Unison Sanity Check Instructions for SQA

Document Change History

Version Number	Date	Contributor	Description
V0.1	6/20/2018	Harrison Markarian	Initial version.
V0.2	5/17/2019	Harrison Markarian	Modified some formatting to be easier to read. Added additional instructions for other sanity tasks
V0.3	6/4/2019	Harrison Markarian	Edited heading. Other small editing changes
V0.4	6/18/2019	Harrison Markarian	Added instructions for updating yumrepo per SPR176574
V0.5	7/29/2019	Harrison Markarian	Added a note in the U1709 install instructions about verifying the issue of SPR176249
V0.6	8/19/2019	Harrison Markarian	Added a small section about making sure calfiles dir is empty before running DCV
V0.7	11/04/2019	Harrison Markarian	Added the step of running checkers on all instruments before full DCV

The function of these instructions is to provide SQA with the tools needed to install Unison releases from a customer perspective while also considering the different steps taken from the SQA perspective. The idea is to emulate the installation as close to how the customer does it as possible by following the same steps they would take to install the latest software, and to mimic all possible ways we know of that a customer can attempt to install a release.

This document is designed to change over time as SQA continues to utilize it and find new ways to more efficiently test incoming releases. Use this document not just to validate our releases, but to find new ways to improve our process and plug holes before we run into them during test.

There are three main sections of this document, as indicated by the contents below:

Contents:

Important: Despite the above processes requiring different steps, it is *essential* during an update release that we do test the update functionality as well as doing a fresh install. This does not need to occur on every SQA tester, but the update function must be tested somewhere. This ONLY matters for U1709 and beyond as that is where "yum" becomes available.

SQA can create a number of scenarios for a testing environment in which to conduct the sanity test:

A) Set up a regularly-used SQA (or otherwise) workstation to install or update the release locally (this will be the most common methodology), or;
B) Install a clean disk that will therefore be completely off the network, or;
C) Use a special sanity check workstation with a regularly-used sanity disk

Installing Unison U1709 and Beyond (Using Yum)

Installing Unison software uses "yum" – the standard Linux tool for installing software from a central repository. It is distributed in packages (components) that can be updated individually as needed. This may be relevant for example in validating firmware changes and updated revisions between old and new releases. Without having you completely erase and install a different Unison release; you would just install relevant packages. This is designed to be run with U1709 and beyond and will **NOT WORK** if installing from before U1709.

From here on until otherwise specified commands must be run as root.

Check off (or fill in if this document is not printed out) the boxes as you progress. This is for your own purposes and **not an official process checklist** that will be posted when complete.

Removing Previously Installed Releases

The main idea of doing a sanity test is making sure the packages will install properly and we have functional testers by conducting a fresh install. As mentioned on the main document page, it is

nevertheless important to test the update feature for any release we test that have a prior version (see the section "**Updating Preexisting Installs of Unison**" for more information.

This section details how to do a new install, so if you DO have a previous version of the release on your tester, remove it:

- ☐ # sll (this sets the links in /opt/ltx to local links, this will be addressed in the following section)
- ☐ # yum erase '*unison-release*'

More specifically, remove all versions of the release to be checked:

Example: # yum erase 'unison*U1709*'

Preparing to Install:

Before running yum commands and especially before running any yum installs make sure you are pointing to local dirs and links in /opt/ltx. This step is not necessary off the network in particular on a special sanity check disk (you may benefit from taking a picture of what the links look like before using this command to verify later):

- ☐ # sll

Go to /opt/ltx and ensure that its contents are pointing to local directories (like releases.local).

NOTE: running "sll" multiple times before restoring the links may lead to broken links, duplicate directories, and other issues. Always make sure you are not in local mode before running this script.

Now begin removing plugins/drivers that the workstation typically has:

- ☐ # su (root)
- ☐ # rpm -qa | grep –i ltx

Remove packages that contain "ltx" in their name from the previous command

- ☐ # rpm –el (--nodeps) (--noscripts)
  - This step can vary depending on the release, but drivers such as gpib or tester drivers should be removed to ensure they reinstall correctly.

Configuring Access to Yum Repositories:

You will need to either install or update unison-yumrepo. This provides tools to configure access to the software repositories.

Note: if you have previously installed the yumrepo on your tester in the past, there is not need to run these commands again, as it should still be there

☐ # yum install http://sr76.com/unison-yumrepo.rpm

Run the ltxcRepoTool (located in /usr/bin if you have issues finding it). This will prompt for an email address that is registered for access to download the Unison software (good.install@xcerra.com)

☐ # ltxcRepoTool –ltxc

Updating yumrepo if Needed:

On infrequent occasions, yumrepo itself may be upgraded to a different revision. If this is the case and you are aware of it, follow these steps to ensure you are using the latest version of yumrepo.

Remove unison-yumrepo

☐ # yum erase unison-yumrepo

Install the current production version

☐ # yum install http://sr01.xcerra.com/unison-yumrepo.rpm

Configure yumrepo and use good.install@xcerra.com

☐ # ltxcRepoTool -ltxc

Switch to the test repository and upgrade unison-yumrepo

☐ # yum update unison-yumrepo

Check that you still have access to the production repository

☐ # yum repolist

☐ # ltxcRepoTool -lc

Unison repository email: good.install@xcerra.com

Cached email address: good_test.install@xcerra.com

Installing Unison

Before installing on an SQA system, make sure to be using the Testing repository to take and install the newest version of the build you are testing from there:

☐ # /user/tools/bld_utils/bin/setRepo -t (the –t signifies “testing”)

Once the yum repositories have been correctly configured you can now access the Unison releases to install using yum. You may need to update the yum cache before installing for it to re-evaluate what is on its servers:

☐ # yum clean all

For the purposes of this instruction U1709 will be used. You will obviously need to substitute this with the release you are actually installing and testing.

List the available releases by running:

☐ # yum list 'unison*release*'

Determine your release to be installed and run:

☐ # yum install unison-{OS-version}-release (ex: # yum install unison-U1709-release.x86_64)

- When this is installing, an additional point to look for as part of the check is to ensure that all packages are up to date. This is a case by case situation depending on the release of course, so the specific packages to look for would be suggested to the team at that time. For example, we might be looking for "unison-U1709-core-8" as opposed to earlier revisions such as "core-5," so when you are seeing what is being installed and you see "unison-U1709-core-8" you'd know that it is up to date for that release.
- You can check /u/dir for what is currently being used for the local install and in the testing repo (again, the build would be made clear to SQA so you would know what packages to check).
    - For example, /u/dir/U1709-RC20/linux/pkgs would contain current information for those packages to look out for during install
- Also, you can run "rpm -qa | grep -i" for the latest and current version of what we need to be installed

If you are actually installing on tester hardware, driver software can be found by running:

☐ # yum list 'unison*tester*'

Determine your tester type and select the appropriate drivers release:

☐ # yum install unison-U1709-tester-{TesterType}

Post Installation

You will now need to run the postinstall script to complete the configuration once Unison is installed:

☐ # /releases/{U1709}/common/scripts/postinstall

Accept any of the prompts (which should be set to "Yes" by default) regarding third-party software and otherwise.

You will also need to obtain a license file in order to run any tests.

> **Note**: similar to yumrepo, SQA configured testers in local mode likely still have their license file, so this step may not be necessary if you know it is already installed

☐ # cd /opt/ltx/site
☐ # wget http://ltx.com/u/sw-opt/site/{site e.g. **westwood**}/LicensingInfo.xml

Config_Tester

Important: The steps below will explain config_tester, which is likely **already set up** if you've A) run this script before on your tester or B) you are running the install on an SQA workstation where the tester should be configured already. This step is more relevant if you are performing a completely clean install on a brand new disk, for example.

On tester workstations or IMA hosts you must now run config_tester. This will set up the workstation as a tester controller and give it a recognizable name. If you choose not to give it a name as an argument the default is to use the workstation name. It's **optional** for our purposes to give it a specific name as it has no hindrance on running the actual testing.

For example:

☐ # /opt/ltx/ltx_os/service/config_tester **diamondx4**

CONFIGURING: LTX tester
TESTER NAME: diamondx4
HOST NAME: newhost72
LTX VERSION: U1.0.0
Okay to continue: ([Y]/N): **Y**
Enter the local name of the nic directories. It should be in a partition large enough to hold dlog data and it should be shared r/w…

If you type <return> the default value will be used:
/opt/ltx_nic

Enter the network name for the tester.
If you type <return> the following default will be used:
/net/newhost72

/opt/ltx/ltx_os/testers/diamondx4 will be set to:

/net/newhost72/opt/ltx_nic
Do you wish to continue? ([Y]/N): **Y**

CREATING: /net/newhost72/opt/ltx_nic/dlog
CREATING: /net/newhost72/opt/ltx_nic/dp
COMPLETED: /ltx/service/config_tester

"You must now halt and re-boot the tester" **(Don't yet)**

It tells you to reboot the tester but before you do you should go to:

- ☐ /ltx/default_sims/<TESTER_TYPE>/pwrup and find the corresponding unison_config file. Copy this file to /ltx/testers/<TESTER_NAME>/pwrup
  Make sure to check the permissions of the unision_config file. Use chmod 777 should the permissions be lacking.

- ☐ Now reboot and login as service.

User Account Setup

Important: like config_tester, this step may be skipped if your tester's accounts **are already set up**.

Use the following script to help you with setting up user account settings. This will look for shell startup files, environmental variables, licensing setups, and directory trees required by Unison. Again, this is a sequence you may skip on a regular SQA workstation where you have only set the local links, but would be mandatory on a new disk. However, you can run through these steps anyway if you wish to ensure their functionality.

- ☐ # /opt/ltx/ltx_os/apps_support/com/account_config

Once that is run and you've run through it, open the service user's .cshrc file using any text editor

- ☐ For example: # emacs ~/.cshrc

And check that it has the following two lines:

- ☐ **setenv LM_LICENSE_FILE /opt/LTXflexnet/license.dat**
  setenv XLM_ENABLE_LM_LICENSE_FILE 1

In order for cmiService to work when using the Unison launcher, add the following line to the same file:

- ☐ **setenv LD_LIBRARY_PATH /opt/ltx/ltx_os/lib:${LIBRARY_PATH}**

- ☐ Now logout and log back in as service.

Installing non-component release packages

Install the release as usual. This should consist of doing an initial install and an update install. You should be pointing at the test repo at this point.

Get a list of all packages that are available. These are packages that do not get automatically installed by running "yum install unison-U1903-release".

Example:

☐ # yum list available '*U1903*'

Install the packages that were listed by the previous command. This is typically 3pdi and plugin-ucst. Be aware that only one "tester" package may be installed on a given tester.

Tester Validation

At this stage the workstation should be prepared to run tests, and the best place to start is by launching SMC+. No need to set_os or anything, the only release it should be looking at is what is on the tester, so just run launcher –f –m and run either a full suite of diags or perhaps more specific instruments depending on the focus of the sanity check, if any. If you wish to change to other installed releases on the tester, you must use -nih (meaning "not in house") after set_os.

Also significant before launching is ensuring that your calfiles directory is either empty, moved to a different name until you are done installing, or however you wish. Just make sure the tester is using a clean calfiles dir before running as an extra step of validation.

When ready, bring up SMC+:

☐ # launcher –f –m

IMPORTANT: do not run the full suite of checkers, cals, and verifies yet. We need to ensure that the checker for each instrument will work independently of whether or not there are any calfiles yet.

☐ Run all checkers on your respective tester on their own

THEN:

☐ Run the full DCVs, every instrument on the machine check, cal, and verify as you normally would for a full SQA test pass (a test pass for sanity will almost always accompany a major release anyway, so fill out your DCV results there)

Look for error messages, unusual failures, or anything out of the ordinary that might appear in the terminal or otherwise. It's always possible to miss a step or that something simply isn't working correctly. Always report anything odd that you see even if you're not sure that it's considered out of the ordinary.

☐ Also be sure to check OpTool and ensure it loads properly and run through the tools to ensure that they open up and load. Open up the **Help** menu in the top right corner and select each option and ensure they function as well (does the link to the documentation work)

☐ Make sure the **Help** menu contains an item called “Submit SPR” and that it does NOT contain an item called “Crash Report”

SPR176249 Validation:

Per the above SPR, it is also important to verify that, after running postinstall specifically, that you check the **/etc/modprobe.d** directory for its contents when running sanity on DxV systems. If it contains “blacklist.conf” or a file of a similar name, be sure to report this and verify its significance to the install packages. Per a full sanity test, this validation (at this point) only matters for DxV systems that use a 34461A meter, as even with the blacklist file and a 3458 meter, you would not see a problem.

☐ If your DxV is equipped with a 34461 AND a 3458 meter, be sure to test diags using both, or at least ensure that SQA has tested each on two different DxV systems.

Once you are done, make sure to reset the /opt/ltx links if you are on an SQA workstation using a typical SQA disk:

ffl

Check /opt/ltx to ensure nothing is still pointing to local directories.

Installing Unison Prior to U1709

Follow these instructions if you are doing a sanity check on a regular SQA disk and workstation. This means that you need to remove the typical packages that are installed for the tester to work and run tests. Note that there may be additional packages to remove not specifically containing "ltx" in their name but this can be determined by the specific release being tested (most packages of significance will have "ltx" though.

Another important note is that while this method works with most releases, **it is only meant to be used** with releases from **BEFORE** U1709 (as in U1703 and backward such as U4.4.1 or U5.2.2). If you are testing U1709 and beyond please refer to the previous set of instructions using yum, as that is the designated method of installing those releases.

From here on until otherwise specified commands must be run as root.

If /opt/ltx/releases exists, verify that it is not a link:

- ☐ # ls -ld /opt/ltx/releases

Inspect the contents of /opt/ltx if it exists:

- ☐ # ls -ld /opt/ltx/*

Set local links for /opt/ltx if they are not set already:

- ☐ # sll

Remove any existing versions of the release you are testing:

- ☐ # yum erase 'unison*U1709*' (again, we're not using yum in this install method, but this is if you wish to clean up previous installs.
  In /u/cdarea there is an "uninstall" script to remove releases prior to U1709

Go to /opt/ltx and verify where directories are pointing to (i.e. "releases.local")

Package Check

Unlike installing in U1709 and beyond, packages that you remove need to be manually installed if you remove them… Installing prior to U1709 will not reinstall the requisite packages. The script that you will use in the next section should automatically run package_check once complete, but if you wish to run package_check manually, you can find it in:

/opt/ltx/OS_TAR/{release}/{OS}/pkgs

You MUST have the needed packages installed to test the release. When you run the script in the next step, you will see that it will tell you which installed packages differ from what is found in the release

media. You need to manually uninstall incompatible packages and reinstall those indicated under the "Included with Release" column for your release.

SQA Local Install

Now there is another unique script that will complete the next sequence of steps as well as provide the unit test tree to cd to once you are ready to run unit tests:

- ☐ # which SQA_local_installation (this produces the directory tree of where the script lives)
- ☐ # su (to root if not already)
- ☐ # /user/tools/LINUX/bin/SQA_local_installation

This script will prompt you for the build to be installed as it finds them from /u/dir. As it iterates through, pay attention to the output and answer the prompts about the correct build to install. Once it finishes, you will notice it asks you to exit in order to run tests and outputs the network path for the unit tests. Go there prior to beginning testing (assumed you are running unit tests).

Tester Validation

Make sure that AFTER you have set local links **not** to run msos, as this will reset /opt/ltx to the incorrect stage. There is no need to run this as the installed release should be the default if everything works correctly. If you wish to change to other installed releases on the tester, **you must use** -nih (meaning "not in house") after set_os.

Example: # set_os -nih U5.2.2

Attempt to launch SMC+ (if this tool is available in this release to run diags, if not, use optool). The release should now be the default that it chooses when launching.

- ☐ # launcher –f –m

Run diagnostics on any or all relevant instruments to verify that they run and function as expected.

You may also run unit tests, which if you have installed on a typical SQA workstation and disk you will be able to access the unit test dir dependent on what release you are testing. This will vary.

After doing all appropriate testing, fix the fmi links:

- ☐ # ffl

This will revert /opt/ltx and its directories back to an SQA ready state. Check to make sure all the directories appear as they did before and are not pointing to local links or directories.

Update/Rollback Pre-Existing Unison Release

***Only compatible with U1709 and beyond**

Whether you have an installed unison release on the tester already or you are sanity checking the process of updating a release immediately after installed the main release, follow these steps to perform the process.

The way in which customers can choose to install the latest version of a release might be to exclusively use the yum update feature, hence why we need to test it and ensure that it works with any release that is not a 1.0 initial release.

From here on until otherwise specified commands must be run as root.

A note on using yum: it will not read the contents of the repository every time you run a yum command. It instead keeps a cache and periodically updates it. You may need to clean out the database and force yum to re-evaluate what is on the servers.

Example: # yum clean all

Preparing to Update

The process of updating previous installs of Unison require that a release be installed prior to the update. If you currently have the absolute latest version of the release installed after following the procedure "**Installing Unison U1709 and Beyond**" you will need to erase it before continuing. Skip this step and move on to Installing the Previous Release if you already have no traces of the release in test on your tester. If you already have the previous release installed, move on to Updating to the Latest Packages.

- ☐ # sll (this sets the links in /opt/ltx to local links, this will be addressed in the following section)
- ☐ # yum erase '*unison-release*'

More specifically, remove all versions of the release to be checked:

Example: # yum erase 'unison*U1709*'

Installing the Previous Release

Configure access to the production repository to install the last release. Note: this assumes you have access to /user/tools, if you do NOT have access see a bit further ahead for what to do.

- ☐ # /user/tools/bld_utils/bin/setRepo -p (the –p signifies "production")

The steps that follow will be approximately the same as the standard installation of Unison on Linux, so ensure that, if you are on an SQA workstation on the network, to first set the local links for /opt/ltx:

☐ # sll

Determine the release being tested/updated and install (remember to use yum list if you are unsure):

☐ # yum install unison-{OS-version}-release (ex: # yum install unison-U1709-release.x86_64)

Determine your tester type and select the appropriate drivers release:

☐ # yum install unison-U1709-tester-{TesterType}

You'll notice the same pattern being followed as earlier in this document about installing Unison on Linux; these are the steps to follow until you are sure the release is successfully installed on the workstation before moving on to the next series of steps for updating. As always, use # ffl whenever you are done with the testing to reset the links in /opt/ltx.

Update to the Latest Packages

If you are starting here because you're updating the packages that were already on the tester, **make sure** you're in local mode by running "**sll**"

Now switch your machine to use the Testing repository:

☐ # /user/tools/bld_utils/bin/setRepo -t (the –t signifies "testing")

☐ # yum update

Again, if you do NOT have access to /user/tools, follow these steps in order to use those commands:

☐ # cd /tmp

☐ # wget http:/ /pdeng.ltx.com/u/tools/bld_utils/bin/setRepo

☐ # chmod 755 setRepo

☐ # . /setRepo -t (for testing)

☐ # ./setRepo -p (for production)

Remember, this assumes you've already installed and are using either the clean disk install or the SQA local links method, or otherwise. It's going to update the release currently on the machine based on what is AVAILABLE to update that yum finds in the repository.

For update releases an important aspect to check is what tools like cmiService outputs when run; obviously checking the functionality of unit tests or diags is necessary too but if there are specific updates to firmware revisions or related changes those should be singled out. This is why, during update testing, it is often important to install the original (or previous) release first, save cmilogs and all relevant data/logs, THEN update and compare them with what is output from the update.

Rollback

Also worth noting is the ability to revert BACK to components from a previous release, which is done simply through yum (don't regard this as a step in the process as it is just a potential yum command for your pruposes):

```
# yum downgrade compName
```

You may instead need to use yum history commands in order to revert backwards. Plugins are a bit more complicated due to their specificity. Again, yum has the tools to assist with this too. You use "yum history list" to find the ID of the system prior to your install. Then run "yum history rollback PriorID"

Here is an actual command sequence where an inconsequential package is installed (zsh-html), and removed:

- ☐ # yum install zsh-html.x86_64 **(this will create an ID number once it is installed)**
- ☐ # yum history list **(shows a list of installs with the latest install at the top of the ID list)**

Assume that the installed package is assigned ID # 12. The LAST install before that would have ID # 11. For the purposes of sanity checking make sure you're installing an update/package AFTER you install the original release or whatever the case may be. This was you have at least two distinct ID #s, the latest item installed and then the one you are reverting back to. For example:

- ☐ # yum history rollback 11

This will revert BACK to the ID # of the item that was installed at that time.