

The InterSync & InterMezzo HOWTO

by the project members

\$Revision: 1.6 \$ \$Date: 2002/09/25 07:19:41 \$

This document explains the configuration and operation of the InterSync daemon on Linux.

Contents

1 Acknowledgements	1
2 Disclaimer and License	1
3 More information	2
4 Installing and running InterSync	2
4.1 Overview	2
4.2 Getting the packages	2
4.3 Building InterSync and the InterMezzo file system	2
5 Running InterMezzo	3
5.1 Quick start	3
5.2 Details	4
6 Configuring InterSync	4
6.1 Config files	4
7 Debugging	5
8 Contact Information	5

1 Acknowledgements

Among the authors of this document are Peter J. Braam <braam clusterfs.com>, Gordon Matzigkeit <gord fig.org>

2 Disclaimer and License

InterMezzo is an experimental file system. It contains kernel code and daemons running with root permissions and is known to have bugs. Please back up all data when using or experimenting with InterMezzo.

InterMezzo is covered by the GPL. The GPL describes the warranties made to you, and can be found in the file COPYING.

Copyright on InterSync is held by Cluster File Systems, Inc.

Copyright on InterMezzo kernel components is held by Cluster File Systems, Inc, Peter J. Braam, Stelias Computing, Carnegie Mellon University, Phil Schwan, Los Alamos National Laboratory and Red Hat, Inc, TurboLinux, Inc., Tacitus Systems, Inc. and Mountain View Data, Inc.

InterMezzo is a trademark of Stelias Computing. It may be used freely to refer to the software on the `\url{http://www.inter-mezzo.org}`{InterMezzo Web Site}

3 More information

There is a detailed description of the InterSync design available on the WWW. Look at `http://www.inter-mezzo.org/docs/intersync.pdf`. The kernel code is described in `http://www.inter-mezzo.org/docs/intermezzo-2001.pdf`.

These documents describe how the daemons function and the purpose of special intermezzo files such as the Kernel Modification Log (KML), which contains the replication log, the Local Modification Log (LML) and the received file (RCVD).

4 Installing and running InterSync

4.1 Overview

The InterSync daemon depends on a kernel that has the InterMezzo file system installed. There are also some utilities to make and manage InterMezzo file systems.

4.2 Getting the packages

You should download the following RPM's or deb's from the web site:

1. intersync
2. curl
3. an intersync tar ball (`intersync-X.Y.Z.tgz`) to build the kernel module.

You will find that intersync depends on the *glib2* and *curl* packages.

InterSync is also available from CVS, see the CVS instructions on the InterMezzo web site and use module **izo**, (**NOT module intermezzo**).

4.3 Building InterSync and the InterMezzo file system

In order to build a kernel module for your kernel, you need to have a configured kernel source tree which includes the dependency files generated during the kernel build. Install the kernel and sources as suggested by your distribution:

Proceed by first preparing your kernel sources, and then building InterSync and the module:

```
<![CDATA[cd /usr/src/izo./autogen.sh ./configure --enable-linuxdir=/lib/modules/2.4.9/build cd fs24make
]]>
```

Configure should correctly detect if user mode linux is being used. You can also use your own configured kernel source:

```
<![CDATA[cd /your/source/linux  make distclean cp your.config .config make oldconfig dep
]]>
```

5 Running InterMezzo

5.1 Quick start

By default, there is almost nothing to configure. Decide on a server export (e.g. an empty formatted ext3 file system or tmpfs) and a client cache (e.g. a disk file system or tmpfs):

1. Create a new export file system on the server. This creates an ext3 (journalled ext2) file system on */dev/hda5*.

```
mke2fs -j /dev/hda5
```

2. Mount your file system on the server and the client. On the server:

```
mount -t intermezzo /dev/hda5 /exports/server
```

On the client:

```
mount -t vintermezzo none /imports/server
```

3. start intersync on the server and client: On the server:

```
intersync /exports/server
```

On the client:

```
intersync --server="server-hostname" /imports/server
```

You should now be able to use the file system on the client and server and see data move to the other side. [There are still issues here. If you can reproduce a bug in a simple scenario, please report it.] Note that there are *two* intermezzo file system types: **intermezzo** is used for device based file systems, **vintermezzo** is used for running over systems like *tmpfs*. Both intermezzo and vintermezzo can filter over other file systems, with the mount option *-o cache_type=fs-name*. The default *cache_type* is:

1. mount as **intermezzo**: cache type **ext3**.
2. mount as **vintermezzo**: cache type **tmpfs**.

Running both client and server on one system is possible two and can be done using the same commands.

5.2 Details

- Intersync creates a configuration and cache directory under `/var/intermezzo-X`, where `X` is the channel number.
- In this directory is a an apache configuration file and a symbolic link to the mountpoint. Httpd is simply running as:

```
httpd -f /etc/intermezzo-X/intersync.conf
```

- Httpd uses a library of CGI scripts, which are installed in `/usr/lib/intermezzo`.
- `/var/intermezzo-0` also contains a file named *sysid*, which contains a UUID identifying the intersync.
- `/var/intermezzo-X` on the server contains a file named *profile*. The profile file is fetched by the client and tells the client what fileset it is mounting. Do not change this from rootset, since you will have to do other configuration work.

6 Configuring InterSync

6.1 Config files

The default behaviour of InterMezzo is that *intersync*, the file server when running on the server or cache manager when running on the client generates all configuration files needed.

However, the details are available below:

To ease the mounting of InterMezzo filesets add one of the following to the `/etc/fstab` file. For testing and developing using a loop device as the cache is easiest:

```
/tmp/cache /izo0 intermezzo loop,fileset=fsetname,mtp=/mnt/izo0,
prestodev=/dev/intermezzo0,cache_type=ext3,noauto 0 0
```

where `/tmp/cache` is a file associated with a loop device, `/izo0` is a mount point (a directory), `fsetname` is the name of the fileset and `/dev/intermezzo0` is the name of the presto device. The creation of the cache file and the presto device is explained in the examples at the end of this section. The kernel must be configured with loopback device support enabled to do this.

Using a genuine block device is a little easier, because you do not need to set up a loop device. To use the block device `/dev/hda9`, the `/etc/fstab` file should contain:

```
/dev/hda9 /izo0 intermezzo fileset=fsetname,mtp=/izo0, prestodev=/dev/intermezzo0,cache_type=ext3,noauto
0 0
```

NOTE:

While the lines may wrap in this document the `/etc/fstab` entry should be a single line. The same holds for the following examples.

the mount point needs to be explicitly passed in the options (future versions of mount will not need this).

Other files The file `/izo0/.intermezzo/fsetname/kml` contains kernel modification log (aka the KML) which keeps track of all of the changes made in an InterMezzo filesystem. The file

`/izo0/.intermezzo/fsetname/last_rcvd` is the `last_rcvd` file which keeps track of the distributed synchronization file. In the current release of InterMezzo, the `KML` and `last_rcvd` files need to be created (usually by running `mkizofs`) before first mounting an InterMezzo filesystem.

Symlink `/var/intermezzo/cache` to the top directory of your mounted InterMezzo cache:

```
ln -s /izo0 /var/intermezzo/cache
```

7 Debugging

To help us find bugs we need logging information. The logs come in two places, from the kernel in `/var/log/messages`, and from `intersync` to `/var/log/intersync/`.

The kernel debugging log slows things down enormously and is activated with:

```
echo 8192 > /proc/sys/intermezzo/debug
echo 1 > /proc/sys/intermezzo/trace
```

Intersync debugging messages also slow things down enormously, and are activated by passing a `-debug X` parameter on the command line. `X` is one of (or a sum of) the values in this table:

Symbol	Value	Messages
<code>D_INFO</code>	1	Informational and startup messages.
<code>D_KML</code>	2	KML related messages.
<code>D_UPCALL</code>	4	Upcall tracking.
<code>D_HTTP</code>	8	HTTP messages.
<code>D_REINT</code>	16	Reintegration sizes and timings.

8 Contact Information

The InterMezzo web site is

<http://www.inter-mezzo.org>.

General questions about InterMezzo can be sent to

intermezzo-discuss@lists.sourceforge.net.

All InterMezzo related mail lists are archived, follow directions on the InterMezzo web site, so it may be worth checking here to see if your question has already been answered.

Bug reports should be filed on sourceforge. Please include the version of InterMezzo you are using and a description of your system configuration and the problem observed.

Also, please include all relevant logs:

1. `/var/log/messages`, and
2. `/var/log/intersync/`
3. for server and clients.