFS can represent two types of objects: an NTFS mount point (for example the `D:\directory`) or an NTFS "directory junction" (a form of "symbolic link" to another part of the file system namespace).

When `nfsbox_options="cross"`, the `MOUNT_POINT` reparse point content itself is not replicated/reintegrated. It is evaluated, and the reintegration/replication process the target content as it would do for the content of a standard directory. This is useful for instance when a replicated directory is a mount point (e.g., replicating a "drive letter" root). This is the default configuration value.

When `nfsbox_options="nocross"`, the `MOUNT_POINT` reparse point content itself is replicated/reintegrated, but not evaluated. Reintegration does not descend into the target of the reparse point. This is useful for instance when a replicated directory tree contains NTFS "junctions" that point to another part of the replicated tree (e.g., when replicating a PostgreSQL database, as PostgreSQL is known to need such objects).

2.10.3 Replicated directory on-line verification

You can check that files are identical on the primary and the secondary, to verify file reintegration and file mirroring, by using the following commands on the secondary server:

3. `safekit rfsverify -m AM > path_for_the_xml_log`

   This command runs on-line verification of replicated directory trees and regular files content of the module AM. It uses the same locking mechanisms as the one used by the file reintegration to run the verification without stopping accesses on the primary. When a difference in file content is detected, the error is logged, and the verifier stops. The verifier's log is in XML format and can be saved into a file (`path_for_the_xml_log` in the example).

4. The log is an XML file that is translated into a readable text log on standard output or into a file with the command:

   On Unix

   ```
   cat path_for_the_xml_log | safekit -r flatlogdump -O [ stdout |
   path_for_the_txt_log ]
   ```

   On Windows

   ```
   type path_for_the_xml_log | safekit -r flatlogdump -O [ stdout |
   path_for_the_txt_log ]
   ```

Since running the verifier leads to an overhead on the servers (for reading trees and files with locking), it must be used with caution on a production server. Even with locking, the verifier can detect an error while it is a transient inconsistency, in the following cases:

- in Windows because modifications are done on disk before being replicated.

- when `async="second"` because reads can bypass the asynchronous writes.

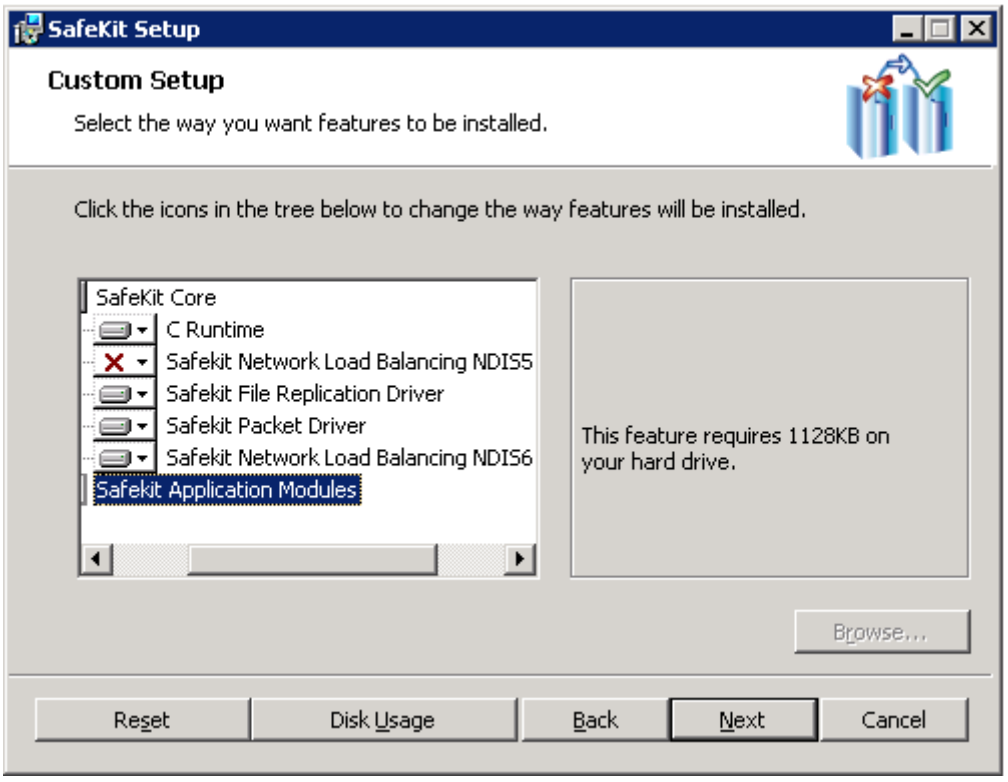
To discriminate false error detection with true inconsistency, you must run once again the verifier while there is no more access on the primary (application activity) and on the secondary (mirroring activity). If verifier logs an error again, the source and destination are different. In Windows, some files may still be different while there is no reintegration/mirroring error. This occurs on files that are modified with `SetvalidData` (it extends a file without resetting the content of the new extend; the content is then the one that is present on the disk at this time).

On-line verifier can be automatically called after file reintegration and before going into the secondary state by setting `verify="on"` in `<rfs>` tag. This option can be set for tests but is not recommended on production servers for performance reasons.

2.10.4 SafeKit package install and upgrade

From latest 7.0.9 packages (produced after May 2010), SafeKit package upgrade procedure has been improved. It mainly consists in not removing installed application modules to not have to reinstall them after the upgrade. The upgrade procedure is fully detailed in *SafeKit User's Guide*.

Since 7.0.10 package, Windows installation has been changed for providing new install options. These options are get by clicking on `"Options"` button in `"Before Starting Installation"` window. You get the following window.



You must install the `C Runtime of SafeKit Core`, but you can choose to not install (install is the default):

- the SafeKit Network Load Balancing Driver

NDIS5 IM Driver in Windows 2003 or NDIS6 IM Driver in Windows 2008 or above. It is the kernel part of `vip` component. When not installed, you will not be able to set `<virtual_interface>` that is needed for running farm modules with load balancing.

- the SafeKit File Replication Driver

  It is the kernel part of rfs component. When not installed, you will not be able to run mirror modules with file mirroring.

- SafeKit Packet Driver

  It provides the `arpreroute` feature (see "Virtual IP address takeover for real_interface") necessary in Windows 2008.

- SafeKit Application Modules

  It is the set of application modules templates delivered with SafeKit.

With the `Browse` button, you can set the prefix, a pathname that do not have embedded spaces, for root installation path for SafeKit. By default, it is `c:\safekit` (if `SystemDrive=c:`).

When a Safekit 7.0.10 version is installed on a Windows 2008 operating system (or above):

- The TCP/IP registry configuration setting (`DisableDHCPMediaSense`) is not applied anymore since this setting is now meaningless on those OS versions.
- The NTFS registry configuration setting (`NtfsDisable8dot3NameCreation = 1`) is not applied anymore since the default value (volume-dependent configuration) is more flexible. Beware that when short names (8.3 DOS compliant aliases) are used to access files, Safekit replication may encounter problems on certain patterns (involving name tunneling). Moreover, access using short names are not guaranteed to work consistently across failover. Users are strongly encouraged to disable short name creation on volumes containing Safekit replicated directories, using the `fsutil` command. For example, to disable short names on the volume whose GUID is{928842df-5a01-11de-a85c-806e6f6e6963}, type:

  ```
  fsutil 8dot3name set Volume{928842df-5a01-11de-a85c-806e6f6e6963}
  ```

2.10.5 SafeKit network load balancing driver install

In Linux, the SafeKit Network Load Balancing Driver (`vip` kernel module) is no more automatically installed with SafeKit package. It will be installed (after compiling if necessary) on module configuration if it uses `<virtual_interface>`.

In Windows, starting with 7.0.10.8, to avoid configuration problems on platforms using software vlans, the `vip` kernel module is not attached to all network interfaces at installation time. Rather, `vip` driver binding are activated on demand at configuration time, at the first reference of a given network interface.

In Windows 2003, since the `vip` driver is an NDIS5 intermediate driver that is not WHQL signed, an administrator must access the newly bound interface's property sheet (the one onto which the new virtual IP address will be added) and click OK to validate the binding, otherwise the safekit module instance referencing this binding will not start (the log will contain a line saying vipplug loading failed). This behavior is a consequence of a Microsoft design choice, it is not overridable, and does not depend on the current driver signing policy. On further references to the same network interface (possibly by other Application Module configurations), the above procedure is not needed.

In Windows 2008, the load balancing uses an NDIS6 filter technology, and the above procedure is not needed.

2.10.6 Virtual IP address takeover for `real_interface` in Windows 2008

Windows 2008 host do not perform anymore gratuitous ARP during IP address setting. This means that ARP tables of switches, routers and other hosts are not refreshed, in mirror architecture with `real_interface`, on module failover when the virtual IP address moves on the standby server. Thus, clients are not rerouted to the new primary server. To refresh ARP tables, you must set `arpreroute="on"` as shown below:

```
<vip>
<interface_list>
<interface check="off" arpreroute="on">
<real_interface>
<virtual_addr addr="VIRTUAL_TO_BE_DEFINED" where="one_side_alias"/>
</real_interface>
</interface>
</interface_list>
</vip>
```

2.10.7 Miscellaneous

- SafeKit log messages

  SafeKit log messages have been changed to ease log analysis by the end user. The command safekit logview displays now only messages for the end user (with levels I and E). To display all the messages from the log, use the command:

  `safekit logview -A`.

- SafeKit license

  SafeKit 7.0.10 can now start with no license key, but it will stop every 3 hours.

- SafeKit web server for IPv6

  SafeKit Apache web server is now configured to support IPv6.

- New module template: `vhd.safe`

  `vhd.safe` is a module template that implements a mirror cluster (primary backup), with real-time file replication and automatic failover of Windows 2008 Virtual Hard Disk (VHD).

- Application modules templates layout

  Since 7.0.10 SafeKit, application modules templates are now installed under:

- `SAFE/Application_Modules/generic`

  for generic modules `mirror.safe` and `farm.safe`

- `SAFE/Application_Modules/demo`

  for all other templates

  (where `SAFE` is the root installation path).

2.11 Major Changes between SafeKit 7.0.8 and SafeKit 7.0.9

2.11.1 File reintegration and replication changes

File reintegration has been improved to not copy twice file blocks when file modification and synchronization occur at the same time. This optimization, suited for big and log files, is enabled with `smartreintegration="5"` in `<rfs>` tag of module configuration file `userconfig.xml`. This is the default value that can be set to `"4"` to revert to the reintegration implementation of the 7.0.8.

In Windows, the rfs filter has been modified for better reliability and for logging transactional opens provided by Windows 2008. This allows to detect file system transactions that are not supported by SafeKit replicated file system.

2.11.2 Degraded mode for mirror architecture with file replication

The implementation of degraded mode, introduced in SafeKit 7.0.8.25, has been enhanced thanks to the new daemon nfsadmin.

When nfsbox, the main `rfs` component, encounters a severe error, it goes into degraded mode on the primary server instead of stopping. This mode is also enabled on abnormal exit of nfsbox.

The secondary server, if one, then runs a stopstart and blocks until the other server comes back into default mode. This improves operational continuity since there is no restart or failover of the application. But in degraded mode, file mirroring and high availability is no more provided. The alone degraded server must be restarted as primary to come back into default mode. This is a manual operation that must be ran by the administrator (`stop-prim` or `stopstart` via SafeMonitor or `safekit` command) when it knows that stopping the application is not critical. The other server will then run file reintegration and become secondary.

You can read server state to get its mode (state via SafeMonitor or `safekit` command). For instance, the following shows the state of a server in degraded mode (ALONE state and up value for resource `rfs.degraded`):

```
--------------------- mirror State ---------------------
Local (127.0.0.1) : ALONE (Service : Available)
Resources
Name State Since
heartbeat.0 up 2009-07-23 08:22:32
heartbeat.flow up 2009-07-23 08:22:32
rfs.uptodate up 2009-07-23 08:22:37
rfs.lastprimstate down 2009-07-23 08:22:37
rfs.swapping down 2009-07-23 08:22:32
rfs.degraded up 2009-07-23
```

2.11.3 Extension of supported platforms

Windows 2008 is a new supported platform.

2.11.4 Virtual Ip conflict detection in Windows

SafeKit 7.0.9 delivers a custom checker for detecting IP address conflicts useful in case of return from network isolation conditions in the context of `one_side_alias` virtual IP configurations. For instance, to check for absence of conflicts on virtual address `192.168.208.125` and run a `stopstart` if there is one, add a custom checker and a failover rule as follows into the module configuration file `userconfig.xml`:

```
<check>
  <custom ident="noipconflict" when="pre" exec="%SAFEBIN%\ipconflictcheck.exe" arg="-D -A
wait 192.168.208.125" />
</check>
<failover>
  <![CDATA[
  ipconflict:
      if (custom.noipconflict == down) then stopstart();
  ]]>
</failover>
```

2.11.5 SafeKit web server

The SafeKit web server has been upgraded to Apache 2.2.11 and now provides a SafeKit apache module for replying to SafeMonitor requests. This saves CPU cycles when monitoring many safekit modules on low end servers.

2.11.6 SafeMonitor messages internationalization

From 7.0.9.26, SafeMonitor messages are located into a separate catalog messages file to be able to change messages according to the language. SafeMonitor is delivered with English messages. The following describes how to add a French catalog messages file:

1. Unzip `safemonitor.jar`
2. Copy `MessageCatalog.txt` (English) in `MessageCatalog_fr.txt`
3. Edit `MessageCatalog_fr.txt`

   It is a set of lines with key = value. Translate values in French.
4. Save in UTF-8 format
5. Install a Java SDK with `native2ascii` command in the `bin` directory
6. Run

   ```
   native2ascii -encoding UTF-8 MessageCatalog_fr.txt MessageCatalog_fr.properties
   ```

   `MessageCatalog_fr`.properties contains Unicode-encoded (\udddd notation) characters required by java (java.util / ResourceBundle class).
7. Update `safemonitor.jar` with `jar` command in `bin` directory of Java SDK

   ```
   jar uf0 safemonitor.jar resource/MessageCatalog_fr.properties
   jar uf0 safemonitor.jar resource/MessageCatalog_fr.txt
   ```

The keywords Start, Stop, Restart, Swap, Prim, Second, ForceStop, StopStart must be translated in a single word (without blank) else the enable/disable logic of these buttons will not work.

When running safemonitor.jar with no parameter, the catalog is chosen depending on your locale language and country. Either a

- MessageCatalog_lang[_country].properties matches and is used ;
- or default MessageCatalog.properties is used

For more information, see ResourceBundle in java.util of java documentation.

You can select the MessageCatalog_fr.properties by running:

```
java -jar safemonitor.jar -lang fr
```

You can also build a catalog with the country (ex.: MessageCatalog_fr_FR.properties) and select it with:

```
java -jar safemonitor.jar -lang fr -country FR
```

2.11.7 New module template: drdb.safe

From 7.0.9.20 release, SafeKit delivers the application module drdb.safe that provides file mirroring with DRDB in Linux. It can be interesting to use DRDB for file mirroring instead of rfs to remove performance problems due to NFS and special kernel dependency. But this implies some prerequisites on disk organization and is more complex to configure than rfs component.

2.12 Major Changes between SafeKit 7.0.4 and SafeKit 7.0.8

2.12.1 File replication enhancement

File replication has been improved for better scalability and performances. File replication now supports 1,000,000 files and 500 GB of data.

The new framework is based on TCP for reliable communications; it offers asynchronous IO on the secondary server for boosting write operations and provides a new algorithm for reducing resynchronization time on the backup server after failure.

2.12.2 Extension of supported platforms

SafeKit 7.0.8 now supports Suse SLES 10 and Red Hat AS 5.1, but with some restrictions when using file replication: see Section 3, "Restrictions and Known Problems" for more details.

2.12.3 New module template: virtualserver.safe

virtualserver.safe is a module template that implements a mirror cluster (primary backup), with real-time file replication and automatic failover of virtual servers.

The full virtual server status is replicated. It consists in the Windows operating system and its configuration, applications with their license, and all the disk content.

- In case of recoverable failure, the primary server dumps the virtual server memory into replicated files. The backup server can then quickly restart the virtual server from its latest status.

- In case of unrecoverable failure, the failover procedure on the backup server consists in rebooting the virtual server.

The required configuration for running SafeKit with Virtual Server is:

- 2 physical servers with Windows 2003.
- Microsoft Virtual Server 2005 Enterprise Edition R2 installed on both physical servers (free).
- SafeKit package and `virtualserver.safe` installed on both physical servers.

2.13 Major Changes between SafeKit 7.0.1 and SafeKit 7.0.4

- The virtual IP address component (`<vip>`) has been enhanced:
  - ✓ It allows the setting of one multicast Ethernet address as the virtual MAC address associated with the virtual IP address(es) ;
  - ✓ It removes the restriction on virtual IP address configuration of SafeKit 7.0.1; the user can configure two modules with one virtual IP address on the same physical network interface. Each virtual IP address will be mapped onto its own virtual MAC address.
- The virtual hostname component (`<vhost>`) is available on Windows.
- The file replication component (`<rfs>`) offers a new configuration option (only on UNIX) for specifying the list of non-replicated entries as a regular expression.
- SafeKit provides a new built-in checker, the SafeKit application module checker (`<module>` tag from `<check>` component). The module checker is used to test the availability of an external (local or remote) module required to execute the application correctly.

All these features are detailed in *SafeKit Configuration Guide.*

2.14 Major Changes between SafeKit 7.0.0 and SafeKit 7.0.1

- SafeKit is available in Windows and Linux.
- The file replication component (`<rfs>`) has been redesigned to remove restrictions on rename operations.
- The file replication component supports the replication of file ACLs in Windows (`<rfs acl="on">`).
- For the free trial of the product, you must first download the package and then obtain a one-month free trial key at the following URL https://www.evidian.com/safekit/requestevalkey.php

2.15 Major Changes between SafeKit 6.2 and SafeKit 7.0.0

With the new multiple application modules feature introduced in SafeKit 7.0.0, SafeKit can run several applications simultaneously on the same physical servers with independent fail-over.

Thus, sophisticated high availability architectures can be implemented.

2.15.1 Mix of farm and mirror applications on the same physical servers

Such architecture makes it possible to implement a multi-tiered architecture such as `apache.safe` (farm with load balancing and fail-over) and `mySQL.safe` (mirror with file replication and fail-over) on only two servers on which the two applications are running. Thus, load balancing, file replication and fail-over can be consistently implemented on the same physical servers.

2.15.2 Mutual takeover with 2 application servers

Each application server works as a backup for the other one. For example, the `Oracledb1.safe` module has an execution priority on the first server, and `Oracledb2.safe` has priority on the second server. When one application server fails, the two applications, Oracledb1 and Oracledb2, are active on the remaining physical server.

And after the failed server is restarted, each application runs again on its default primary server. Note that in case of failure of one server in such architecture, the remaining server must be able to support the load of both applications.

2.15.3 N-1 architecture with N active application servers and only one backup

If one of the N active application servers fails, the single backup server restarts the application that was running on the failed server. When the failed server restarts, the application returns from the backup to the active server. Unlike the mutual-takeover architecture, the backup server cannot be overloaded by the execution of several applications, if there is only one failure at a time. Note that the solution can support several application server failures at the same time, but in this case all the failed applications will be restarted on the single backup server.

2.15.4 Independent application fail-over

One advantage of the multi-module feature is the existence of independent fail-over and restart procedures per application.

With the old single-module restriction in release 6.2, three applications (Appli1, Appli2 and Appli3) had to be integrated in the same ".Safe". Start and stop scripts of the unique ".Safe" included the start and stop of all 3 applications. And if a critical Appli1 process failed, all three applications (Appli1, Appli2 and Appli3) were restarted by executing the unique stop and start scripts.

With the new multiple application modules feature, each application is integrated in a different ".Safe" module (Appli1.Safe, Appli2.Safe, Appli3.Safe). Each module contains start and stop scripts for one application. And if a critical process of Appli1 fails, only Appli1 is restarted. The other application modules stay in their current state (no fail-over for them).

2.15.5 Load balancing of applications controlled by an administrator

With the SafeKit console (a multi-platform Java application) and the multi-module feature, an administrator can easily decide what application runs on what server. For example, if Appli1, Appli2 and Appli3 are running on server1 and no application is running on server2, the administrator can switch Appli3 to server2, just by clicking the swap button in the SafeKit console. Appli3 will be stopped on server1 and started on server2.

2.15.6 Process monitoring enhancement

The `<errd>` monitoring process has been enhanced with the multi-module feature.

Instead of only monitoring the name of processes, `<errd>` can be configured on the name of the process plus its list of arguments. Also, it is possible to include regular expressions in the list of arguments.

Example: `name="oracle" argregex=".*db1.*"` will monitor an oracle instance running on the database "db1" (db1 being in the arguments of the oracle process)

3.Restrictions and Known Problems

This section lists the main restrictions and known problems with the latest SafeKit release at the time the present document was written. This list is not exhaustive and must be completed with:

- *SafeKit Knowledge Base*

  An up-to-date list of all known problems and restrictions.

- *SafeKit User's Guide*

  It gives some information about each SafeKit component (file replication, fail-over and network load balancing). See

- Evidian Knowledge Base

  It contains a set of technical articles created and validated by Evidian Support. For this, log on to https://support.evidian.com.

Most of the problems listed here are also included in *SafeKit Knowledge Base* (with the associated ID: SK-<num>). When problems are fixed, it will be reported into *Software Release Bulletin*.

3.1 Restrictions and Known Problems with SafeKit 7.5.2

Restrictions on SafeKit 7.5.1 (see 3.2 page 51) are still valid for SafeKit 7.5.2.

See the SafeKit knowledge base for an uptodate list.

3.2 Restrictions and Known Problems with SafeKit 7.5.1

Restrictions on SafeKit 7.4.0 (see 3.3 page 52) are still valid for SafeKit 7.5.1.

See the SafeKit knowledge base for an uptodate list.

3.2.1 Known Problems

- Administer all the clusters of the inventory with the web console

  Since SafeKit 7.5, this global administration of modules from all clusters is incompatible with the configuration of user authentication based on file or LDAP/AD server. This means that it is incompatible with the default configuration of the SafeKit web service. If you need this feature, change the default configuration to the unsecure one or the secured one based on HTTPS and client certificates. Refer to section "Securing the SafeKit web service" in the *SafeKit User's Guide*.

3.3 Restrictions and Known Problems with SafeKit 7.4.0

Some restrictions on SafeKit 7.3.0 (see 3.4 page 52) are still valid for SafeKit 7.4.0.

See the SafeKit knowledge base for an uptodate list.

3.3.1 Known problems

- Since SafeKit 7.4.0.16, you can set DNS names into the cluster configuration. It remains some problems in Windows in some cases described in SK-0079.
- In Windows 10 Pro, the execution policy of PowerShell scripts must be changed as described in SK-0083.
- SafeKit relies on a certificate for securing module internal communications. With SafeKit <= 7.4.0.31, the validity period for this certificate is 1 year. When this certificate expires in a mirror module with file replication, the data synchronization fails. See SK-0084 for a solution.

3.4 Restrictions and Known Problems with SafeKit 7.3.0

Restrictions on SafeKit 7.2.0 (see 3.5 page 53) may be still valid for SafeKit 7.3.0.

3.4.1 Restrictions

In the SafeKit web console, the action "Estimate the data sync" may timeout when the replicated tree contains too many entries (>= 1 million). For getting the estimation, directly run the command on the server: `safekit rfsdiff -m AM`

3.4.2 Known problems

3.4.2.1 MySQL module and SELinux

When using MySQL module on a Linux configured with SELinux, MySQL may not properly work. See SK-0071 and SK-0072 for workarounds.

3.4.2.2 Linux Network Manager

When the Linux NetworkManager is used to manage network interfaces, if the network cable is unplugged, the network interface is automatically unconfigured by the Network Manager. When the cable is plugged, SafeKit may not properly detects it and sometimes requires the module stop and start (`safekit stop -m AM ; safekit start -m AM`) or dynamic update (`safekit update -m AM`) to use this network again.

3.4.2.3 DNS names

Some bugs in the DNS name resolution leads to module internal communication failures if the cluster configuration contains DNS names. See SK-0080 for workarounds.

3.5 Restrictions and Known Problems with SafeKit 7.2.0

Some restrictions on SafeKit 7.1.3 (see 3.6 page 55) may be still valid for SafeKit 7.2.0.

3.5.1 Restrictions

- If you have configured two SafeKit clusters and installed modules, you must be careful when merging these two clusters into the same one. It is recommended to uninstall first the modules on one of the SafeKit cluster before merging. Once the merge is done, you can reinstall the modules.
- Dynamic configuration may not work properly in some cases
- The 7.2.0 SafeKit web console is an intermediate version for managing the new SafeKit features. Some options are no longer supported but may be reintroduced later. For instance, you cannot set anymore the polling timeout and interval for getting the state of the modules installed on SafeKit servers.
- For farm module, when configuring the relative weight load balancing you must use the same node names as those defined into the SafeKit cluster configuration.

3.5.2 Restrictions and known problems with 3 nodes replication

3.5.2.1 Windows PowerShell version in Windows 2008 R2

The 3 nodes replication (3nodesrepli.safe) configuration relies on PowerShell scripts that require for a correct execution the change of the execution policy and the 4.0 version. In Windows 2008 R2 you may have to (see SK-0063):

- Change the execution policy

  For this:

  - ✓ start a Windows PowerShell session
  - ✓ run Set-ExecutionPolicy RemoteSigned
  - ✓ reply yes when prompt
- Install version 4.0 of PowerShell

  First, check the PowerShell version by:

  - ✓ start a Windows PowerShell session
  - ✓ run $PSVersionTable

  If necessary, upgrade to 4.0 PowerShell by following the procedure described in How to install PowerShell 4.0.

3.5.2.2 Configuration restriction

When configuring the 3nodesrepli module, it automatically configures the associated spare module 3nodesrepli_spare. The spare module is configured for using only the cluster network named "default". If you want to use another name for the network or a second network, you must directly edit the spare module configuration and apply the new configuration. Be aware that all networks set into the SafeKit cluster configuration must include the 3 nodes.

3.5.2.3 Node roles on SafeKit upgrade

With a 3nodesrepli module you must define roles for the SafeKit nodes: 2 nodes with the main role and 1 node with the disaster recovery role. On SafeKit upgrade, these roles are not preserved (see SK-0073). To set the roles again after he SafeKit upgrade and 3nodesrepli module re-configuration, apply the following procedure:

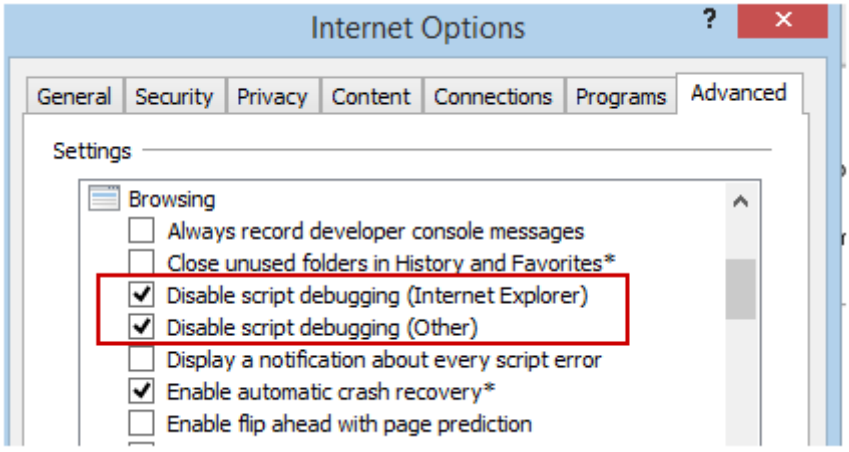
- In the SafeKit web console
- Go to Configuration tab
- Click on of the top level of the 3nodesrepli module to open the "Emergency procedures" menu
- Select "Set the disaster recovery node (DR)" entry
- Apply all the steps of the wizard to select the disaster recovery node and the main nodes

This procedure is no more required since SafeKit > 7.3.0.14.

3.5.3 Known problems with the SafeKit console

- The SafeKit web console may not properly work with some proxy configuration
- The SafeKit web console does not correctly manage literal IPv6 addresses (see SK-0062). The work around is to set names instead of literal IPv6 addresses when configuring the web console inventory and the SafeKit clusters.
- For administering SafeKit, you must not mix the use of the SafeKit web console and the legacy java console SafeMonitor. We encourage users to use the SafeKit web console.
- The browser may hang when running the SafeKit web console. In that case, kill the browser and launch a new web console. The SafeKit web console also hangs when using invalid client certificates.

- You must disable script debugging with IE 11 as shown below:



- For users still using SafeMonitor, you must not install new .safe templates and cannot take advantages of the SafeKit 7.2.0 new features. Since 7.2.0.23 release, the "Advanced Configuration" tab is not properly working and will be fixed later.

3.6 Restrictions and Known Problems with SafeKit 7.1.3

The restrictions on SafeKit 7.1.2 (see 3.7 Restrictions and Known Problems with SafeKit 7.1.2page 55) are still valid for SafeKit 7.1.3.

3.6.1 Restrictions with the web console

- You have installed and configured modules with HTTP. You now wish to secure your servers with HTTPS. For this, we recommend following the procedure described in SK-0048.
- The web console does not work properly when mixing HTTP and HTTPS connections.
- With Internet Explorer:
  - ✓ You must reload the web console after adding server(s) address(es) into trusted zone
  - ✓ TLS protocol must be enabled if you want to use the web console secured with https. Warning: IE can't display an https site when TLS 1.2 and SSL 2.0 are selected, see Microsoft KB2851628.
- When restoring a saved configuration (into last configs) with the web console, the module id is not preserved (for SafeKit <= 7.1.3.4). If you need to keep the same module id for the module, you will have to reset it manually with the command `safekit module setid` or with the web console/ Advanced Configuration/Server tab/ module/right click menu/Other Config submenu/Set module id/

3.6.2 Known problems

- The web console may not work properly when the browser is configured with a proxy server.
- Using the web console for installing a new module, there is no control that new module name is not already in use on the target nodes (for SafeKit 7.1.3).
- IE8 is no longer supported.
- With the new module templates layout (see 2.6.3.3 Module templates layout page 33), the advanced module templates cannot be anymore installed from the legacy java console SafeMonitor. We recommend using the new SafeKit web console.
- The web console may get unusable with IE11 and HTTPS. Either exit from IE and start a new IE process instance (reload does not solve the problem); use Chrome or Firefox instead of IE.
- With IE11, the animated progress bar is not displayed into the web console (see SK-0051)
- When a license is invalid, the error messages are not the one documented into the SafeKit User's Guide (for SafeKit <= 7.1.3.4).
- Error into the "Securing the console with https" section of the SafeKit User's Guide. You must connect to "http://CAserverIP:9001/conf/ca/certs" to import certificates.

3.7 Restrictions and Known Problems with SafeKit 7.1.2

The restrictions on SafeKit 7.1.1 (see 3.8 page 57) are still valid for SafeKit 7.1.2.

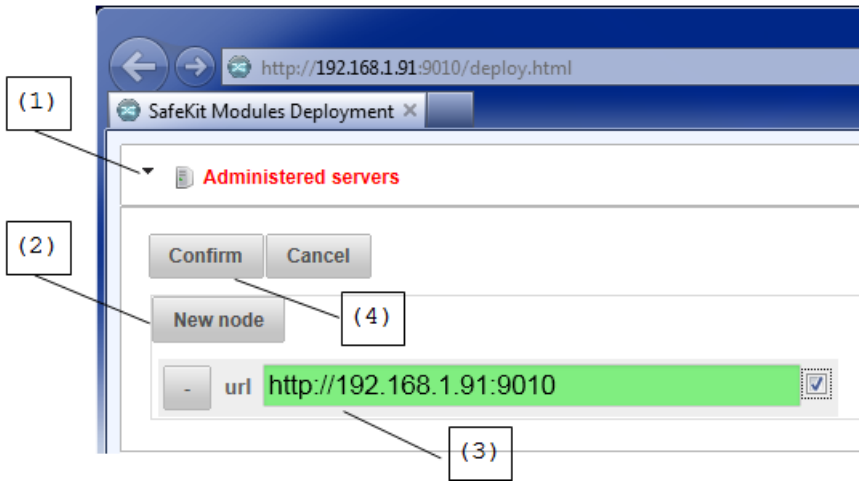
3.7.1 Using the new Web console after SafeKit package upgrade

3.7.1.1 The console does not properly work

You must clear your browser's cache to get the new web console pages (see SK-0046). A quick way to do this is a keyboard shortcut that works on IE, Firefox, and Chrome. Open the browser to any web page and hold CTRL and SHIFT while tapping the DELETE key. The dialog box will open to clear the browser. Set it to clear everything and click Clear now or Delete at the bottom. Close the browser and re-open it fresh to test what wasn't working for you previously.

3.7.1.2 Installed modules are not displayed

From SafeKit 7.1.2, only modules installed on administered servers are displayed into the console. When deploying a new module, the nodes defined as administered network nodes, are automatically inserted into the administered servers list. But when you upgrade to 7.1.2, this list is empty, and no modules are displayed. For displaying installed module(s) on upgraded SafeKit servers, you have first to add the server's URL into the administered servers list. For this:



- (1) Click on the icon to open the panel for managing administered servers list
- (2) Click on `New node` button to add a new server
- (3) Fill in the administration IP address of the server and then press the Tab key to automatically insert the default protocol and port. You can also specify the full URL if you prefer.

Do not use localhost or 127.0.0.1 as administration IP address.

- Repeat (2) and (3) when necessary
- Click on `Confirm` button to save the new administered servers list

At this step, the page should be refreshed to display all the modules installed on each administered server that you have filled.

3.7.2 IE8 restriction

The new web console does not properly work with Internet Explorer 8. Prefer IE 9, IE 10, or IE 11 instead.

3.7.3 Known problems

- Dynamic Configuration

  The <errd> tag and full subtree can be changed with a dynamic configuration. But, when the class set for the <proc> is defined by the user (i.e., different from prim, both, second, sec), the process monitoring of these processes does not apply anymore after the dynamic configuration.

- SafeKit SNMP agent

  SafeKit SNMP agent (safesnmpagent service) does not work for SafeKit releases before 7.1.2.15.

- In windows, process monitoring fails when the process name contains uppercase letters (see SK-0050).

  The SafeKit User's Guide recommends using the command safekit -r errdpoll_running to get the name of running processes. The displayed name can be used to configure the process monitoring into the <errd> section of the configuration file *userconfig.xml*. Since SafeKit 7.1.2.18, the displayed name is case sensitive while it should be in lower case. The reason is that the process name comparison for the process monitoring is not case sensitive. Thus, when defining a process monitoring into the <errd> section of *userconfig.xml*, the value of the attribute name for <proc> must in lower case. If not, the process name matching will fail.

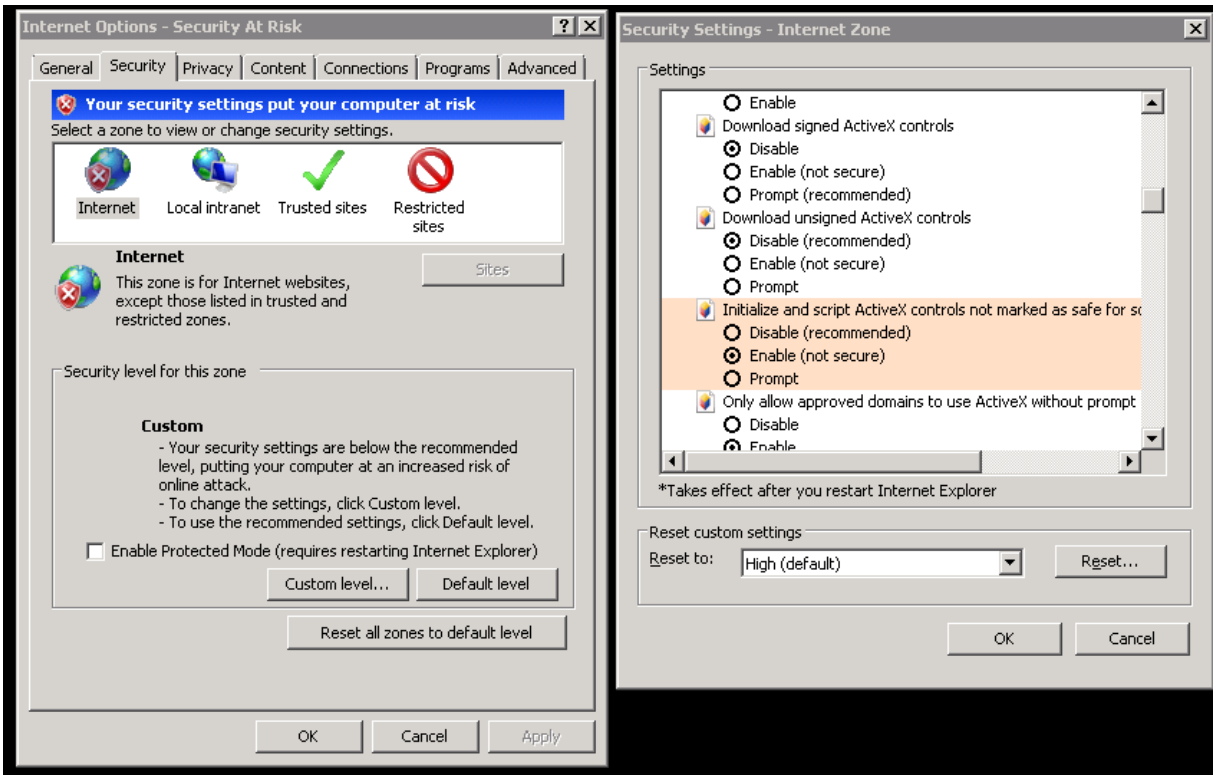
3.8 Restrictions and Known Problems with SafeKit 7.1.1

The restrictions on SafeKit 7.0.11 (see Section 3.9 page 58) are still valid for SafeKit 7.1.1, except for the restrictions on farm architectures with IPv6.

3.8.1 Web console and IE 8

The SafeKit web console may not be correctly displayed in IE 8 and returns "xml parse" errors (see also SK-0041).

The workaround consists in setting the checkbox "Enable (not secure)" for "Initialize and script ActiveX controls not marked as safe for scripting (not secure)" into the "Security Settings" panel of the Internet Zone.



3.8.2 Farm architectures and IPv6 addresses

SafeKit 7.1.1 comes with a new protocol for synchronizing nodes into a load-balanced group. This protocol replaces the spread communication protocol and supports IPv6 addresses. Then, you can configure the farm topology, into the `<farm>` tag, with Ipv6 or Ipv4 addresses.

3.8.3 Configuring 3 servers in a mirror cluster

SafeKit 7.1 comes with a new feature that permits to use a third machine as spare (see 2.8.3 page 37). In the SafeKit User's Guide such configuration is described into the section Using a third machine as spare. Since the procedure described is not yet fully supported, you must apply instead the procedure described into the SafeKit KB SK-0044.

3.9 Restrictions and Known Problems with SafeKit 7.0.11

The restrictions on SafeKit 7.0.10 (see Section 3.10 page 59) are still valid for SafeKit 7.0.11.

3.9.1 IPv6 support

SafeKit now supports configuration with Ipv6 addresses with the following restrictions.

3.9.1.1 Restrictions for mirror architectures

In mirror architecture, when defining a network, such as heartbeat network into `<heart>` tag and replication flow into `<rfs>` tag, the 1st server address and the 2nd server address must both be IPv4 or IPv6.

At the time the document was written, all heartbeat networks must be either IPv6 or IPv4. It will be fixed and notified into SafeKit Software Release Bulletin.

3.9.1.2 Restrictions for farm architectures

When configuring a load-balanced virtual ip address with the option `type="vmac_invisible | vmac_visible"` on the virtual interface, the `spread` protocol is used to synchronize the nodes in the load-balancing group. Since this protocol does not support IPv6, you must configure the farm topology, into the `<farm>` tag, with Ipv4 addresses only. The virtual IP addresses can nonetheless be IPv6 addresses.

3.9.2 HTTPS

- When using Internet Explorer 9 and HTTPS for running the SafeKit web console, you may get an error when downloading files from the SafeKit web servers. To fix this issue refer to http://support.microsoft.com/kb/2549423.
- In Manage tab/ Remote server, the operation Save to Application_Modules is not permitted when the server is configured for HTTPS.

3.9.3 Web console and IE 9

Latest KB2761451 Cumulative Update for I.E. 9 causes I.E. to stop responding.

3.9.4 Web console upgrade

Since the SafeKit web console is still under development, backward compatibility may not be preserved when upgrading SafeKit. Thus, for administering a SafeKit server you must connect the web console to this server, and you may not be able to administer another SafeKit server with a different version.

3.9.5 Permission denied in Windows

Since SafeKit 7.0.11, the `SafekitUser` account is used to run the safewebserver service and to secure associated files in the SafeKit file tree. If you encounter permission denied problems after upgrade to SafeKit 7.0.11, check that SafeKit file tree permissions are correctly set for the `SafeKitUser`.

3.10 Restrictions and Known Problems with SafeKit 7.0.10

The restrictions on SafeKit 7.0.9 (see Section 3.11 page 61) are still valid for SafeKit 7.0.10.

3.10.1 File Replication in Windows 2008 SP2, Windows 2008 R2, Windows 7

File replication errors may occur when an application extends a file (most notably, in writethrough mode): this problem is due to misbehavior of the Microsoft NTFS.sys filesystem driver described on the Microsoft support site at http://support.microsoft.com/kb/976538/en-us/. It is mandatory to update Windows (2008 SP2, 2008 R2, 7) at least to the level indicated in the above Microsoft knowledge base entry, otherwise files may not be correctly replicated.

3.10.2 Red Hat 6

If NetworkManager is used to manage network interfaces, SafeKit can't work properly in case of network failure when configuring a <virtual_address>. To configure network

interfaces, you must stop the NetworkManager and use system-config-network instead. For this, on the SafeKit server run:

1. service NetworkManager stop
2. chkconfig NetworkManager off
3. chkconfig network on
4. service network start

And then, run system-config-system to manage your network interfaces.

See also SK-0034.

3.10.3 Virtual IP

When configuring `<virtual_interface>` in Windows 2003, you have to follow the instructions described in 2.10.5 page 43. Otherwise, the module start will fail (see also SK-0032).

In Windows 2008, when using `real_interface` in `<vip>` configuration, you must set `arpreroute="on"` as shown below, for rerouting the virtual IP address on failover:

```
<vip>
<interface_list>
<interface check="off" arpreroute="on">
<real_interface>
<virtual_addr addr="VIRTUAL_TO_BE_DEFINED" where="one_side_alias"/>
</real_interface>
</interface>
</interface_list>
</vip>
```

3.10.4 Boot start of modules

Modules configured to start at boot (with `safekit boot -m AM on`) are automatically started on boot by the SafeKit administration service (`safeadmin`). On some Windows platforms, the module boot start fails because the network configuration is not ready. Since a fix is not yet available, you can apply one of the two workarounds below.

First workaround

It consists in setting manual start for the safeadmin service and applying the following procedure (it must be applied on all SafeKit servers):

- Start the MMC console with the `mmc` command line
- File - Add/Remove Snap-in Add - "Group Policy Object Editor" – OK
- Then, under "Console Root"/"Local Computer Policy"/"Computer Configuration"/"Windows Settings"/"Scripts (Start-up/Shutdown)", double click on " Start-up "
- Click on Add then set for "Script Name:"

  `<path for safekitbootstart.cmd>`

The script `safekitbootstart.cmd` contains:

```
@echo off
rem change installation path if not installed on C:\safekit
c:\safekit\private\bin\sleep.exe 90
net start safeadmin
```

Second workaround

Alternatively, you may set the safeadmin start mode to "Automatic, delayed". As administrator :

- Start the "services" control panel applet
- Select the "safeadmin" service, right-click on it to bring up the contextual menu, and select "properties"
- In the "properties" panel, change the "startup type" value to "Automatic (delayed start)"
- Close the "properties" panel

3.10.5 Zone reintegration after Windows server reboot

In Windows, to enable zone reintegration after reboot when the module has been properly stopped, you must:

- set `namespacepolicy="3"` in `<rfs>` tag of module configuration file `userconfig.xml`.
- Activate USN change journal on the volume containing the replicated directories (see `fsutil usn` command for creating USN change journal on a volume).

Even with this configuration, full reintegration is used instead of zone reintegration when:

- the USN change journal associated to the volume has been deleted/recreated for administration reasons
- a discontinuity in the USN journal is detected

3.11 Restrictions and Known Problems with SafeKit 7.0.9

The restrictions on SafeKit 7.0.8 (see Section 3.12 page 62) are still valid for SafeKit 7.0.9. But there is now a work-around for replicating a file system root in Linux (see SK-0030).

3.11.1 File Replication in Windows 2008 SP2, Windows 2008 R2

File system transactions provided with Windows 2008 are not supported.

SafeKit 7.0.9.30 brings a correction to file replication errors that may occur when an application extends a file (most notably, in `write_through` mode). This problem is a part due to misbehavior of the Microsoft NTFS file system driver and is described on the Microsoft support site at http://support.microsoft.com/kb/976538/en-us/.

To avoid file corruption, it is mandatory to update the windows operating system at least at the level indicated in the above Microsoft Knowledge Base entry, and to upgrade SafeKit at least at the level 7.0.9.30.

3.11.2 SuSe SLES 11

In SLES 11, modules in farm mode are unable to start because `vip` kernel module is not allowed to load. See SK-0029 for solving this problem.

In SLES 11 SP1, `vip` kernel module is not supported. It means that you cannot yet implement a farm module solution on this operating system. It will be notified into SafeKit Software Release Bulletin when it will be supported.

3.12 Restrictions and Known Problems with SafeKit 7.0.8

The restrictions on SafeKit 7.0.4 (see Section 3.13 page 62) are still valid for SafeKit 7.0.8.

3.12.1 File Replication and Red Hat 5

The Red Hat 5 kernel freezes with file replication on heavy write load (see SK-0018). In that case, the system hangs but the other server from the cluster does not detect the error since network communication is still working. You then must reboot the broken server.

The kernel freeze is under investigation with Red Hat. In the meantime, you can try to the change kernel parameters as follows:

1. Insert into the file `/etc/sysctl.conf`:

   ```
   vm.dirty_ratio=10
   vm.dirty_background_ratio=2
   ```

2. Run `sysctl -p`

Our tests show that these settings help to solve the problem in some cases.

Linux kernel patches fix this problem, but they are not yet integrated into Red Hat distribution. To allow the use of SafeKit file mirroring with Red Hat 5, we deliver a specific kernel that contains the appropriate patches.

3.12.2 SafeKit SNMP Agent in Windows

SafeKit SNMP agent (`safesnmpagent` service) does not work in Windows with SafeKit 7.0.8.7. This problem has been fixed in SafeKit 7.0.8.25.

3.13 Restrictions and Known Problems with SafeKit 7.0.4

3.13.1 File replication

- Windows

  File encryption and file compression are not supported on replicated files (see SK-0009).

- UNIX

  Replicated directory cannot be a file system root when `mountover="on"` (mandatory on Linux) (see SK-0010).

- Linux

  NFS server on RedHat 4 Update 3 does not support ACL. Thus, acl attribute for a replicated directory cannot be set to `"on"` (see SK-0012).

- All OS

  NFS mounts of replicated directories are not supported (see SK-0014).

  Hard links are not supported.

4. Migration Instructions

This section gives instructions and recommendations for SafeKit server migration.

4.1 Migrating from SafeKit 7.5.1 to SafeKit 7.5.2

SafeKit 7.5.2 is compatible with SafeKit 7.5.1. Therefore, the upgrade procedure is the standard one described in the *Upgrade Procedure* section of the *SafeKit User's* Guide.

4.2 Migrating from SafeKit 7.4.0 to SafeKit 7.5.1

Since SafeKit 7.5, some changes into protocol imply that this release is not compatible with older releases. Therefore, you must stop and upgrade servers at the same time when migrating to release 7.5.

Moreover, SafeKit 7.5 comes with a major change in internal data storage and web service. You can not apply the standard procedure but only the one described below.

4.2.1 Upgrade procedure

SafeKit is installed in:

	Windows if %SYSTEMDRIVE%=C:	Linux

SAFE	C:\safekit	/opt/safekit
SAFEVAR	C:\safekit\var	/var/safekit

4.2.1.1 Pre-upgrade operations

On each node to migrate:

1. Note the state "on" or "off" of services and modules started automatically at boot `safekit boot webstatus; safekit boot snmpstatus; safekit boot status -m AM` (where AM is the name of the module)
2. For a mirror module

   Note the server in the `ALONE` or `PRIM` status to know which server holds the up-to-date replicated files
3. Run the command `safekit module listid` to know the installed modules and note their ids
4. Optionally, take snapshots of modules

   Uninstalling/reinstalling will reset SafeKit logs and dumps of each module. If you want to keep this information (logs and last 3 dumps and configurations), run the command `safekit snapshot -m AM /path/snapshot_xx.zip` (replace AM by the module name)

4.2.1.2 Old SafeKit package uninstall

On each node to migrate:

1. Log as administrator/root
2. Open a PowerShell/shell console
3. Stop all modules using the command `safekit shutdown`

   For a mirror in the `PRIM-SECOND` status, stop first the `SECOND` server to avoid an unnecessary failover
4. Close all editors, file explorers, shells, or terminal under `SAFE` and `SAFEVAR` (to avoid package uninstallation error)
5. Uninstall the SafeKit package
   - in Windows, using the Control Panel-Add/Remove Programs applet
   - in Linux, using the command s`afekit uninstall`
6. undo all configurations that you have done manually for the firewall setup

Uninstalling SafeKit includes creating a backup of the installed modules in `SAFE/Application_Modules/backup`, then unconfiguring them.

4.2.1.3 Post-uninstall operations

On each node, save the cluster configuration and clean some directories:

1. Copy the file `SAFEVAR/cluster.xml` in another directory
2. Delete the directory `SAFE/modules/`
3. Delete the directory `SAFEVAR`

4.2.1.4 New SafeKit 7.5 package install

This is the quick install and setup procedure. For details, see "SafeKit install" in in the *SafeKit User's Guide*.

On each node:

1. Install the package

Windows	Linux

a. Log as administrator
b. Double click on the package
`safekitwindows_7_5_y_z.msi`
c. Open a PowerShell console
d. To setup the Windows firewall, run:
`cd SAFE\private\bin\`
`.\firewallcfg add`
e. To initialize the web service with the `admin` user and its password, for instance, `pwd`, run
`cd SAFE\private\bin\`
`.\webservercfg.ps1 -passwd pwd`
	a. Log as root
	b. Open a system console
	c. Run `chmod +x safekitlinux_7_x_y_z.bin`
	d. Run `./safekitlinux_7_x_y_z.bin`
	It extracts the package and the `safekitinstall` script
	e. Run `./safekitinstall`
	⇨ Reply `yes` for firewall automatic configuration (with `firewalld` or `iptables`)
	⇨ Reply with the password, for instance, `pwd` to initialize the web service with the `admin` user

The password must be identical on all nodes that belong to the same SafeKit cluster. Otherwise, web console and distributed commands will fail with authentication errors.

The last step is for initializing the web service that relies by default, since SafeKit 7.5, on user authentication. Once this initialization is done on all the cluster nodes:

⇨ you can authenticate in the web console with the name `admin` and the password you provided. The role is Admin by default.

⇨ you can run distributed command `safekit -H …`

Skip this initialization if you want to setup another configuration for the web service. For other setups, see section "Securing the SafeKit web service" in the *SafeKit User's Guide*.

2. If you use the web console, clear the browser cache with CTRL and SHIFT while tapping the DELETE key
3. Check with the command `safekit level` the installed SafeKit version and the validity of the license (which has not been uninstalled)

4.2.1.5 Migration operations for the web service

In the previous version of SafeKit, you may have modified the default configuration of the web service to customize it to your needs. In that special case, the customized files in `SAFE/web/conf/` have been saved in `SAFE/web/conf/<file name>.conf.<date>`.

Since SafeKit 7.5, the configuration of the web service has evolved. Carrying over your old configuration to the new version of SafeKit may require some adaptations. For details on the default setup and all predefined setups, see section "Securing the SafeKit web service" in the *SafeKit User's Guide*.

4.2.1.6 Migration operations for the cluster configuration

From one node:

1. Log as administrator/root
2. Open a PowerShell/shell console
3. Configure the cluster
   - Copy the saved `cluster.xml` file into `SAFE/var/cluster/`
   - Apply the cluster configuration on all nodes with

     ```
     safekit cluster config
     safekit -H "*" -G
     ```

   - Check the cluster configuration is consistent (same signature for all nodes)

     ```
     safekit -H "*" cluster confinfo
     ```

4.2.1.7 Migration operations for the modules

From one node:

1. Log as administrator/root
2. Open a PowerShell/shell console
3. Install the modules

   Old installed modules must be re-installed with:

   - their configuration saved into `SAFE/Application_Modules/Backup`

     Select the last saved configuration for the module (`.safe` file with the module name as prefix)

- their module id that you have noted in step 4.2.1.1. The module id is necessary if the value of the communication ports used by the module is important in your environment.

For instance, to re-intall the mirror module with the saved configuration `SAFE/Application_Modules/Backup/mirror.safe` and the id 2, run:

```
safekit module install -M 2 -m mirror SAFE/Application_Modules/Backup/mirror.safe
```

4. Configure the modules

   If the module was configured to automatically start at boot, change the module configuration to insert the attribute `boot="on"` (see 2.2.4.1 "Module boot configuration" page 13). This option replaces the command `safekit boot -m AM`.

   At this step, module can be re-configured either with the web console or commands:

   - create cryptographic for the module if necessary

     ```
     safekit module genkey -m AM
     ```

   - configure and export the module on node1 and node2 (node name in `cluster.xml`)

     ```
     safekit -H "node1,node2" -E AM
     ```

5. Start the modules

   To restart modules after the upgrade:

   - farm module

     web console/ Control/ on the module/ Start/ or command line `safekit start -m AM` (replace AM by the module name)

   - mirror module

     On the server that has the up-to-date replicated files (former `PRIM` or `ALONE`): web console/ Control/ on the node/Expert/Force start/as prim/ or command line `safekit prim -m AM` (replace AM by the module name)

     On the other server (former `SECOND`): web console/ Control/ on the node/Expert/Force start/as second/ or command line `safekit second -m AM` (replace AM by the module name)

4.2.2 Configuration of the module boot start

Before SafeKit 7.5, the configuration to start the module at boot was done with the command `safekit boot -m AM on | off` (which had to be executed on each node).

Since SafeKit 7.5, this configuration is included into the module configuration. This simplifies the configuration on both nodes and preserves the configuration on SafeKit upgrade. For details, see 2.2.4.1 "Module boot configuration" page 13.

4.3 Migrating from SafeKit 7.3.0 to SafeKit 7.4.0

Since SafeKit 7.4, some changes into the heartbeat protocol and the reintegration packet size value imply that this release is not compatible with older releases. Therefore, you must stop and upgrade servers at the same time when migrating to release 7.4.

Follow the procedure described in the *Upgrade Procedure* section in the *SafeKit User's* Guide for upgrading from SafeKit 7.3.0 to SafeKit 7.4.0.

4.4 Migrating from SafeKit 7.2.0 to SafeKit 7.3.0

Follow the procedure described in the *Upgrade Procedure* section in the *SafeKit User's* Guide for upgrading from SafeKit 7.2.0 to SafeKit 7.3.0.

4.5 Migrating from SafeKit <= 7.2.0.29 to SafeKit >= 7.2.0.32

Since SafeKit >= 7.2.0.32, the default size of blocks has been increased to improve synchronization time. It is set by the `reipacketsize` and `ruzone_blocksize` attributes in mirror modules using file replication.

The default value in SafeKit < 7.2.0.32 is `65536`; the default value for mirror modules in SafeKit >= 7.2.0.32 is `131072` (see 4.6.2.3 page 71). When the block size value is different on the two servers, the reintegration on the secondary node fails. Therefore, the upgrade procedure must be slightly changed compared to the standard one.

When migrating from SafeKit <= 7.2.0.29 to SafeKit >= 7.2.0.32, apply the standard procedure. If you stop and upgrade both servers at the same time, no change to the default procedure is required. The mirror module will then run with the new default value (`131072`) for the reintegration and zone block size.

If you stop and upgrade first the secondary server, while keeping the primary server up with the old SafeKit package, before re-configuring the mirror module edit the userconfig.xml and insert into the `<rfs>` tag: `reipacketsize ="65536"`. You will have also to set this attribute on the primary server before re-configuring the module after the upgrade. The mirror module will then run with the old default value (`65536`) for the reintegration and zone block size.

4.6 Migrating from SafeKit 7.1.3 to SafeKit 7.2.0

4.6.1 Upgrade Procedure

The *Upgrade Procedure* section in the *SafeKit User's* Guide describes the light upgrade procedure (for instance, for upgrading from 7.2.0.18 to 7.2.0.23). Since migrating from SafeKit 7.1.3 to SafeKit 7.2.0 is a major upgrade, follow the procedure below. This procedure also limits the downtime of SafeKit modules.

4.6.1.1 Upgrade the SafeKit package

Consider that modules are up (UP for farm module, PRIM-SECOND for mirror module) on server1 and server2. We upgrade first server1 that is in SECOND state for mirror module.

On server1:

1. Stop all modules with the command line `safekit shutdown`. The other server is still up for the modules.
2. Undo all configuration for firewall
3. Uninstall the SafeKit package
   - On Windows as Administrator, uninstall SafeKit using the Control Panel-Add/Remove Programs applet
   - On Unix as root, uninstall SafeKit using the SAFE/safekit uninstall command

   Uninstalling automatically creates a backup of the installed modules in SAFE/Application_Modules/Backup.
4. Install the SafeKit 7.2 package as detailed in the *Install Package* section in the *SafeKit User's* Guide but **do not start the SafeKit web console**.

   AT THIS STAGE, DO NOT USE THE WEB CONSOLE. It will not work properly since the SafeKit level is not the same on both servers.
5. Apply firewall configuration (see *Firewall Settings* section of the *SafeKit User's* Guide)
6. Reconfigure all modules with the command line `safekit config –m AM` (where AM is the name of your module)
7. Start all modules with the command line `safekit start –m AM`

At the end of this procedure, modules are up on server1 and server2 and you can apply the same procedure (1-7) on server2. Once SafeKit has been upgraded on all servers, it is safe to use the SafeKit console for administering the servers (see *The SafeKit Web console section* in the *SafeKit User's* Guide). First, you must clear your browser's cache to get the new web console pages (see SK-0046). A quick way to do this is a keyboard shortcut that works on IE, Firefox, and Chrome. Open the browser to any web page and hold CTRL and SHIFT while tapping the DELETE key. The dialog box will open to clear the browser. Set it to clear everything and click Clear now or Delete at the bottom. Close the browser, if necessary, stop processes still running in background, and re-open it fresh to load the SafeKit web console.

If you were using HTTPS configuration with SafeKit 7.1.3, refer to 4.6.1.2 page 69.

The upgrade procedure is completed but modules are still configured for SafeKit 7.1.3. Go to 4.6.1.3 page 70 for upgrading SafeKit modules if necessary.

4.6.1.2 Instructions for HTTPS configuration

You have configured the SafeKit web console and SafeKit servers for using HTTPS with SafeKit 7.1.3 package. Follow the upgrade procedure to install 7.2.0 package. Once upgraded, before using the SafeKit web console with HTTPS, you don't have to fully apply the procedure described in the section *HTTPS Quick Configuration with the Configuration Wizard* of the *SafeKit User's Guide*. Just apply the following procedure on each SafeKit servers:

1. Start the SafeKit CA web service as described in 11.1.1 and remember the password you have entered

2. Using a browser, connect to the local CA Web service. The CA web service is located at https://localhost:9001. At the login prompt, enter CA_admin as username, and the password you have just specified. Then, the HTTPS configuration wizard is displayed.
3. In the first tab, check that all the certificates are "available"
4. Go to the second tab, click on the "Restart in https mode" button as described in 11.1.3
5. Go to the third tab, click on the "Apply firewall rules" button to set firewall rules for the operating system firewall. For more details, see 11.1.4.
6. Go to the fourth tab, stop the SafeKit CA web service as described in 11.1.6

When this is done on all SafeKit servers, you can go using the SafeKit web console with HTTPS.

4.6.1.3 Upgrade the SafeKit modules

To get benefits of the new SafeKit features and administration, you must upgrade 7.1 modules to 7.2 templates. For upgrading 7.1 modules installed with a previous SafeKit release, the simplest way is to first stop the module on all servers; then apply the following procedure for each module:

1. Stop the module on server1 and server2 with the SafeKit console or the command line `safekit stop -m AM`
2. Save a copy of the module with the SafeKit web console
   - In Advanced Configuration tab
   - Node tab
   - Installed modules
   - Right-click on the module and choose Save to Application_Modules
3. Uninstall the module on server1 and server2 with the SafeKit console or the command line (see *Uninstall a module* section of the *SafeKit User's Guide*)
4. Install the module using the generic module templates, farm.safe or mirror.safe, delivered with the SafeKit 7.2 package and give the name of the previously installed module:
   - use the SafeKit web console for the quick installation and configuration of the module (see *Configure a Module with the Web Console* of the *SafeKit User's Guide*)

   or

   - if you mind the module's id, install the module with the command line and the argument `-M id` that sets the module id. For example, run `safekit module install -m AM -M 2 SAFE/Application_Modules/generic/farm.safe` for installing the AM module from the farm template with the module id 2
5. Report your configuration parameters (except obsolete configuration options) and user scripts into the new installed module (see *Advanced configuration of a module* of the *SafeKit User's Guide*)
6. Apply the new configuration on server1 and server2

At this step, the module is ready to start and use the SafeKit 7.2 features.

4.6.2 Configuration Changes

4.6.2.1 New declaration for <heartbeat> and <farm> tags

The definition of monitoring networks, into `<heartbeat>` and `<farm>` tags, in the userconfig.xml configuration file has changed. But the old format is still supported.

You may use the new format if you want to use the new module templates and if you wish to gain benefit of the network topology abstraction. For this, the best way is to use the SafeKit web console to:

- define the SafeKit cluster required for the new configuration format
- migrate installed modules as described in 4.6.1.3 page 70
- configure modules with the configuration wizard that manage the new configuration format

4.6.2.2 Obsolete declaration

Since SafeKit 7.2.0, the `<application_module>` header, in the userconfig.xml configuration file is no longer supported. The web console:

- get the list of nodes implementing the module from the SafeKit web server
- use the name set into the SafeKit cluster configuration as display name for each node

In the current implementation, you cannot set the display name for the module and the set of resources to display.

4.6.2.3 File replication configuration changes

Since SafeKit > 7.2.0.23, the attributes for the file replication configuration have changed as described below:

`<rfs`	
`[packetsize]`	Unix only. Maximum size in bytes for NFS replication packets. It must be lower than the maximum size allowed by the NFS server of both servers. When it is set into the configuration, it is used as mount options for rsize and wsize. By default, the size is the one of the NFS server.
`[reipacketsize="131072"]`	Maximum size in bytes of reintegration packets. In Unix, this value must be less or equal to `packetsize`. Default value in Unix: value of `packetsize` if it is set into the configuration and is lower than `131072`; else `131072` Default value in Windows: `131072`
`[ruzone_blocksize="131072"]`	Size of a zone for the modification bitmap of a file.

	It must be a multiple of `reipacketsize` attribute. Default value: value of `reipacketsize` if it is set into the configuration; else `131072`

When migrating to SafeKit > 7.2.0.23:

- In Unix, if your configuration contains `packetsize` and/or `ruzone_blocksize` settings, no changes are required
- In Windows, if your configuration contains `packetsize` setting, change the `packetsize` attribute name to `reipacketsize` name

The change of the default value for `ruzone_blocksize` has an impact on the upgrade procedure described in 4.5 page 68.

4.6.3 Miscellaneous

4.6.3.1 SafeKit Web Console Changes

- The file SAFE/Application_Modules/webconsole/servers.xml, used in previous release for saving the list of administered servers, is no longer used
- The server connected to the SafeKit web console must be included into the SafeKit cluster

4.6.3.2 Obsolete Command

The command `safekit module setid -m AM` is no longer supported. To force the module id:

- install the module with the command line and the argument `-M id` that sets the module id. For example, run `safekit module install -m AM -M 2 SAFE/Application_Modules/backup/farm.safe` for installing the AM module from the farm template with the module id 2

or

- insert into the in the `<service>` tag of the userconfig.xml configuration file, the attribute `id="value"` (ex: `id="5"`)

4.6.3.3 Legacy java console

Since SafeKit 7.1.1, the legacy java console, SafeMonitor, is in maintenance mode. It is still delivered with the SafeKit package but is not anymore operational by default. We encourage users to use the new SafeKit web console to get benefits of the new SafeKit features and administration.

For users that want to keep using SafeMonitor with SafeKit 7.2, you must change the SafeKit web server configuration to enable SafeMonitor access. For this, follow the procedure below after upgrading SafeKit:

1. Change the firewall rules to allow connection on port 9000.

   When using the Windows firewall, open the Windows firewall advanced settings for changing "Inbound Rules". Edit the properties of SafeKit-webserver-http and insert into "Protocols and Ports" the value 9000

2. Edit the file `httpd.conf` into `SAFE/web/conf` directory

   Remove the `#` char at the beginning of the line:

```
#Include "SAFE/web/conf/httpd.safemonitor.conf"
```

(where `SAFE` is set to the SafeKit installation path)

3. Restart the SafeKit web server with the command `safekit webserver restart`

At this step, you can download SafeMonitor from the URL `http://servername:9000` and run it.

To avoid unpredictable behavior when using SafeMonitor:

- You must not use the SafeKit web console
- You must not use the 7.2 configuration for the cluster and modules. You can only use 7.1 module templates.

Some SafeMonitor features are no more supported:

- The module id management is no more operational. You must use command line instead:

  `safekit module install -m AM -M 2 SAFE/Application_Modules/backup/farm.safe` for installing for example the AM module from the farm template with the module id 2

- The "Quick configuration" tab is no more operational for the edition of the module configuration. You must go to the "Expert Configuration" tab, for the raw edition of the userconfig.xml of the module.

Since SafeKit > 7.2.0.23, SafeMonitor version is 7.2.0.26 and the tabs corresponding to obsolete features are no more available ("Id Management" into the "Server Administration" tab, and the "Quick configuration" tab).

4.7 Migrating from SafeKit 7.1.2 to SafeKit 7.1.3

Migration instructions are the same as the ones described in 4.8 page 74.

4.7.1 Upgrade Procedure

You can limit the downtime of SafeKit modules by following the procedure below. Consider that modules are up (UP for farm module, PRIM-SECOND for mirror module) on server1 and server2. We upgrade first server1 that is in SECOND state for mirror module.

8. Stop all modules on server1 with the SafeKit console or the command `safekit shutdown`. server2 is still up for the modules.
9. Upgrade SafeKit as detailed in the *Upgrade Procedure* section of the *SafeKit User's Guide.* At this step, the console may not properly work since SafeKit level is not the same level on all servers. Therefore, prefer the command line instead of the use of the SafeKit console for configuring modules (`safekit config -m AM`)

   Be careful to reapply the firewall settings since firewall rules have changed compared to previous releases.

   At the end of this procedure, modules have been reconfigured and started. Modules are up on server1 and server2.

At this step, you can apply the same procedure (1. and 2.) on server2. Once SafeKit has been upgraded on all servers, it is safe to use the SafeKit console for administering the servers.

4.7.2 Instructions for the web console

After upgrade, you must clear your browser's cache to get the new web console pages (see SK-0046). A quick way to do this is a keyboard shortcut that works on IE, Firefox, and Chrome. Open the browser to any web page and hold CTRL and SHIFT while tapping the DELETE key. The dialog box will open to clear the browser. Set it to clear everything and click Clear now or Delete at the bottom. Close the browser and re-open it fresh to test what wasn't working for you previously.

Once the web console properly reloaded, if you still encounter problems when using the configuration wizard for modules installed in a previous version of SafeKit, you must:

1. uninstall the module
2. reinstall the module using the module templates delivered with the SafeKit 7.1.3 package
3. report your configuration parameters and user scripts into the new installed module (see 3.7.1 Advanced configuration of a module into the *SafeKit User's Guide*)

Module templates stored into `SAFE/Application_Modules/published` are no longer displayed into the web console. If you have used this directory to store templates, move the templates into `SAFE/Application_Modules/demo`.

4.8 Migrating from SafeKit 7.1.1 to SafeKit 7.1.2

Migration instructions are the same as the ones described in Section 4.9 page 75.

4.8.1 Upgrade Procedure

The upgrade procedure is fully detailed in the *Upgrade Procedure* section of the *SafeKit User's Guide.* The upgrade preserves modules that were previously installed:

- For mirror modules, 7.0.9, 7.0.10, 7.0.11, 7.1.1 and 7.1.2 protocols are compatible, so you do not have to stop all the servers before upgrade.
- For farm modules, you must stop all nodes that belong to the cluster before upgrading only if you change load-balancing configuration (`vmac_invisible` to `vmac_directed`).

For using the SafeKit web console coming with SafeKit 7.1.2, apply the procedure described in section 3.7.1 page 56.

4.8.2 Configuration Options Changes

The following configuration options, set into the userconfig.xml configuration file, have changed:

- In <errd> tag, the default value for `atleast` attribute has been changed from -1 to 1. When upgrading to 7.1.2, you then must set `atleast="-1"` if this attribute was not set.
- In <intf> checker tag, the attribute `intf`, that is the network interface name to check, is no more required. When upgrading to 7.1.2, this attribute will be ignored.

4.9 Migrating from SafeKit 7.0.11 to SafeKit 7.1.1

Migration instructions are the same as the ones described in Section 4.10 page 76.

4.9.1 Upgrade Procedure

The upgrade procedure is fully detailed in the *Upgrade Procedure* section of the *SafeKit User's Guide.* The upgrade preserves modules that were previously installed.

- For mirror modules, 7.0.9, 7.0.10, 7.0.11 and 7.1.1 protocols are compatible, so you do not have to stop all the servers before upgrade.
- For farm modules, 7.1.1 comes with a new group communication protocol. Thus, you must stop all nodes that belong to the cluster before upgrading if you change configuration (`vmac_invisible` to `vmac_directed`).

4.9.2 Farm Modules

We encourage users that publish their own application module in farm architecture, to use the new load-balancing implementation (see 2.8.4 page 37). For this, replace into the configuration file, `type="vmac_invisible"` by `type="vmac_directed"` into the `<virtual_interface>` tag.

4.9.3 Legacy java console

The SafeKit java console, SafeMonitor, is still delivered with the SafeKit package but we encourage users to use the new SafeKit web console. For users that want to keep using SafeMonitor, you can download it from the SafeKit home page at `http://servername:9000`. From SafeKit 7.1.1, SafeMonitor goes into maintenance mode.

When upgrading/fixing SafeKit servers, you must restart SafeMonitor to get from the SafeKit server the new messages catalog for the log of the module.

4.9.4 Obsolete Configuration Options

The following configuration options, set into the userconfig.xml configuration file, are no longer supported:

- `nbthread` into `<rfs>` tag
- `addr="broadcast/multicast IP address"` into `<lan>` tag
- `type="vmac_visible"` into `<virtual_interface>` tag
- `where="one_side"` into `<virtual_addr>` tag. Replace with `"one_side_alias"`.

4.9.5 Mapping a virtual IP address to a virtual MAC address in a mirror module

Some mirror modules, implemented in SafeKit < 7.1, may have been configured for mapping a virtual IP address on a virtual MAC address. With this setting, arp rerouting is not necessary after a swap. It was configured in SafeKit < 7.1 as follow:

```
<vip>
<interface_list>
<interface check="on">
<virtual_interface type="vmac_invisible">
```

```
<virtual_addr addr="192.168.1.50" where="one_side_alias"/>
</virtual_interface>
</interface>
</interface_list>
</vip>
```

With SafeKit 7.1, this configuration must be changed as described into the SafeKit KB SK-0043 (be careful, in this case the installation of the vip kernel module is mandatory).

4.10 Migrating from 7.0.10 to SafeKit 7.0.11

4.10.1 Upgrade Procedure

The upgrade procedure is fully detailed in the *Upgrade Procedure* section of the *SafeKit User's Guide.*

7.0.9, 7.0.10 and 7.0.11 protocols are compatible, so you do not have to stop all the servers before upgrade. The upgrade preserves modules that were previously installed.

4.10.2 Migrate SafeKit Web Server Configuration

The SafeKit Web Server configuration has been completely reorganized to implement the SafeKit Web Console.

The directory `<SAFE>/web/conf` contains the SafeKit web server configuration files. Either you do not perform any special configurations and the default configuration delivered with SafeKit will be applied after upgrade, or you have made a special configuration and then your files have been saved during uninstall with the date as suffix. You will then have to put back your previous changes in the new configuration files.

In SafeKit 7.0.10, the command `safekit -C d <action> <arg>` permit to execute an action on all servers specified in the file SAFEVAR/default_cluster.txt. Example: `safekit -C d module list`: list modules on all servers defined in default_cluster.txt

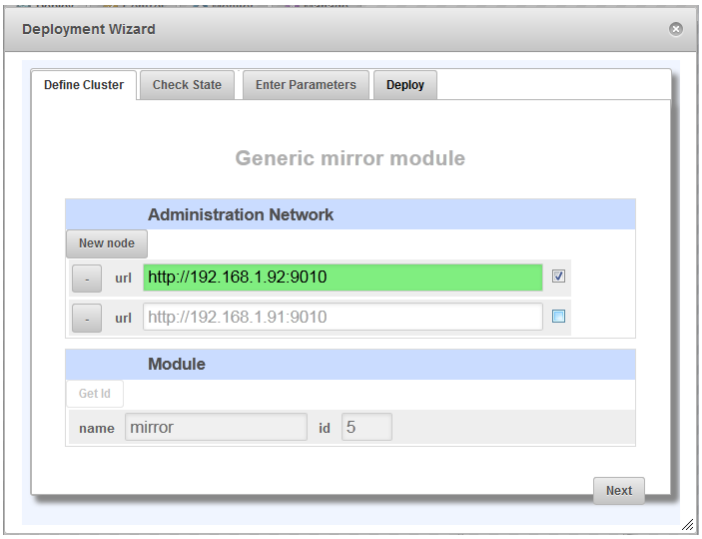
The file default_cluster.txt contains for example:
http://192.168.0.1:9000
http://192.168.0.2:9000

In SafeKit 7.0.11, you must edit this file and replace the port value 9000 with 9010.

4.10.3 Recommendation when using the web console

The SafeKit web console delivered with 7.0.11 is still under development and will change in 7.1.1 release. Anyway, if you plan to use the SafeKit web console, re-configure the modules with the Deploy tab of the console. Be careful, if you do not upgrade all the server(s) at the same time (one node runs the latest release while other server runs an older one):

- Upgrade the 1st server (e.g., 192.168.1.92) with the latest release
- Connect the web console to this server (http://192.168.1.92:9010) and start the deployment wizard. You must define all the nodes that belong to the cluster but select the node's checkbox only for the server installed with the latest release.



This permit to fully define the cluster but the module is configured only the selected node.

- For later deployment, when the other nodes have been upgraded, select the node's checkbox for all the nodes so that the configuration is applied on all of them.

4.11 Migrating from SafeKit 7.0.9 to SafeKit 7.0.10

4.11.1 Recommendation for Modules using `<rfs>`

Releases 7.0.9 of SafeKit are fixed with releases 7.0.10. To reduce the downtime of applications, we have improved the migration procedure for mirror modules using file replication (only with `mode="read_only"`). For this run the following procedure:

1. server1 and server2 are up with the SafeKit 7.0.9. server1 is primary and server2 is secondary.
2. stop the module on server2 and upgrade the server with 7.0.10. Configure the module and start it with safekit second. SafeKit 7.0.10 will detect that the primary server is running 7.0.9 and will use the basic reintegration protocol.
3. After server2 reintegration, server1 is primary with 7.0.9 and server2 is secondary with 7.0.10.

At this step, don't swap since a primary server with 7.0.10 and a secondary server with 7.0.9 will not work. The backward compatibility is provided only for the secondary server.

4. stop module on server1 and upgrade the server with 7.0.10. Configure the module and start it with safekit second. This time, the 7.0.10 reintegration protocol is used.
5. server1 and server2 are up with the SafeKit 7.0.10. server1 is secondary and server2 is primary.
6. At this step, you can use SafeKit as usual.

4.11.2 Recommendation for Modules using <vip>

In Windows 2008, when using real_interface in <vip> configuration, you must set arpreroute="on" as shown below, for rerouting the virtual IP address on failover:

```
<vip>
<interface_list>
<interface check="off" arpreroute="on">
<real_interface>
<virtual_addr addr="VIRTUAL_TO_BE_DEFINED" where="one_side_alias"/>
</real_interface>
</interface>
</interface_list>
</vip>
```

In Windows 2003, when using virtual_interface in <vip> configuration, after the module configuration and before starting the module, you must access the corresponding network interface's property sheet (the one onto which the new virtual IP address will be added) and click OK to validate the vip kernel module binding. This procedure is needed only the first time that this interface is configured as virtual_interface (by one or many modules).

4.11.3 Upgrade Procedure

From latest 7.0.9 packages (produced after May 2010), SafeKit package upgrade procedure has been improved. It mainly consists in not removing installed application modules to not have to reinstall them after the upgrade. The upgrade procedure is fully detailed in the *Upgrade Procedure* section of the *SafeKit User's Guide*. It also covers the upgrade from older releases.

4.11.4 Migrate SafeKit Web Server Configuration

The SafeKit Web Server configuration has changed to implement the SafeKit Web Console.

The directory <SAFE>/web/conf contains the SafeKit web server configuration files. Either you do not perform any special configurations and the default configuration delivered with SafeKit will be applied after upgrade, or you have made a special configuration and then your files have been saved during uninstall with the date as suffix. You will then have to put back your previous changes in the new configuration files.

4.11.5 Miscellaneous

- SafeKit log messages

  SafeKit log messages have been changed to ease log analysis by the end user. This implies that integrations that parse log messages via safekit logview/safekit logsave or SNMP traps must be changed. Moreover, the command safekit logview displays now only messages for the end user (with levels I and E).

- vipifctrl command for support

  In Windows, vipifctrl command has been renamed in vip_if_ctrl.

4.12 Migrating from SafeKit 7.0.x < 7.0.9 to SafeKit 7.0.10

Migration instructions are the same as the ones described in Section 4.13 page 79.

SafeKit Releases < 7.0.9 can be fixed with releases 7.0.10, but these releases are not compatible in terms of file replication protocol. Therefore, you must stop and upgrade servers at the same time when migrating to release 7.0.10.

4.13 Migrating from SafeKit 7.0.x to SafeKit 7.0.9

Migration instructions are the same as the ones described in Section 4.14 page 79. Releases 7.0.8 of SafeKit are fixed with releases 7.0.9, but these releases are not compatible in terms of file replication protocol. Therefore, you must stop and upgrade servers at the same time when migrating to release 7.0.9.

4.14 Migrating from SafeKit 7.0.x to SafeKit 7.0.8

4.14.1 Recommendation

Releases 7.0.x of SafeKit are fixed with releases 7.0.8, but these releases are not compatible in terms of file replication protocol. Therefore, you must stop and upgrade servers at the same time when migrating to release 7.0.8.

You may also need to modify the file replication configuration (`<rfs>`) as explained later.

4.14.2 Upgrade Procedure

The upgrade procedure is the one explained in the *Upgrading/Fixing the SafeKit Package* section of the *SafeKit User's Guide.*

4.14.3 `<rfs>` Configuration

SafeKit 7.0.8 offers improvements and performance boosts for file replication. Thus, all tunings set into the file replication configuration (`<rfs>`) for release 7.0.x are no longer required.

Moreover, some attributes are no not supported any longer since they are specific to UDP implementation that is replaced by TCP. The following obsolete attributes must be removed from the configuration file: `nbnfsd`, `nbbiod`, `maxnbretrans`, `synctime`, `nfs_vers`, `transport`.

A standard 7.0.8 `<rfs>` configuration looks like this:

```
<rfs mountover="on|off" acl="on|off"
       [packetsize="buffer size in bytes to be specified only for Oracle"]>
…
</rfs>
```

Try this default configuration and refer to the `<rfs>` section in *SafeKit Configuration Guide* if you encounter any problems.

4.15 Migrating from SafeKit 7.0.x to SafeKit 7.0.4

4.15.1 Recommendation

Releases of SafeKit < 7.0.4 are fixed with releases 7.0.4, but since these releases are not compatible in terms of file reintegration protocol, when you upgrade you must stop and

upgrade servers at the same time. This is the only constraint when migrating to release 7.0.4.

4.15.2 Upgrade Procedure

The upgrade procedure is the standard one explained in the *Upgrading/Fixing the SafeKit Package* section of the *SafeKit User's Guide.*

4.16 Migrating from SafeKit 6.2 to SafeKit 7.0.4

4.16.1 Recommendation

When migrating from SafeKit 6.2 to 7.0, you must stop and upgrade servers at the same time.

You also need to upgrade your SafeKit license key as described in Section 4.18 page 82

Moreover, you must change the software error detector configuration (<errd>) of your application module which has been modified in 7.0 with the multi-module feature. This is explained later.

4.16.2 Upgrade Procedure

Follow the procedure explained in the *Upgrading/Fixing the SafeKit Package* section of the *SafeKit User's Guide* to upgrade from SafeKit 6.2 package to SafeKit 7.0.

During the upgrade procedure, you must reboot the system between the uninstall and the install procedures.

4.16.3 <errd> Configuration

In SafeKit 7.0, <errd> configuration is different from the one in SafeKit 6.2.

The following shows how to translate a SafeKit 6.2 configuration to a SafeKit 7.0 configuration:

- SafeKit 6.2 <errd> section for a process started in start_prim <user> script

```
<errd>
<proc name="cdsp" failureifexit="off" failureifexception="on" action="restart" />
<proc name="alw" failureifexit="off" failureifexception="on" action="restart" />
</errd>
```

- SafeKit 7.0 <errd> section for a process started in start_prim <user> script

```
<errd>
<proc name="cdsp" atleast="1" action="restart" class="prim" />
<proc name="alw" atleast="1" action="restart" class="prim" />
</errd>
```

Refer to <errd> in *SafeKit Configuration Guide* for more information.

4.17 Migrating from SafeKit 6.1 to SafeKit 7.0.4

4.17.1 Recommendation

When migrating from SafeKit 6.1 to 7.0, you must stop and upgrade servers at the same time.

4.17.2 Upgrade Procedure

- Uninstall SafeKit 6.1

  on UNIX, run `safekit uninstall` and reboot;

  on Windows, uninstall with "Add-Remove Program" and reboot.

  Refer to the 6.1 manual.

- Your original configuration files `userconfig.xml` and user scripts are kept in the `safekit/conf` and `safekit/bin` directories.

- Install SafeKit 7.0

  Refer to the installation procedure in the *SafeKit User's Guide*

- Do not install any pre-configured application module.

4.17.3 Configuration Migration

For more information on configuration, refer to *SafeKit User's Guide and SafeKit Configuration Guide.*

- Run `safekit module install -m <YourApplicationName> <SAFE>/Application_Modules/expert/upgrade.safe`

  (`<SAFE>=/opt/safekit` on UNIX and `SAFE=C:\safekit` on Windows).

- Copy your original SafeKit 6.1 configuration files in the following directories: `<SAFE>/modules/<YourApplicationName>/conf`

  `<SAFE>/modules/<YourApplicationName>/bin`

- Remove, from your old `userconfig.xml <proxyudp>` tag since it is not supported in SafeKit 7.0.

- Remove, from your old `userconfig.xml` in the `<errd>`, `<vhost>`, `<custom>`, `<proxyudp>` sections, any references to special scripts with an absolute path using the old SafeKit 6.1 path. Instead, put these special scripts in `<SAFE>/modules/<YourApplicationName>/bin` and set inside `userconfig.xml` relative path to this directory.

  For more information, see the `<errd>`, `<vhost>`, `<custom>` tag configurations in *SafeKit Configuration Guide*.

- Remove from your old start and stop scripts in `<SAFE>/modules/<YourApplicationName>/bin` any references to special scripts with an absolute path using the old SafeKit 6.1 path. Use `$SAFEUSERBIN` or `%SAFEUSERBIN%` in your scripts. See `<user>` tag 7.0 configuration in *SafeKit Configuration Guide* for more information.

- Remove `<http />` and `<snmp />` from your `userconfig.xml`.

- If you have modified `snmpd.conf` or `httpd.conf`, copy your modifications in `<SAFE>/web/conf/httpd.conf` and `<SAFE>/snmp/conf/snmpd.conf`. See *SafeKit User's Guide* for more information.

- See Section 4.16 page 80 to complete the migration.

- Run `<SAFE>/safekit config -m <YourApplicationName>`

- Reboot

⇨ Use the new SafeKit console, SafeMonitor as described in *SafeKit User's Guide*.

Once tested, you can package your migration in a new application module with `<SAFE>/safekit module package –m <YourApplicationName> <YourApplicationName>.safe` (absolute path is mandatory for `<YourApplicationName>.safe`). Before packaging, set "title" and "description" inside the file `safekit/modules/<YourApplicationName>/manifest.xml`, so it can be presented correctly in the "Server Admin" tabbed panel of the SafeMonitor.

If you put `<YourApplicationName>.safe` package in `<SAFE>/safekit/Application_Modules/`, then your application module will be displayed in the "Server Admin" tabbed panel of the SafeKit console as an installable Application Module, with the title and description previously set.

Without the "web" directory and the "index.lua" file after a migration, you will get in the "Quick Configure" tabbed panel, the `userconfig.xml` file. For more information on `index.lua`, see *SafeKit Configuration Guide*.

4.18 Upgrading SafeKit 6.x License Keys

You cannot use a SafeKit 6.x license key with SafeKit 7. An upgrade must be ordered to get a permanent license key.

In the meantime, you can get a free one-month SafeKit 7 evaluation license key from `http://www.evidian.com/safekit/requestevalkey.php`.

See "Install License Key" in the *SafeKit User's Guide* for more information.

Table of Contents