



# Stxb Use case discussion

# Use case 1

I am an organization with a site who is working with StarlingX. I want to provide a local mirror for my employees to use so they don't have to download all artifacts from the external internet every time.

I want to create an automated job which I will run daily to download all artifacts from the Internet (if they do not already exist) and place them in directory

`/export/mirrors/starlingx.`

Assume the user running the automated job has permissions to write to `/export/mirrors/starlingx`

- *We might need to consider building without the use of a container*

# Use Case 2

I am a developer within an organization which has a local mirror of artifacts available in `/import/mirrors/starlingx`. I want to build a StarlingX ISO without downloading rpms or src.rpms from the external internet.

the remote mirror will contain all binary rpms along with their corresponding src rpm, such that if no changes all the building of an ISO would be done from the mirror rpms, no actual local rpm build required unless it changes

# Use Case 3A

After doing a build, I just performed a repo sync, and the .lst files were not updated. If I perform another build of the ISO, will the system attempt to download external artifacts, even though nothing has changed

# Use Case 3B

After doing a build, I just performed a repo sync, and noticed that the .lst files have been updated. If I perform another build of the ISO, will the stale mirror content be detected and the new additions downloaded (while not re-downloading anything which has not changed)?

# Use Case 4

I have manually made a change to a puppet manifest in sysinv (stx-config git) but want to test before I commit anything.

How can I build a new ISO with my changes incorporated?

I would say not just puppet manifest, but any change to files in the repos.

# Use Case 5

I want to test StarlingX with my own additional program (foobar.x86\_64.rpm) on the ISO.

How would I perform a build with this file added?

# Build Avoidance

Needs to be included