Deployment of SuperMap iServer 7C Tar Package on Linux

1. Installation environment

1.1 Hardware requirements
The minimum hardware requirements for installation of SuperMap iServer 7C on Linux include:
- Processor: 800MHz
- Memory: 512MB
- Free disk space: 20GB
- Network adapter: Network adapter installed in the system
- Display adapter: 64M display (with drive), OpenGL version: 1.5

1.2 Software requirements
- CentOS 5.6 or above, 6.x
- Red Hat (Red Hat Enterprise Linux 5.4 or above, 6.x)
- SUSE 11 or above
- Asianux Server 3
- Neokylin OS server version 5.0/6.0

Other software requirements
- JRE 1.6 or above (included in iServer 7C)
- SuperMap iObjects Java 7C for Linux (included in iServer 7C)

2 Installation and configuration of iServer 7C

Note: in this document, SuperMap iServer 7C is deployed in RedHad 6.4 OS
Linux: `rhel-server-6.4-x86_64 (Kernel: 2.6.32)`
SuperMap iServer 7C: `smiserver_700_10804_1592_linux64_chs.tar.gz`
The configuration of the installation of SuperMap iServer 7C on Linux has three main steps: `Install iServer 7R`, `configure license`, `access map`. The steps are shown below:

2.1 Install iServer 7C

1) Copy the installation package of acquired iServer 7C to Linux:

```
smiserver_700_10804_1592_linux64_chs.tar.gz
```

2) Unzip the installation package using the following commands:

```
[root@linux opt]# ls
smiserver 700 10804 1592 linux64 chs.tar.gz
```

Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, CHINA, 100015
Tel: +86-10-59896503  Fax: +86-10-59896666  E-mail: request@supermap.com
Website: www.supermap.com/en/
Grant clearance to installation package: `chmod 755 smiserver_700_10804_1592_linux64_chs.tar.gz`

Unzip the package:

```bash
.tar -zxvf miserver_700_10804_1592_linux64_chs.tar.gz
```

3) After unzipping, a new folder will be generated, shown as below:

```
[root@linux opt]# ls
smiserver_700_10804_1592_linux64_chs.tar.gz
```

4) Using command `cd SuperMapiServer7C` to enter the unzipped the folder:

```
[root@linux SuperMapiServer7C]# ls
BUILD_10804_1592 License.txt SuperMap_iServer_7C_Readme_Linux_CHS.pdf
```

2.2 Configure license

1) Enter the folder of license, (in the support folder of unzipped iServer folder)

```bash
cd /opt/SuperMapiServer7C/support/SuperMap_License/Support
```

2) Unzip the license installation program:

```bash
tar -xvf aksusbd-2.0.1-i386.tar
```

3) Enter the newly generated folder `cd aksusbd-2.0.1-i386` to run `./dinst` the following picture shows the installation completion (default usage time for 3 months)
2.3 Access map

1) Enter the bin folder of iServer 7C installation folder: `cd /opt/SuperMapiServer7C/bin`
2) Run the command: `sh startup.sh`
3) Type: `http://[localhost_ip]:8090/iserver/` in the browser to create new username and password
4) View world map:
3 Common problems

1. The unzipping of iServer 7C products has to be done in iServer 7C
2. If using remote tool like putty to activate iServer 7C, then using the remote tool to log in linux server, open a terminal, run:
   a)  X :n& activate xServer in No. n screen, n is an arbitrary. (e.g. 20, 30, etc)
   b)  export DISPLAY=:n.0
       set DISPLAY variable to the xServer of local machine, then activate iServer 7C
3. if fails to activate iServer 7C, the problems can be found using the following methods:
   a)  Check the completeness of libraries of iServer 7C

   i. Configure java.profile files (including JRE_HOME and Objects Java)

```
export JRE_HOME=/opt/SuperMapiServer7C/support/jre
export PATH=$JRE_HOME/bin:$PATH
export SUPERMAP_ROOT=/opt/SuperMapiServer7C/support
export LD_LIBRARY_PATH=/opt/SuperMapiServer7C/support/objectsjava/bin:$LD_LIBRARY_PATH
```

   ii. Type command source java.profile to make environment variable effective, then use the following commands to see if the configuration correct:

```
[root@linux bin]# java -version
java version "1.7.0_45"
Java(TM) SE Runtime Environment (build 1.7.0_45-b18)
Java HotSpot(TM) 64-Bit Server VM (build 24.45-b08, mixed mode)
[root@linux bin]# echo $LD_LIBRARY_PATH
/opt/SuperMapiServer7C/support/objectsjava/bin:
```

   iii. Switch to objectsjava bin folder, cd /opt/SuperMapiServer7C/support/objectsjava/bin
   type ldd libWrapj.so
   Check the completeness of needed libraries, if there is ‘not found’. If there is, it means the needed libraries of iServer 7C is incomplete, the rpm package needs to be installed. You can find the libraries from disk or iso from linux
iv. After installation of needed libraries, type again `ldd libWrapj.so` to check, if all of libraries are installed, restart iServer.

b) Check iServer log, type the following information, then the libgomp package is not installed, needs to aquire from disc.

```
[root@linux akusb0-2.2.1-i386]# ./dinst
The 32bit support is missing. Please install the x86 compatibility packages required by your distribution and retry the installation. See the installation guide for more details. Aborting...
```

The problem is caused by lacking 32bit support libraries in Linux 64bit OS, installing 32bit support libraries is needed, which can be installed by loading system disc or acquiring from the system website.

Example: RedHat Linux 64

a) put system ios file into linux system, like in the example

```
rhel-server-6.4-x86_64-dvd.iso  put under /home/
```

b) Then configure yum source:

```
vim /etc/yum.repos.d/install_rhel6_4.repo
```

add the following information

```bash
[rhel6_4_base]
name=local rhel6_4  // (Do not change, it is the machine name)
baseurl=file:///mnt  // Change to the same
gpgcheck=0  // Change to the same
enable=1  // Change to the same
```

c) Load iso

```
mount /home/rhel-server-6.4-x86_64-dvd.iso /mnt/ -o loop
```

d) Check yum source

```
yum repolist
```
e) Check the needed packages

```
yum install libstdc++-devel.i686 glibc.i686 libgcc.i686 libstdc++.i686 glibc-devel.i686
```

Note: Change the libstdc++-devel.i686, etc. into corresponding libraries of ISO, the icon may be different due to different versions. Unzipping the iso files before checking the library names in the package.

```
[root@linux akususb-2.2.1-1386]# yum install libstdc++-devel.i686 glibc.i686 libgcc.i686 libstdc++.i686 glibc-devel.i686
Loaded plugins: product-id, refresh-packagekit, security, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
Resolving Dependencies
--> Running transaction check
---> Package glibc.i686 0:2.12-1.107.el6 will be installed
    --> Processing Dependency: libfreebl3.so(NSSRAWHASH_3.12.3) for package: glibc-2.12-1.107.el6.i686
    --> Processing Dependency: glibc-devel.i686 0:2.12-1.107.el6 will be installed
    --> Package libgcc.i686 0:4.4.7-3.el6 will be installed
    --> Package libstdc++.i686 0:4.4.7-3.el6 will be installed
    --> Package libstdc++-devel.i686 0:4.4.7-3.el6 will be installed
    --> Running transaction check
    --> Package glibc-headers-x86_64 0:2.12-1.107.el6 will be installed
    --> Processing Dependency: kernel-headers = 2.12-1.107 for package: glibc-headers-2.12-1.107.el6.x86_64
    --> Processing Dependency: kernel-headers for package: glibc-headers-2.12-1.107.el6.x86_64
    --> Package nss-softokn-freebl.i686 0:3.12.0-11.el6 will be installed
    --> Running transaction check
    --> Package kernel-headers-x86_64 0:2.6.32-358.el6 will be installed
    --> Finished Dependency Resolution

Dependencies Resolved

Packages Resolved

```

Package Arch Version Repository Size
--|-----|----------|---------|--------
| glibc | i686 | 2.12-1.107.el6 | rhel6 4 base | 4.3 M
| glibc-devel | i686 | 2.12-1.107.el6 | rhel6 4 base | 974 k
| libgcc | i686 | 4.4.7-3.el6 | rhel6 4 base | 122 k
| libstdc++ | i686 | 4.4.7-3.el6 | rhel6 4 base | 309 K
| libstdc++-devel | i686 | 4.4.7-3.el6 | rhel6 4 base | 1.6 M

Note: Change the libstdc++-devel.i686, etc. into corresponding libraries of ISO, the icon may be different due to different versions. Unzipping the iso files before checking the library names in the package.

Type y to install:
After installing 32bit library, then install license tool, the following picture shows the success of installation.

5. If using Oracle type datasource, then load environmental variable into the config files.
   For example: oracle11g configuration is shown as below:

   export JRE_HOME=/opt/SuperMapServer7C/support/jre
   export PATH=$JRE_HOME/bin:$PATH
   export SUPERMAP_ROOT=/opt/SuperMapServer7C/support
   export ORACLE_HOME=/u01/app/oracle/product/11.2.0/xe
   export ORACLE_LDB=ORACLE_LIB=/opt/SuperMapServer7C/support/objectsjava/bin:$LD_LIBRARY_PATH
   export NLS_LANG=ZHS16GBK

   ORACLE_HOME is the parent directory of oracle installation folder. In the example, network directory path is /u01/app/oracle/product/11.2.0/xe/network
   The environment variable of ORACLE_HOME is ORACLE_HOME = /u01/app/oracle/product/11.2.0/xe
   ORACLE_LDB is the lib folder of ORACLE_HOME
NLS LANG is the symbol dataset of oracle client
At last, add the environment variable of ORACLE_LIB into LD_LIBRARY_PATH