Hipster Handbook - Sun Ray Software

Contents

Some notes for installation of Sun Ray Software on OpenIndiana Hipster. Sun Ray Software is used to drive desktop sessions on SunRay thin clients (desktop unit: DTU).

NOTE:

This configuration is not supported by OI nor by Oracle.

NOTE:

Since GNOME is replaced by Mate in Hipster, installation is a little bit more difficult. For Sun Ray it is still necessary to use GNOME GDM and some other GNOME applications. But it is possible to use Mate desktop with its applications like pluma or atril.

1 Install Prerequisites

If you have installed new Hipster, it is necessary to install GNOME packages for GDM and its greeter-session. For user session it is fine to select ‘Mate Session’, it is no longer necessary to install other packages needed by Gnome Session (only Gnome packages which Mate session still uses).

Be sure to have Mate desktop environment installed:

```
# pkg install mate_install
```

Latest known working OpenIndiana GNOME packages versions are the following:

```
gnome-session@2.32.1
gnome-settings-daemon@2.32.1
gdm@2.30.7
libgnomekbd@2.32.0
```

These are still in hipster repo, but are obsoleted empty metapackages.

At http://pkg.toc.de/sunray/ you can find the old GNOME packages with newer release date, so that these should be able to install it on current Hipster. This publisher is provided and supported by community member Carsten Grzemba not by the OpenIndiana project. For this to work add this publisher in a way, that it takes precedence over the default openindiana.org publisher:

In order to install packages with other versions such as userland-incorporation, it is necessary to release the version locks

```
# pkg change-facet facet.version-lock.gnome/gnome-session=false
# pkg change-facet facet.version-lock.gnome/gnome-settings-daemon=false
# pkg change-facet facet.version-lock.system/display-manager/gdm=false
# pkg change-facet facet.version-lock.library/gnome/libgnomekbd=false
# pkg change-facet facet.version-lock.gnome/window-manager/metacity=false
# pkg change-facet facet.version-lock.library/desktop/gnome-desktop=false
# pkg change-facet facet.version-lock.cde/cde-runtime=false
# pkg change-facet facet.version-lock.library/motif=false
# pkg change-facet facet.version-lock.library/tooltalk=false
# pkg change-facet facet.version-lock.compatibility/packages/SUNWxwplt=false
```

configure the publisher

```
# pkg set-publisher --search-before=openindiana.org -g http://pkg.toc.de/sunray
 ↪  sunray
# pkg set-publisher --non-sticky openindiana.org
```

and install package sunray-essential from sunray publisher

```
# pkg install sunray-essential
```

This will install all necessary packages for using GDM and Sun Ray Software tools on OI Hipster.

2 Install Sun Ray Server Software

Sun Ray Software still can be downloaded at http://edelivery.oracle.com (registration required). Download the Software for Solaris x86-64 and for Linux x86-64. From the Linux software we take scripts for configure ISC DHCP server described later.

Extract the archives.

The Solaris packages require Sun DHCP server, which we do not provide anymore. To solve this problem we replace the dependency with isc-dhcp in the IPS packages SUNWutr and SUNWuto with the following script:

update_dhcp_dependency

```
#!/bin/ksh

# set -x

if [ -z $1 ] || [ ! -d $1 ]; then
echo "usage: $0 repodirectory"
exit
fi

if [[ $1 != /* ]]; then
   repo=$(pwd)/$1
else
   repo=$1
fi

republish="republish$$"

mkdir ${republish}

for p in SUNWutr SUNWuto; {
(cd ${republish}
  pkgrecv -d . --raw -s $repo $p || exit 2
  for d in $(ls $p); {
  (cd $p/$d
    sed -i s#service/network/dhcp#service/network/dhcp/isc-dhcp# manifest
    pkgrepo -s $repo remove $p
    pkgsend publish -s $repo -d . manifest
```

```
  )
  }
);
}
rm -r ${republish}
```

Run the script `update_dhcp_dependency <extractdirectory>/srs_5.4.5.0-Solaris_11plus.i386/IPS.i386` to replace the dependcy of package `service/network/dhcp` by `service/network/dhcp/isc-dhcp`.

Install IPS packages of the Sun Ray Software from the local repository

```
# pkg set-publisher -g <extractdirectory>/srs_5.4.5.0-Solaris_11plus.i386/IPS.i386/
 ↪  sunray
# pkg install SUNWut-srss SUNWut-srwc SUNWuti
```

Do **not** use `utinstall` script, because its will setup a sunray publisher by its own which will interfere with our publisher of GNOME packages.

Due to package dependencies following packages will be installed:

```
SUNWut-srss (sunray)                    5.4.5.0.38-0.0          i--
SUNWuta (sunray)                        4.5.4.0.38-0.0          i--
SUNWutdso (sunray)                      3.5.0.0.2-0.0           i--
SUNWutdsr (sunray)                      3.5.0.0.2-0.0           i--
SUNWutesa (sunray)                      4.5.0.0.44-0.0          i--
SUNWutgsm (sunray)                      4.5.0.0.44-0.0          i--
SUNWuti (sunray)                        4.5.5.0.38-0.0          i--
SUNWutid (sunray)                       4.5.5.0.38-0.0          i--
SUNWutk (sunray)                        4.5.0.0.44-0.0          i--
SUNWutm (sunray)                        4.5.4.0.38-0.0          i--
SUNWuto (sunray)                        4.5.4.0.38-0.0          i--
SUNWutps (sunray)                       4.5.0.0.44-0.0          i--
SUNWutr (sunray)                        4.5.0.0.44-0.0          i--
SUNWutref (sunray)                      4.5.0.0.44-0.0          i--
SUNWutscr (sunray)                      4.5.4.0.38-0.0          i--
SUNWutsrs (sunray)                      4.5.5.0.38-0.0          i--
SUNWutstk (sunray)                      4.5.0.0.44-0.0          i--
SUNWutsto (sunray)                      4.5.0.0.44-0.0          i--
SUNWutstr (sunray)                      4.5.0.0.44-0.0          i--
SUNWutsvt (sunray)                      1.1.0.0.3-0.0           i--
SUNWuttsc (sunray)                      2.6.4.0.11-0.0          i--
SUNWuttscd (sunray)                     2.6.4.0.11-0.0          i--
SUNWuttscr (sunray)                     2.6.4.0.11-0.0          i--
SUNWutu (sunray)                        4.5.0.0.44-0.0          i--
SUNWutwa (sunray)                       4.5.2.0.5-0.0           i--
SUNWutwar (sunray)                      4.5.0.0.44-0.0          i--
SUNWutwh (sunray)                       4.5.0.0.44-0.0          i--
SUNWutwl (sunray)                       4.5.0.0.44-0.0          i--
```

2.1 Configure isc-dhcp

On Hipster ISC DHCP replaced the Sun DHCP server, but that's not a problem as ISC DHCP server for Sun Ray Server was used on Linux already.

So Sun Ray Server on Hipster has to use DHCP scripts like on Linux. But it is also possible to configure DHCP manually and to skip the `utadm` step. Here an example:

```
ddns-update-style none;

# option definitions common to all supported networks...
option domain-name "sunray.lan";
option domain-name-servers 192.168.1.6, 192.168.1.1;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.1;
option space SunRay;
option SunRay.Intf code 33 = text;
option SunRay.Intf "nge1";
option SunRay.AuthSrvr code 21 = ip-address;
option SunRay.AuthSrvr 192.168.1.2;
option SunRay.AltAuth code 35 = array of ip-address;
option SunRay.AltAuth 192.168.1.2;
option SunRay.FWSrvr code 31 = ip-address;
option SunRay.FWSrvr 192.168.1.2;
option SunRay.NewTVer code 23 = text;
option SunRay.NewTVer "11.1.7.0_38_2016.10.27.13.09";
option SunRay.AuthPort code 22 = integer 16;
option SunRay.AuthPort 7009;
option SunRay.LogHost  code 24 = ip-address;
option SunRay.LogHost 192.168.1.2;
option SunRay.LogKern code 25 = integer 8;
option SunRay.LogKern 6;
option SunRay.LogNet code 26 = integer 8;
option SunRay.LogNet 6;
option SunRay.LogUSB code 27 = integer 8;
option SunRay.LogUSB 6;
option SunRay.LogVid code 28 = integer 8;
option SunRay.LogVid 6;
option SunRay.LogAppl code 29 = integer 8;
option SunRay.LogAppl 6;
option SunRay.sunray-servers code 68 = text;
option SunRay.sunray-servers "oi-sr.sunray.lan";
option SunRay.sunray-config-servers code 67 = text;
option SunRay.sunray-config-servers "oi-sr.sunray.lan";

default-lease-time 86400;
max-lease-time 86400;

class "sun" {
```

```
  match if substring (option vendor-class-identifier, 0, 4) = "SUNW";
}

log-facility local7;

host sr1.sunray.lan {
  hardware ethernet 00:14:4f:57:a0:c1;
  fixed-address 192.168.1.50;
  vendor-option-space SunRay;
}

subnet 192.168.1.0 netmask 255.255.255.0 {
  range 192.168.1.51 192.168.1.59;
}
```

Vendor specific DHCP options are described in the Sun Ray Software: Alternate Client Initialization Reqs Using DHCP document. The config file `/etc/dhcp/dhcpd.conf` has to link to `/etc/inet/dhcp4.conf` so that the `svc:/network/dhcp/server:ipv4` service can leverage it.

To switch Sun Ray Software to isc-dhcp we take scripts located in path `/opt/SUNWut/lib/dhcp/isc` from the Sun Ray Software package for Linux (SUNWuto-4.5-44.i386.rpm). Extract the package with `rpm2cpio`:

```
$ rpm2cpio SUNWuto-4.5-44.i386.rpm | cpio -idv <tmprpmdirectory>
```

Rename `dhcp_config_linux` to `dhcp_config_solaris` and replace the `sort` command with his GNU implementation `gsort` by appling this patch on script ‘dhcp_config_linux:

```
diff --git a/dhcp_config_linux b/dhcp_config_linux
index 001aa42..0165941 100755
--- a/dhcp_config_linux
+++ b/dhcp_config_linux
@@ -462,7 +462,7 @@ TranslateEther() {
    BEGAN=false
    VALUE=""

-   ls $UTDHCPDIR | sort -g | grep "SunRay-ether" > $TMPLISTFILE
+ ls $UTDHCPDIR | gsort -g | grep "SunRay-ether" > $TMPLISTFILE
    if [[ $? != 0 ]]; then
            rm -f $TMPLISTFILE 2>/dev/null
            return 1
@@ -552,7 +552,7 @@ TranslateSubnet() {

    if [ -z "$INFILE" ]; then
            # list all the subnet files
-           ls $UTDHCPDIR | sort -g | grep "SunRay-subnet" > $TMPLISTFILE
+         ls $UTDHCPDIR | gsort -g | grep "SunRay-subnet" > $TMPLISTFILE
            if [[ $? != 0 ]]; then
                    rm -f $TMPLISTFILE 2>/dev/null
                    return 1
@@ -616,7 +616,7 @@ TranslateInterface() {
    BEGAN=false
```

```
    VALUE=""

-   ls $UTDHCPDIR | sort -g | grep "SunRay-interface" > $TMPLISTFILE
+ ls $UTDHCPDIR | gsort -g | grep "SunRay-interface" > $TMPLISTFILE
    if [[ $? != 0 ]]; then
            rm -f $TMPLISTFILE 2>/dev/null
            return 1
```

put the scripts below `<tmprpmdirectory>/opt/SUNWut/lib/dhcp/isc` in `/opt/SUNWut/lib/dhcp/isc`

Reference the scripts by setting this link:

```
root@oi-sr:/etc/opt/SUNWut# ln -s /opt/SUNWut/lib/dhcp/isc dhcp
```

`utadm` command needs changes shown in the following patch:

```
From a1072acfffd91457d91cd6d202a988d88bc3fb8a Mon Sep 17 00:00:00 2001
From: Carsten Grzemba <cgrzemba@opencsw.org>
Date: Mon, 3 Feb 2020 10:58:31 +0100
Subject: [PATCH] apply changes for:
* change the dhcp config file name
* different ipadm argument names
* name IP interface utadm to refer to the IP address creator
* do not use /etc/hostname.<if>

---
 utadm | 41 +++++++----------------------------------
 1 file changed, 7 insertions(+), 34 deletions(-)

diff --git a/utadm b/utadm
index cffac27..448c171 100644
--- a/utadm
+++ b/utadm
@@ -116,7 +116,7 @@ UTDHCPSERVICE_SUCCESS=0
 UTDHCPSERVICE_DISABLED=3

 SVCADM="/usr/sbin/svcadm"
-DHCP_FMRI="svc:/network/dhcp-server:default"
+DHCP_FMRI="svc:/network/dhcp-server:ipv4"

 UTLIB="/opt/SUNWut/lib"
 UTSBIN="/opt/SUNWut/sbin"
@@ -207,6 +207,7 @@ function SetPlatformDependencies  {
     IFCONFIG_KEY_NETMASK="netmask"
     if [ -x /usr/sbin/ipadm ]; then
       IPADM_CONF=true
+          DHCPCONFIG="/etc/dhcp/dhcpd.conf"
     fi

     ;;    # end case SunOS
@@ -1783,14 +1784,6 @@ function DoAddNetworkConfig {
 DHCPONLY="N"
```

```
 for test in ${INTF_ALL}; do
     if [ "${INTF}" = "${test}" -a -f "${HOSTNAME_C}${INTF}" ]; then
-        #
-        # Need to catch the case where hostname.<intf> exists but the hostname
-        # defined in the file may not be configured locally in the hosts file.
-        #
-        if [ ! -f ${HOSTNAME_C}${INTF} ]; then
-            print -u2 "Error: missing \"${HOSTNAME_R}${INTF}\" file."
-            return 1
-        fi
     if [[ $OS == "SunOS" ]]; then
         NAME=`getIfname ${INTF}`
         if [ -z "${NAME}" ]; then
@@ -2567,9 +2560,9 @@ should an auth server be located by broadcasting on the
 ↪  network?" "Y"
     fi
     # Clear any pre-existing state on interface, then create
     # new persistent configuration
-        ipadm delete-ip ${INTF} 2>/dev/null
-        ipadm create-ip ${INTF}
-        ipadm create-addr -T static -a local=${IPADDR}/${MASKBITS} ${INTF}/v4static
+        ipadm delete-if ${INTF} 2>/dev/null
+        ipadm create-if ${INTF}
+        ipadm create-addr -T static -a local=${IPADDR}/${MASKBITS} ${INTF}/v4utadm
   else
     # ifconfig the new interface
     ifconfig ${INTF} plumb >/dev/null 2>&1 ;
@@ -2710,7 +2703,7 @@ function DoList {
 if [[ $BLOCKTYPE = "interface" ]] ; then
   if Intf=${UT_DHCP_INTERFACE} ; then
     if [[ ! -s ${HOSTNAME_R}.${Intf} ]]; then
-        IntfAddr=""*MISSING*""
+        IntfAddr=`ipadm show-addr -p -o ADDR ${Intf}/ | awk '{split($1,a,"/");
↪  print a[1]}'`
     else
         typeset H=`getIfname ${Intf}`
         IntfAddr=`getent hosts $H | awk '{ print $1 }'`
@@ -3151,14 +3144,6 @@ function DoPrint {

 INTF=${UT_DHCP_INTERFACE}

-    #
-    # Need to catch the case where hostname.<intf> exists but the hostname
-    # defined in the file may not be configured locally in the hosts file.
-    #
-    if [ ! -f ${HOSTNAME_R}.${INTF} ]; then
-        print -u2 "Error: missing \"${HOSTNAME_R}.${INTF}\" file."
-        return 1
-    fi
```

```
 H=`getIfname ${INTF}`
 if [ -z "${H}" ]; then
     print -u2 "Error: interface ${INTF} is partially configured.  Hostname not" \
@@ -3452,19 +3437,7 @@ function getIfname {
 awk '{print $1 ; exit}' "${HOSTNAME_R}.$1"
 return
 else
-    # With ipadm, there's no way to create a persistent
-    # configuration without it immediately being reflected in the
-    # runtime configuration, nor does ipadm provide any convenient
-    # way to find the IPv4 addrobj for the interface, so just
-    # check runtime configuration via ifconfig.
-    IPADDR=$(ifconfig $1 | awk '
-        /[     ]inet /{
-            for (i = 1; i < NF; ++i) {
-                if ($i == "inet") {
-                    print $(i+1)
-                }
-            }
-        }')
+    IPADDR=`ipadm show-addr -p -o ADDR $1/ | awk '{split($1,a,"/"); print a[1] }'`
 if [ -z "$IPADDR" ]; then
     print -u2 "Error in parsing of ifconfig addresses"
     exit 2
--
2.23.0
```

2.2 Note on Java Runtime Environment

The Sun Ray Software needs the Sun/Oracle JRE 1.7 which is shipped with the Sun Ray Software.

3 Config

You can run `utconfig` and `utadm` the way described in the Sun Ray Software document.

4 XScreenSaver

Latest Hipster delivers XScreenSaver only in 64-bit. The SunRay PAM module are shipped as 32-bit only so unlocking works only with 32-bit XScreenSaver. Newer Xscreensaver has the problem like some other newer Gnome and Mate components, that they expect, there is a main display :0. This isn't true on SunRay. XScreensaver prefer to crash on SunRay. That's why we need the XScreenSaver package with 32-bit bins from http://pkg.toc.de/sunray/.

It is more comfortable to use mate-screensaver. For PAM is only the module mate-screensaver-pam-helper needed. This is not an shared object, instead it is an executable and can be shipped in 32bit independent from the other commands.

5 GDM

the GDM service has to be enabled. lightdm should not run however.

```
# svcadm enable graphical-login/gdm
# svcadm disable graphical-login/lightdm
```

Login Screen (gdm-greeter) won’t reapear after logout

Contents

After logout the gdm-login won't reappear. Install the following script as `/opt/SUNWut/lib/gdm/revivesrse` and call it by the `/etc/opt/SUNWut/gdm/SunRayPostSession/helpers/revivesession` helper script.

```
/opt/SUNWut/lib/gdm/revivesrsession.py &
```

`/opt/SUNWut/lib/gdm/revivesrsession.py` contains:

```
#!/usr/bin/python3.9

'''
 Reconnect SunRay X-session: some times after logout, the gdm will not start login
 ↪  screen on the X-session and the DTU remains in state 26D
 INSTALL: - cp to /opt/SUNWut/lib/gdm/revivesrsession.py
      - create script /etc/opt/SUNWut/gdm/SunRayPostSession/helpers/revivesession:
        /opt/SUNWut/lib/gdm/revivesrsession.py &
'''

import subprocess as sp
import re
import pdb
import time
import logging

logformat = "%(asctime)s %(levelname)s:%(message)s"
logging.basicConfig(format = logformat)
logger = logging.getLogger(__name__)

logger.setLevel(logging.DEBUG)

time.sleep(7)

pid = sp.Popen(['pgrep','gdm-binary'], stdout=sp.PIPE).stdout.readline().strip()
logger.debug("GDM pid %s ", pid.decode())
dpl = [ p.split()[2].strip(b':') for p in sp.Popen(['ptree', pid],
 ↪  stdout=sp.PIPE).stdout.readlines() if b'/opt/SUNWut/lib/Xnewt' in p ]
logger.debug("Xnewt display {}".format( dpl))

# error -4 gdm-simple-slave not started for Display, no UT sessions
for sess in sp.Popen(['/opt/SUNWut/sbin/utsession','-px'],
 ↪  stdout=sp.PIPE).stdout.readlines():
logger.debug("{}".format(sess))
for t in sess.split(b';'):
    if b'STATE' in t: state = t.split(b'=')[1]
    if b'DISPLAY' in t: disp = t.split(b'=')[1]
if state == 0 and disp in dpl:
    # ok
    pass
else:
    logger.debug("restart display %s"  % disp.decode())
    sp.Popen(['/opt/SUNWut/lib/gdm/utgdmdynamic','-a', disp])
```

Contents

SunRay thin clients can be grouped to a multihead configuration, where multiple monitors can used for a single user session. For two clients this can be done with:

```
$ /opt/SUNWut/bin/utxconfig -m on -R 2x1 -S 0,1
```

Since GTK 3.10 multi screen setup is not supported anymore. The command line option `--display` whith a dot value greate 0 will not work any more (e.g.: earlier the command `firefox --display=:11.1` has started firefox on the second display).

metacity or marco will manage only the default screen. For multihead setup it is now necessary to enable ‘xinerama’ mode, so that the Window manager expands the screen across all DTU’s in a group.

```
$ opt/SUNWut/bin/utxconfig -x on
```

Refer to the original documentation from Oracle for configuration in general: Multiple Monitor Configurations