

Nokia Siemens Networks Security



Certificate Authority for Network Elements

Platform Installation Instructions

Version 3.1

Table of Content

1.	Introduction	3
1.1	Executive Summary.....	3
1.2	Scope of this document	3
2.	Outline of the Installation Procedure	4
3.	Step by Step Installation	5
3.1	Step 1: Installation Prerequisites	5
3.2	Step 2: Configure RAID controller and BIOS	5
3.2.1	Configuring the LSI RAID Controller	5
3.2.2	Changing the BIOS Boot order	7
3.3	Step 3: Fill the Planning Sheet and call the Macro	9
3.3.1	Editing the planning excel-sheet	9
3.3.2	Allowing Macro Execution.....	10
3.3.3	Customer Parameters to be entered into the Planning-Sheet	10
3.3.4	Running the Macros over the Input Planning-Data.....	12
3.4	Step 4: Begin actual installation.....	14
3.4.1	Choosing the server role to install.....	15
3.5	Step 5: Create PSKs to register FE servers.....	17
3.5.1	Creating PSK/Refnum in the BE.....	17
3.6	Step 6: Install the FE servers.....	19
3.6.1	Registration Security and reusing PSKs	19
3.7	Step 7: Enrolling the FE certificates in the BE.....	19
4.	Initialize Intrusion detection.....	24
5.	Disabling root login from SSH after installation	25
	Appendix A: Handling of Installation Media	26
	Appendix B: Description of the Installation Media	30
	Appendix C: Enable Excel Macro execution.....	31
	Appendix D: References	33

1. Introduction

1.1 Executive Summary

Nokia Siemens Networks provides a cost-effective and easy-to-use Public Key Infrastructure (PKI) solution for issuing and managing X.509v3 digital certificates. The lifecycle management of issued certificates can be automated and adapted to operator procedures.

The described Public Key Infrastructure is based on the INSTA DefSec Oy Certifier[®] software product. INSTA DefSec is part of the INSTA group (<http://www.insta.fi/en>) and has signed official and exclusive partnership with NSN, for joint development of a TelCo environment capable PKI.

In order to run the INSTA software in a TelCo operator environment, a highly available, scalable Platform is defined by NSN in agreement with INSTA. This architecture called HAPF2.0 defines following possible deployments:

- ☐ A single server without redundancy features, to use for basic testing.
- ☐ A full HA setup comprising several servers for scalable traffic separation.

1.2 Scope of this document

Following Chapters describe the installation procedure of the NSN INSTA HAPF 2.0 Platform. It is defined by NSN and used along with INSTA Certifier 5.x releases. A new HAPF version will be defined and maintained by NSN along upcoming INSTA Certifier releases.

The description covers the planning procedure for the IP and Linux HA environment and guidance through the automated installation procedure. After following the steps of this procedure, a fully functional, but not yet certificate signing HA PKI installation will be available. This “rapid setup procedure” minimizes the need to consistently edit Linux configuration files over several machines, thus significantly enhances setup and recovery time.

In addition to this document, kindly see the references to find further explanations.

2. Outline of the Installation Procedure

Following Flowchart outlines the installation Procedure for HAPF2.0. Each step marked in the graph is described in an own paragraph of Chapter 3 in this document.

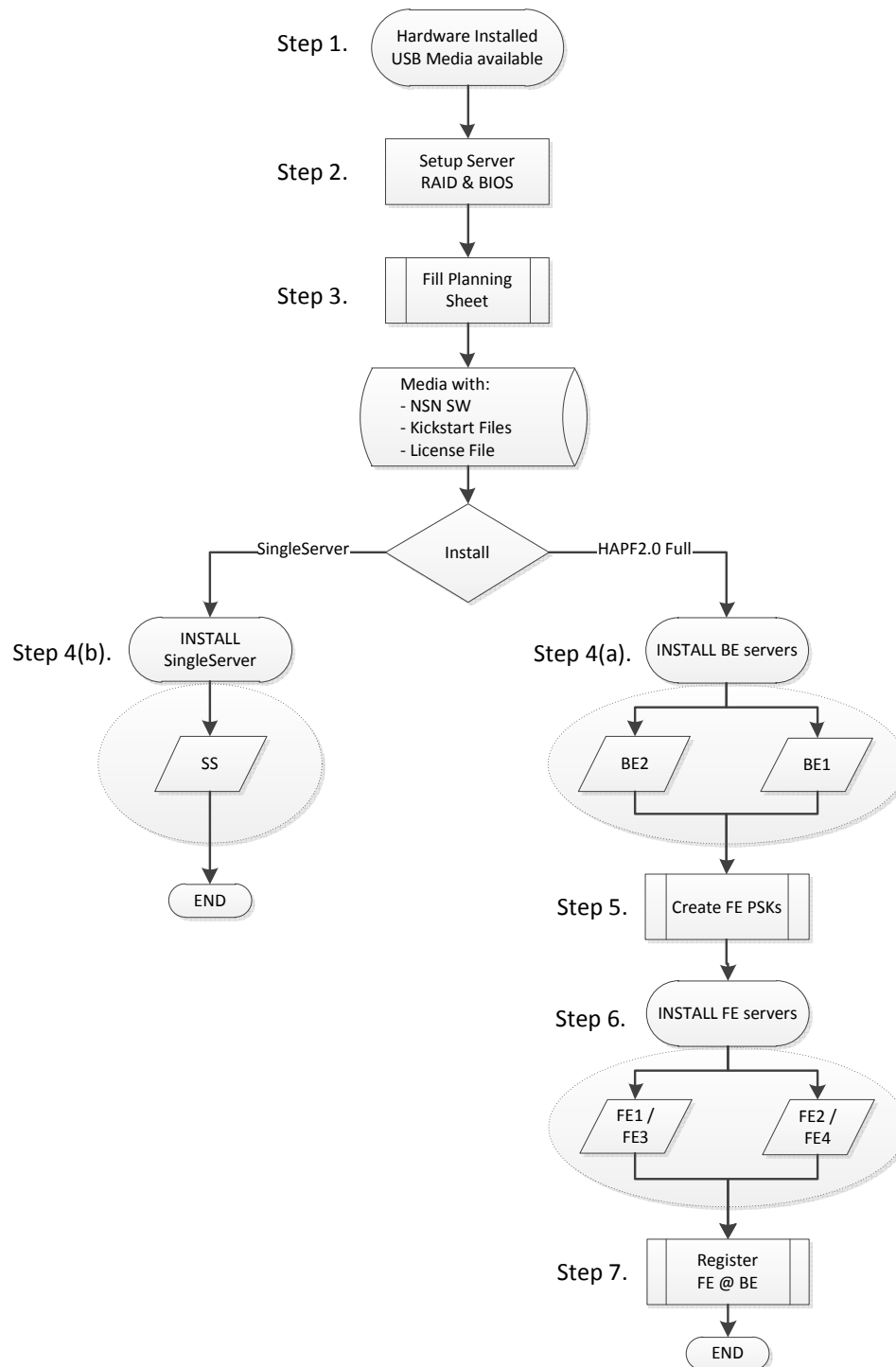


Figure 1: Steps to install HAPF2.0

3. Step by Step Installation

3.1 Step 1: Installation Prerequisites

Make sure you have the following basic conditions met or are prepared for it:

- 1) Your Hardware is SUN Oracle Netra 4270M2 as described in the architecture specification (/Ref 1). If you use anything else, contact internal T3 support first.
- 2) Server can be switched on: When normally operating, the Netra 4270 has an orange and a green LED on the upper left end on the chassis front both solid-on, not flashing. After power-up, the servers need about 90sec at least before first boot messages show on the screen. Test if you can use the keyboard.
- 3) To configure ILOM, you will need to establish a serial cable connection. An RJ to SUB-D Adapter is delivered with the servers as well as *normal* console cable will do it. Alternatively you can use a VGA monitor and a USB keyboard. Configure ILOM by at least assigning an IP address and default GW.
- 4) HAPF20 installation media is needed. For the physical installation, a USB Pendrive with Platform Software is the standard media. It can be delivered as binary image of a master-media, or finalized as a physically prepared drive.
- 5) Before beginning the Installation Process on the servers, make sure your Installation Media matches the “*Appendix B: Description of the Installation Media*” and that your Planning sheet data can be transferred to the USB media.
- 6) If no interruption to the regular boot sequence will be given, the Netra servers would boot a preinstalled default OS via GRUB. For HAPF2.0 setup, the boot-process will be interrupted and this OS will be overwritten while reconfiguring the LSI RAID Controller.

3.2 Step 2: Configure RAID controller and BIOS

When the Netra 4270 is powering on, the built-in LSI RAID controller must be configured for HAPF2.0 and the BIOS boot order must be changed. You start with configuring the LSI RAID because any changes made to the BIOS boot-order will be lost after exiting the LSI setup menu.

3.2.1 Configuring the LSI RAID Controller

In order to get into the LSI controller menu, press [CTRL]-C to interrupt the normal boot sequence. Once the BIOS received the command, you will see a message

Please Wait, invoking SAS configuration Utility



You should see the following screens:

```

LSI Corp Config Utility      v7.03.04.00 (2010.05.06)
Adapter List Global Properties
Adapter                     PCI PCI PCI PCI FW Revision Status Boot
                             Bus Dev Fnc Slot          Order
SGX-SAS6-INT-2             0D 00 00 04 5.00.17.00-IR Enabled 0

```

Esc = Exit Menu F1/Shift+F1 = Help
Alt+N = Global Properties +/- = Alter Boot Order Ins/Del = Alter Boot List

Figure 2: Screen 1 of RAID adapter configuration, select the only available controller using the keyboard

```

LSI Corp Config Utility      v7.03.04.00 (2010.05.06)
Adapter Properties -- SAS2008
Adapter                     SGX-SAS6-INT-2
PCI Slot                    04
PCI Address(Bus/Dev)        0D:00
MPT Firmware Revision       5.00.17.00-IR
SAS Address                  500605B0:035A2B50
NUDATA Version              05.02.00.16
Status                      Enabled
Boot Order                  0
Boot Support                 [Enabled BIOS & OS]
RAID Properties
SAS Topology
Advanced Adapter Properties

```

Esc = Exit Menu F1/Shift+F1 = Help
Enter = Select Item +/-/Enter = Change Item

Figure 3: Navigate to RAID properties menu and open it to get the following screen (next Figure)

Chose to create a “RAID-1” Volume and assign both disks in slots 4 and 5 to it. Accept that all data on those disks will be erased. At the end, you should see flowing screen:

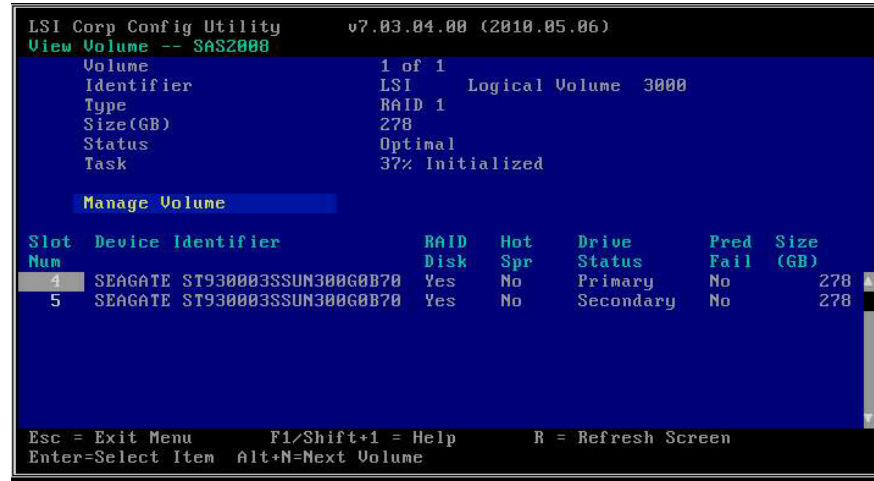


Figure 4: After assigning both drives into a RAID 1 configuration, above screen will indicate success

After configuring the RAID, the machine will reboot. You will have to interrupt this reboot cycle again, in order to change the BIOS boot order.

3.2.2 Changing the BIOS Boot order

After the RAID controller is configured during the previous step, the server will reboot. Interrupt this reboot again to change the BIOS boot order.

In order to be able to prioritize booting from USB, you must have the USB install media inserted during the boot process! Insert the USB media before booting into the BIOS configuration menu!

Press “f2” after you see the AMI BIOS screen right at the beginning of a boot cycle and wait until the Hardware is initialized far enough to bring you to the BIOS settings menu. From the Main, navigate into the Boot menu and choose option

“Boot Device Priority”

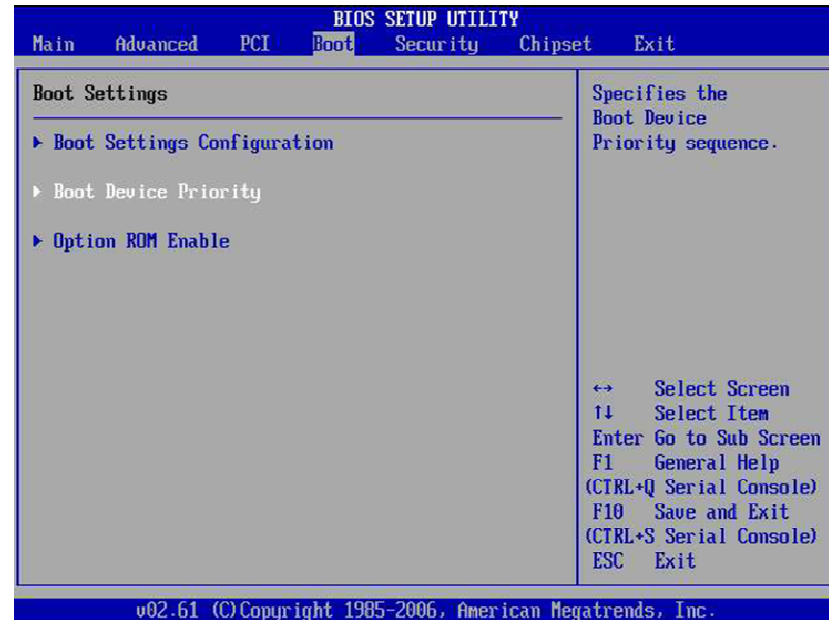


Figure 5: In the BIOS configuration menu, navigate to the Boot settings and choose “Boot Device Priority”

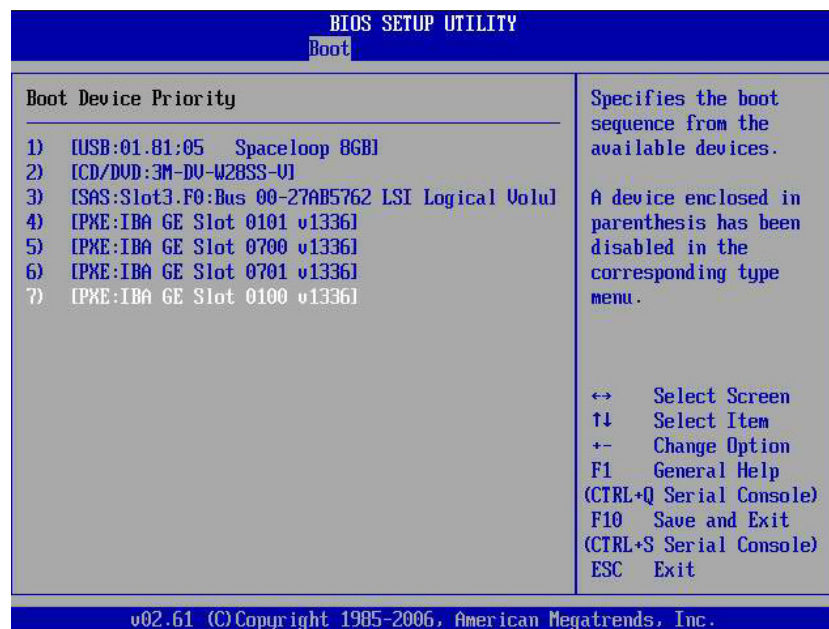


Figure 6: Only if USB media is really inserted, you will be able to prioritize it and move it to position 1

Change the priorities to be USB first, followed by DVD and SAS Harddisk. At the end the menu should display as above. Save the settings and exit.

Mind: If you use the ILOM java console and not a screen/keyboard directly connected, use the “keyboard” menu in the console to send an [F10] key!

3.3 Step 3: Fill the Planning Sheet and call the Macro

In order to run Insta in a scalable HA environment, all customer-specific parameters have to be configured in various Linux files throughout the different servers of the system. For example, Linux HA settings, IP addresses, hostnames, timezone etc ... This process can be quite error-prone and often leads also to non-standard installations that are later on difficult to support.

This should be encountered by a procedure called HAPF2.0 rapid-setup. All customer specific data is defined in a single excel sheet. This sheet should be filled, saved to the USB pendrive media and after data-validation all needed files will be created automatically to the USB install drive.

3.3.1 Editing the planning excel-sheet

On the USB installation media you can find an excel-file `inst_params.xlsm`. Open the file in a recent version of MS excel so that you can fill in the customer specific installation parameters.

	A	B	C	D	E	F	G	H	I
1		VALIDATE	CREATE !	SingleServer	fe1	fe2	be1	be2	fe3
2		CREATE_ROLE	no	yes	yes	yes	yes	no	no
3		TIMEZONE		America/Sao_Paulo	America/Sao_Paulo	America/Sao_Paulo	America/Sao_Paulo		
4		IPADDReth0		192.168.1.68	192.168.1.69	192.168.1.51	192.168.1.52		
5		NETMASKeth0		255.255.255.128	255.255.255.128	255.255.255.128	255.255.255.128		
6		HOSTNAME		pkir-fe1	pkir-fe2	pkir-be1	pkir-be2		
7		NAMESERVER		192.168.1.1	192.168.1.1	192.168.1.1	192.168.1.1		
8		DOMAIN		endofinternet.org	endofinternet.org	endofinternet.org	endofinternet.org		
9		DEFGW		192.168.1.1	192.168.1.1	192.168.1.1	192.168.1.1		
10		LOCALINTIP							
11		NTPTSERVER1		0.rhel.pool.nip.org	0.rhel.pool.nip.org	0.rhel.pool.nip.org	0.rhel.pool.nip.org		
12		NTPTSERVER2		1.rhel.pool.nip.org	1.rhel.pool.nip.org	1.rhel.pool.nip.org	1.rhel.pool.nip.org		
13		NTPTSERVER3							
14		NETGW1		169.254.101.129	169.254.101.129	169.254.101.126	169.254.101.128		
15		NET1		169.254.101.0	169.254.101.0	169.254.101.128	169.254.101.128		
16		NETMASK1		255.255.255.128	255.255.255.128	255.255.255.128	255.255.255.128		
17		NETGW2							
18		NET2							
19		NETMASK2							
20		NETGW3							
21		NET3							
22		NETMASK3							
23		NETGW4							
24		NET4							
25		NETMASK4							
26		NETGW5							
27		NET5							
28		NETMASK5							
29		HBIPADDR		169.254.254.1	169.254.254.2	169.254.254.5	169.254.254.6		
30		HBIPMASK		255.255.255.248	255.255.255.248	255.255.255.248	255.255.255.248		
31		HBMCADDR		226.94.2.1	226.94.2.1	226.94.1.1	226.94.1.1		
32		HBMCPORT		5401	5401	5401	5401		
33		VLANID		101	101	101	101		
34		VLANIPADDR		169.254.101.252	169.254.101.253	169.254.101.1	169.254.101.2		
35		VLANIPMASK		255.255.255.128	255.255.255.128	255.255.255.128	255.255.255.128		
36		VLANIPVirtual		169.254.101.254	169.254.101.254	169.254.101.3	169.254.101.3		
37		VLANID		200	200				

Figure 7: Planning Parameters sheet to be filled before installation, used to create `configure.sh` and `ks` files

The HAPF2.0 Architecture description has background information on the context of the parameter blocks of the sheet. More detailed information on each single parameter (lines) is found inside the sheet itself.

For further descriptions here in this document, open the sheet and mind the blocks of parameters that have been linked together by Column A.

3.3.2 Allowing Macro Execution

You must enable execution of Macros in order to allow the sheet to work. If you do not see a shortcut button or warning for that during opening the file, follow the instructions of "*Appendix C: Enable Excel Macro execution*".

3.3.3 Customer Parameters to be entered into the Planning-Sheet

Parameters are seen by HAPF2.0 under the names given to them in Column B, rows 3 downwards. Each Parameter has an "Index" (Row 1) or a "Role", for example, the parameter HOSTNAME[fe2] in cell E6 defines the hostname for the server with role fe2 in the full-HA setup.

Before any Parameter Values can be filled into an empty sheet, the roles must be enabled by setting the dropbox for "CREATE_ROLE" in row 2 to "yes". You can only enter parameters for roles that are enabled. Once a role is enabled, a kickstart file will be created for it.

Some parameters, like e.g. the UDP port for pacemaker Multicast in row 32, can never be changed.

3.3.3.1 Additional Static Routes

Since BE and FE pairs are located in different IP networks and/or VLANs, additional static routes can be defined on the hosts, pointing to the corresponding gateways. "Additional" means that the default-GW on eth0 will in many setups not be able to route traffic between FE and BE, or from the production Network to the FE. Therefore, up to 5 additional routing entries can be made in row-Blocks B1-B5. If more than 5 routes are needed, they can be added manually, for example in /etc/sysconfig/static-routes on each server after the auto-installation completed.

At least one static route should usually be needed on each pair of FE/BE, a route for the HA TLS-connection between FE and BE.

3.3.3.2 VLANs and HA IP addresses

On systems with HA, an Ethernet trunk is defined using eth1 and eth2. This trunk uses bonding-mode 1 (Active Standby) and requires the L2 device on which the servers are connected, that it can recognize 802.1Q VLAN tags in the Ethernet frames coming from the servers. This is the only requirement, since the bonding mode 1 has only one active interface at a time and no STP or link-aggregation protocols on L2 are needed.

In Parameter Blocks D1-D6, it is possible to define 6 additional HA-addresses/VLANs per server pair. For example, in Block D1 row 33-36, a VLANID is defined (same on both servers of course) and a physical IP (VLAN1IPADDR) for each server. On top, a virtual IP (VLAN1IPvirtual) is defined and will be handled by both servers via Linux HA.

3.3.3.3 Disksizes

In Block E, line 59 allows to choose from 2 predefined disksizes “normal” or “minimum”. Size “normal” is based on a RAID1+0 config with 2 300 GB disks as described later in the HW setup (the default disk configuration). Filesystem sizes as given under this option can be used for regular operation of production servers.

Size “minimum” is based on the absolute minimum requirement for installation of the SW but it is not suitable for operation in the long run. Neither in testbed nor in other environments. It can be used – for example – to create small virtual appliances with a SingleServer setup and once the vm is running, additional virtual disks should be added! In the long run, it is not recommendable to operate a testbed with less than 15GB disk space in the certifier filesystem.

When you set the option in Column 59 to “customized”, the filesystem sizes can be changed, where row 67 shows the summarized disk space of all filesystems. In order to not get stuck during the installation, remember, that your RAID will require an overhead portion of the “delivered” disk size, so that in total you may end up with approx 20-30% less disk space than the total size. In the “normal” setup for example, we only assign 240GB of the 300GB available per (un-RAIDed) disk.

For the automated installation, the system needs to know 2 disk device files that could theoretically differ from the sheet default:

INITIALDSKDEVPREFIX describes the linux device file under /dev, that is used for access to the disk on which the PKI will be installed. It needs to be the file for the entire disk, not for a single partition only.

PENDRVUSBDEVICE describes the full path to the device file of the USB Pendrive from which we will start the installation. Meaning, after the Anaconda linux system installer has started, it will see the USB pendrive under this device file.

3.3.3.4 SNMP integration

In Block F the Parameters for SNMP Integration can be defined. How this should be done is depending on whether a generic SNMP trap-sender functionality is required or a plug&play integration to NSN NetAct. In every case – even if no SNMP Alarm information should be sent anywhere – a “rudimentary, local” SNMP configuration will be made by the Platform. Rudimentary local config does not send traps to an NMS, but is needed in order to supervise OS resources (mainly to restart the most vital processes like ntpd or sshd in case they crash).

If the SNMPENA parameter in row 68 is set to “no”, only local configuration for SNMP will be made and no Fault-notifications will be sent to a remote system. If SNMPENA has been set to “yes”, a trap receiver must be specified and it can receive – depending on the other parameters – only SNMPv2c, SNMPv3 or both notifications and/or informs. Polling will always be (theoretically) possible from the local shell but is blocked on the iptables firewall if SNMPENA is set to “no”.



3.3.4 Running the Macros over the Input Planning-Data

To access the information of the excel sheet during HAPF2.0 installation, a tool called “mkks61.exe” will extract all relevant customer parameters from the saved and edited sheet and generate a set of customer specific installation files¹. This process happens automatically in 2 steps:

First the input data must be validated, to ensure that the automated installation will not get stuck due to incomplete or formally wrong data entered in the planning phase. This step is executed by the “Validate” Macro.

In a second step, the validated data is used to complete the install media. This means, kickstart files will be created from the excel content and saved to the USB media. Further, the license file will be copied into place so that it can be handled during the HAPF2.0 installation scripts.

3.3.4.1 Data Validation Macro

After editing the sheet, press the button “VALIDATE” in the upper left corner. This will start a procedure to validate your input. The procedure will stop with an error message and throw you into the first invalid cell it finds. Or it will just go straight through and you can then create the kickstart files for installation.

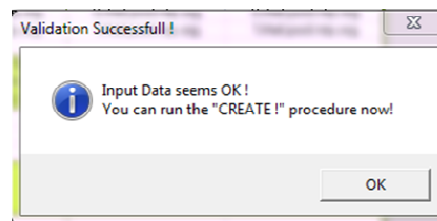


Figure 8: Dialogue Box indicating successful data-validation

MIND: Data validation mainly checks that no cells have been accidentally left empty or that no IP addresses are mistyped! However, it does not provide a full-scale plausibility test. Hence, if you enter e.g. 192.168.1.1 as a netmask, validation would accept this! Check your data carefully before starting the installation!

¹ i.e. one ANACONDA kickstart file per active server role and one common definition “configure.sh” where all parameters are set to environment variables for bash. HAPF2.0 scripts will read those variables while dynamically creating the configuration files during install.

3.3.4.2 Finalising the USB drive with Kickstart & License File

After successful validation is done, press “CREATE” in order to create the needed kickstart files (see also Appendix B). The kickstart files should be created on the USB drive/directory that will be used for the installation. Choose the USB-pendrive in the file-picker dialogue.

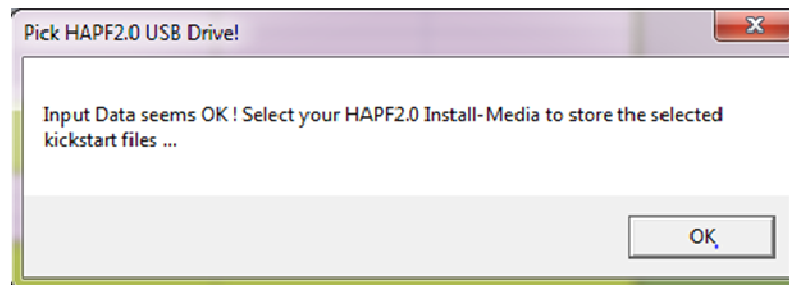


Figure 9: Dialogue Box before the “File-Picker” dialogue to choose the drive/directory where the kickstart files shall be stored. Use the USB Installation Media or otherwise you will see an error message.

Mind, that you HAVE TO chose the USB pendrive and no other directory for the kickstart creation to work:

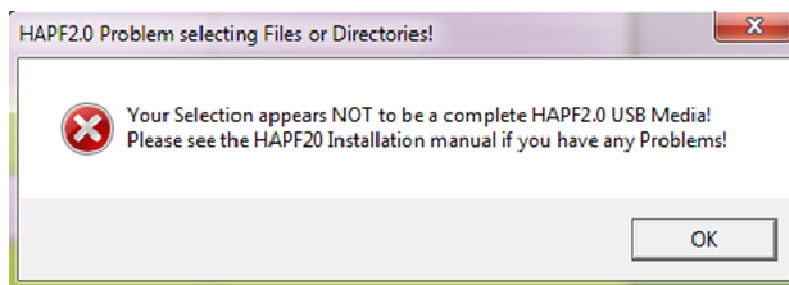


Figure 10: Error message in case a different directory than the USB install media is selected to create the kickstart files. The Macro will not continue in this case, the “Create” procedure has to be called again..

The Macro will also copy the license file to the proper location on the pendrive. Direct the program to your license file received from NSN so it can copy it to the USB.

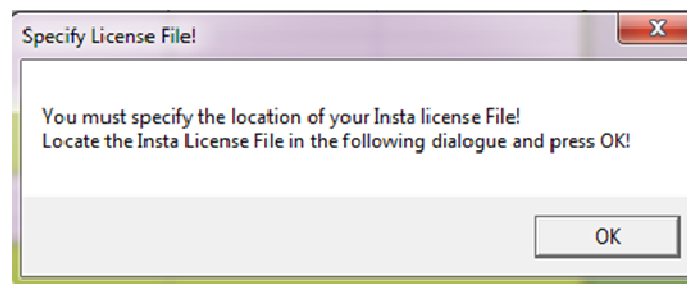


Figure 11: Before the kickstart files are created, a dialogue will ask to show the location of the license file.

After above dialogue is accepted, the macro is finished and the install media ready!

3.3.4.3 Key data created by the Macro

After the “CREATE” Macro in excel is finished, the following data has been added to the USB installation media:

- Kickstart files ks-`{ROLE}`.cfg for each role that has been marked with CREATE_ROLE “yes”
- A file configure.sh containing all parameters and indexes extracted from the excels sheet/
- The Insta license file for the installation, copied into the USB drive directory `~/insta/license_data.lic`
- A new file “syslinux.cf” is created on the pendrive, reflecting the disk device files of your installations (without that, you wont be able to boot the install media!)
- A file “stampfile.txt” is created on the installation media. It contains key data to identify your installation and match the parameter sheet against the USB media.

The stampfile contains 3 variables that characterize each installation:

- NOW is a 12-digit number representing the date and time when the “fileset” for the installation was created. A “fileset” is the combination of kickstart and config files created by the Macro, the license file plus the files that were already contained on the installation media. The “NOW” timestamp is used throughout the entire installation to name logfiles etc ... In case multiple installations of PKI clusters are done, it helps to identify them and match the files belonging to it.
- MEDSET is a string describing the kind of media that is used. For HAPF2.0 this should be PF20I50RH62-1.

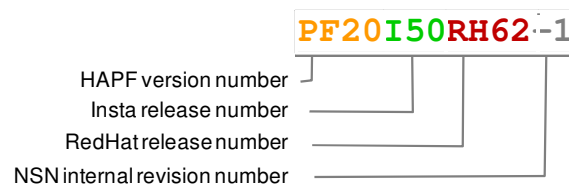


Figure 12: MEDSET release string in the stampfile.txt

- GENVERS is the version of the perl source-file that created all the other customer specific files. The perl source is normally contained on the pendrive as well and can be called if no MS-excel is available to (re)generate the files e.g. on a linux platform.

3.4 Step 4: Begin actual installation

After the server HW (RAID and BIOS) is prepared and the customer specific kickstart-files and configure.sh are created/copied to the USB installation media, the actual installation can start. Insert the USB media into a USB port at the rear of the server and reboot.



3.4.1 Choosing the server role to install

When booting from the USB pendrive, a boot menu will display the installation options to choose from. Install servers the following order, each one by choosing its role when booting from the media:

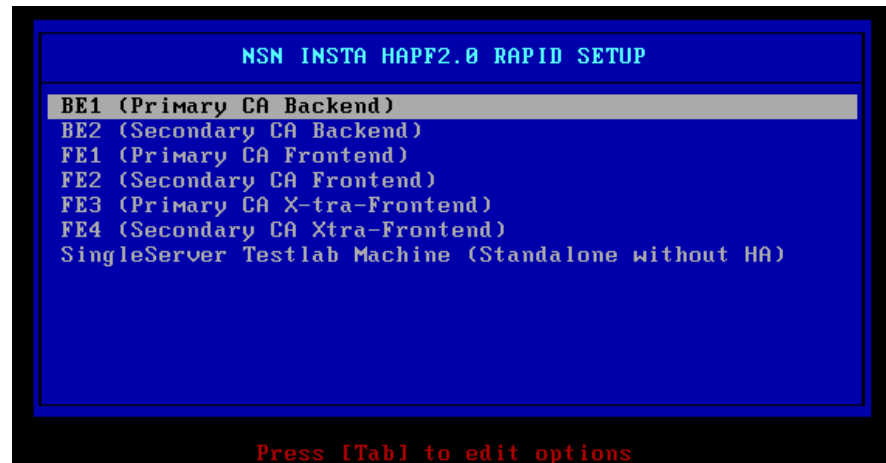


Figure 13: Boot menu screen as defined in `system.cfg` on the USB install media. Choose the role to install.

Installing SingleServer role is straightforward: Just select the role and the system will be installed and configured automatically. If a full-HA setup is to be implemented, install first BE1 and then BE2.

Important Notice

Mind, that you should use the SAME USB drive to install BE1 and BE2: Authentication data for corosync, SSH-keys and eventually other data are copied from the primary to the secondary server via this USB drive. Same applies for the other HA pairs FE1-FE2 and FE3-FE4. If you want to use different media and not copy Authentication data via USB, you must set the option "REMOTE_INSTALL" to "yes" and provide a connection on eth0 between the HA peers of one role!

The Macros will show you only those options for installation which you have selected in the sheet! A screen as in above figure would normally miss several options.

Above menu is defined in the `syslinux.cfg` file, which is also macro-created. If you specified a wrong device file for the USB Installmedia parameter (name in excel is PENDRVUSBDEVICE), then you will not be able to boot any of the options above.

If you selected a wrong value of the device file that Linux installer Anaconda will see as local HDD in the server (parameter name in excel is INITIALDSKDEVPREFIX), then you can boot the above options. But you will get stuck after Anaconda has just started because the system will not find the kickstart files.

Independent from the server role, installation always happens in 2 Phases:

- Phase 1 is the regular Linux anaconda installation. At the end of this phase, a few commands/scripts are executed to set up host-security and networking (anaconda %post scripts). After the Phase 1 is finished, anaconda will ask you to reboot. (See Figure 14 below).
- Phase 2 is executed after the reboot. The %post scripts have replaced the original `/etc/rc.local` script with a customized version that completes the auto-configuration by setting up other subsystems, installing insta and completing the HA setup.

Normally installation of one server takes < 10 minutes and can slightly vary only depending on the read-speed of the used USB drive (which is minimal). Between Phase 1 and Phase 2, reboot the servers.

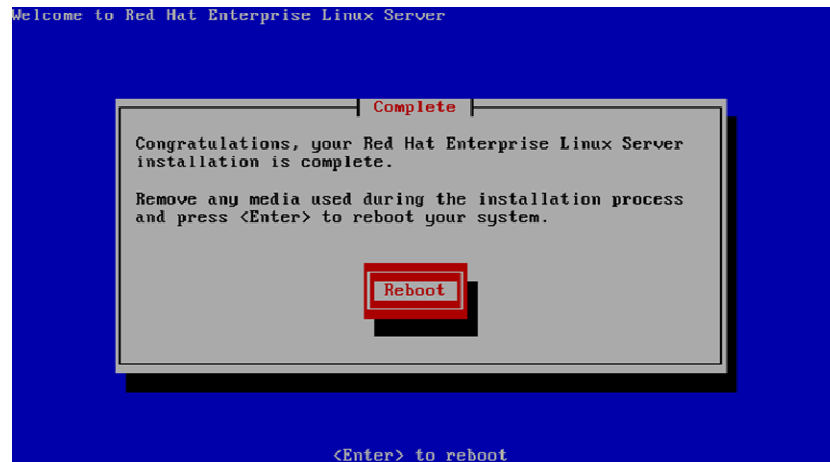


Figure 14: After successful Anaconda installation of linux, you will see this screen and can reboot.

During most part of Phase 2, the servers can already be reached via SSH on their OAM IP address. A logfile in `/tmp/Installmedia/instlog*.log` will show the progress of the installation, if it is not anyway displayed on the console screen.

Most time during the installation Phase-2 is used to collect entropy data, either needed by Insta or to create corosync authentication data. The process of entropy collection is significantly speeded up by creating i/o on the system. For this purpose, the steps of entropy collection are started with a `"dd if=/dev/sda of=/dev/null"` in the background, just to create i/o. However, if the CPU power and I/O capacity of some custom hardware is not sufficient, the "acceleration effect" might be small or even reverse! If your installation takes longer than 10 minutes, you should have an eye on this fact!

After the reboot is completed and the login prompt shown on the console, the BE pair is operational and the Certifier Admin service can be reached on port 8083. The default iptables policy allows that 8083 can be accessed on any IP address. Later tuning of the FW policy should limit this according to the operator security policy!

3.5 Step 5: Create PSKs to register FE servers

FE servers must be defined inside certifier BE with a PSK/Refnum. This combination is used to enroll X.509 certificates of the FEs for the TLS connection between FE and BE. To do so, 2 operations are needed:

1. After the BE1 and BE2 are installed, the certifier CA/PKI must be started on the BEs and the FE server Objects with PSK/Refnum must be created (This paragraph)
2. After the FE servers are installed, the first root-login after Phase 2 will jump into a “jail”-dialogue that can’t be left before the FE is successfully registered at the BE (described in Step 7).

Follow the instructions below to create the PSKs in the BEs, then do the actual FE installation in Step 6 and finally register the FE to the BE as described in Step 7.

3.5.1 Creating PSK/Refnum in the BE

Follow those steps to create the PSK/Refnum in the certifier BEs:

1. Log in to the certifier GUI using a web browser. The default user / password after installation is “admin” / “admin”, your browser must connect to port 8083 of the certifier OAM address (on eth0) or – later – to the HA-IP that you have eventually assigned to the “Administration Service”.
2. Go to section “servers” (left side, main menu), then click the “Add New Server” button and you will get to the “Create New Server Entity” Dialogue.

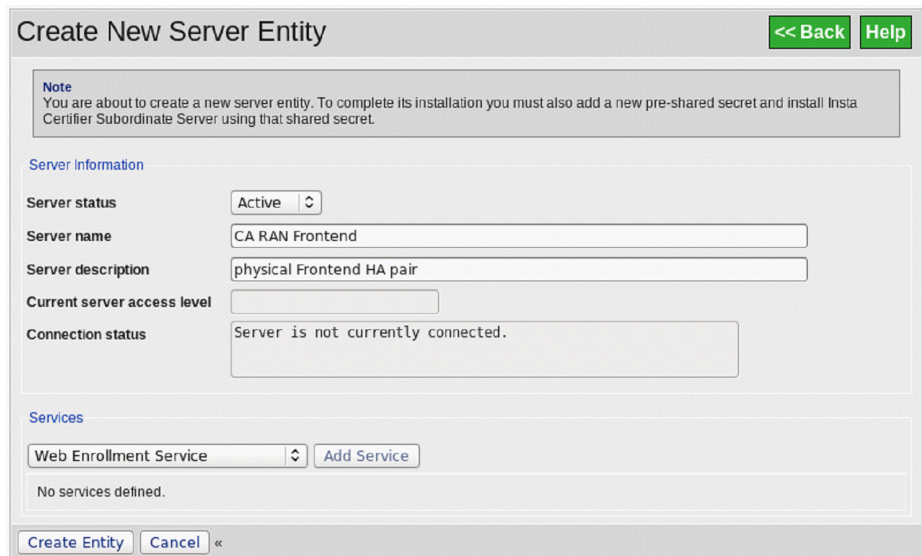


Figure 15: GUI section to add a new server to the INSTA PKI topology, needed before PSK generation

3. Enter a name and description for each Frontend server you are going to use! Mind, that each FE server will have an own certificate with the server’s

hostname used as Subject Name. Therefore, each server must be created as an own entity!

4. After pressing “Create Entity” you are thrown into the “Edit Server Entity” page. Here you go to the “Pre-Shared keys” section at the end and press “Add PSK”. This will create a PSK and connect it to the “End Entity” for which a certificate request is now expected. Press “Edit PSK” to change the PSK parameters and maybe even the PSK itself.

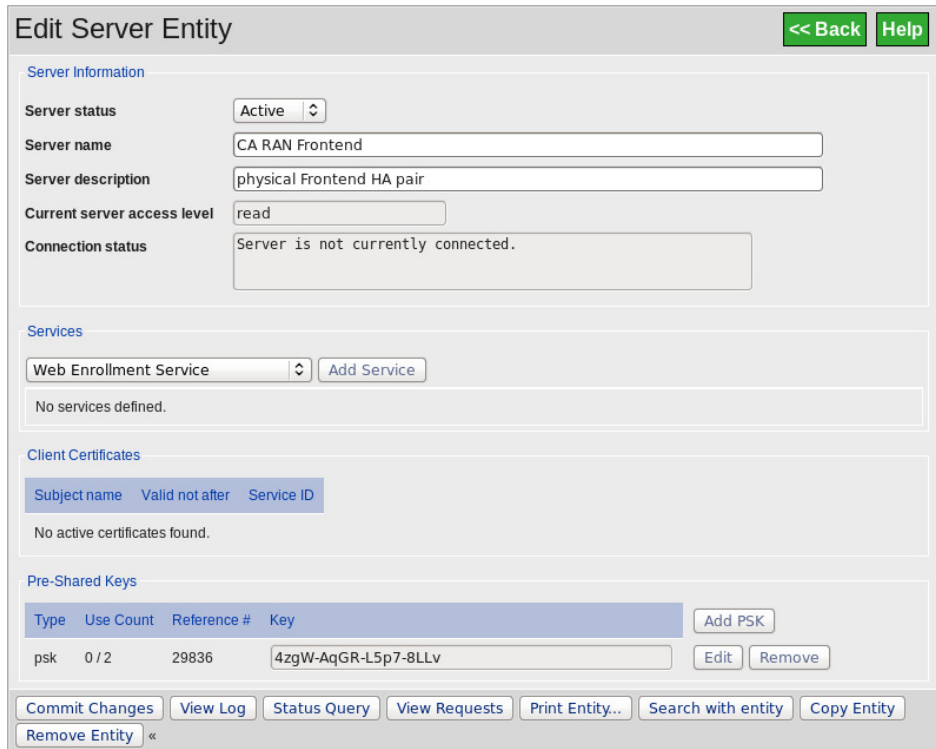


Figure 16: After the new server (i.e. FE pair) is created, a key must be added with “Add PSK” resulting as above

5. In the “Pre-Shared Key” dialogue window (not depicted here), you can also change the “use count” of the key to allow multiple retries. This would be a security risk for that particular situation and should be avoided - every server will enroll only once! You CAN also edit the key itself and change the rather complicated default keys to a simple word-phrase like “mykey”. After you changed the key parameters, press “Commit Changes” and you will be thrown back into the “Edit Server Entity” page where you shall see latest modifications to the PSK now.
6. Repeat the step of entity creation until you have one entity with active PSK for each server. Note down the PSK that is valid for each of the servers. You can not re-use PSKs among different entities. If you have a 6-server setup with 4 FEs, you will need to create 4 FE server entities!

3.6 Step 6: Install the FE servers

At this point, Step 6 is not different from the BE server installation described in Step 4. Naturally, a different role is chosen now and again the SAME USB drive must be used for both servers of the same HA pair.

In case of a 6-server setup, it is possible to install both FE pairs simultaneously to save time – a (one!) copy of the USB drive is then needed. It is also possible, to install FE's before the PSKs are generated. However, it will not be possible to log in on FEs console or via SSH until the registration of the FE has been concluded successfully. This is a security measure!

3.6.1 Registration Security and reusing PSKs

Later fact because in connection with the use-count "1" of the PSK generated in Step 5, it is ensured, that only a legitimate Frontend machine can connect or that any other abuse would at least be detected because the PSK has been used up! It is therefore recommended, to leave the use-count of the PSK at "1".

If a Problem has occurred after the PSK has been used in below Step 7, and the Frontend must register again, a new certificate will be issued (same Subject name CN="{hostname of the FE}" but new serial number and keys). To do the re-enrollment, the FE entity has to be deleted in Insta BE, then created again with a (new or same as previous) PSK and enrolled again. This should normally not be necessary.

Enrollment certificates for the internal communication between Insta FE and BE servers are signed by the certifier's internal TLS CA. Communication happens via port 7001 via a TLS secured proprietary, internal interface.

3.7 Step 7: Enrolling the FE certificates in the BE

After following the instructions from the previous paragraph, the FEs are now installed and a PSKS on the BE have been successfully created.

When the Anaconda phase has finished and the FE machines are completely rebooted, certifier processes are still not started and the insta installation is not complete on the FEs. In order to get the FE functional, you have to log in as root and complete the setup by supplying the PSK for the particular FE and the IP address of the BE service listening to port 7001. This will trigger the enrollment via ssh-ca-setup. Follow the operations as described below to complete this step:

- 1) Log in to the FE server as root. No other login name will be accepted. You are directly brought to the dialogue that finishes the enrollment:

```
YOU ARE ABOUT TO FINISH SETUP OF THE INSTA CA FE
=====
This procedure will finalise the Frontend configuration
by registering to the Backend engine. This process requires
that you have the following information at hand:
- IP address and TCP port of the BE to connect to
- PSK generated by the BE in the GUI-section "Servers"
THE BACKEND ENGINE MUST BE RUNNING WHEN YOU START THIS PROCEDURE
If you do not have all needed information at hand, then
type "logoff" and come back after clarification!
type: [Any key to continue] || [logoff]          [>] :
```

- 2) When you press any key in the above dialogue, the procedure will continue and you are expected to have the IP address of the BE service and the PSK for this particular FE at hand. If this is not the case, you can type "logoff" and press enter to leave the dialogue.



- 3) If you continue, the procedure will look if there has been a “LOCALINTIP” parameter configured for the BE pair that belongs to your FE. Internal communication as well as enrollment should go via this address. If there is nothing (correctly) configured OR if that IP can currently not be reached with “ping” – e.g. because an external firewall blocks either ICMP or all communication between FE and BE, the procedure will suggest other IP addresses which it can reach. However, successful enrollment requires that you reach the BE!

```

YOU ARE ABOUT TO FINISH SETUP OF THE INSTA CA FE
=====
This procedure will finalise the Frontend configuration
by registering to the Backend engine. This process requires
that you have the following information at hand:
- IP address and TCP port of the BE to connect to
- PSK generated by the BE in the GUI-section "Servers"
THE BACKEND ENGINE MUST BE RUNNING WHEN YOU START THIS PROCEDURE
If you do not have all needed information at hand, then
type "logoff" and come back after clarification!
type: [Any key to continue] || [logoff]          [>] :

looking for a BE IP address in your configuration ...OK!
Enter the PSK you generated in the Backend
(type only the key, no refnum)                    [>] :testkey
Calling ssh-ca-setup with PSK testkey-fe2 on BE  tcp://192.168.1.81:7001
/

or withdraw input and start again, or logoff
type: [Any key to continue] || [other] || [logoff] [>] :

```

- 4) Confirm the IP address and PSK suggested by the procedure or just type “other”, then press “[RETURN]” and enter a new IP address and a different PSK. You can use this either if you mistyped the PSK or if the IP address suggested by the system is no longer valid.

- 5) After you accepted an IP address and PSK for a go, the system will continue and start the customized “ssh-ca-setup” script. If the enrollment is successful, you will see a log message on the screen that confirms to have received a TLS certificate and at the very end a line will prove that connection to the BE has been made.

```

Jan 23 12:02:20 pki-fe1 etc-profile rc-stop of corosync requested - script returned
"0": "corosync is stopped"
Jan 23 12:02:20 pki-fe1 etc-profile sourcing external script : "-rwxr-x--- 1
certfierdaemon 25386 Jan 22 19:39 /usr/local/certifsub/ssh-ca-setup-nsn-certifsub.sh"
Jan 23 12:02:20 pki-fe1 etc-profile #-----
-----

Jan 23 12:02:20 pki-fe1 etc-profile entering Insta setup routine ssh-ca-setup for FE
role
Jan 23 12:02:20 pki-fe1 config_insta.sh-ssh-ca-setup using configure index 1, will
define tcp://192.168.1.81:7001/

.... some more log

Insta Certifier subordinate server installation complete.
This Certifier server will start automatically at system boot (runlevel 2).

Please run /usr/local/certifsub/ssh-ca-start to start Certifier Server now.
Jan 23 12:02:21 pki-fe1 config_insta.sh-ssh-ca-setup leaving Insta setup routine ssh-
ca-setup for FE role
Jan 23 12:02:21 pki-fe1 config_insta.sh-ssh-ca-setup #-----
-----

Jan 23 12:02:21 pki-fe1 etc-profile returned with 0 from external script : "-rwxr-x--
- 1 certfier daemon 25386 Jan 22 19:39 /usr/local/certifsub/ssh-ca-setup-nsn-certifsub
.sh"
Jan 23 12:02:24 pki-fe1 etc-profile copied init script into place: -rwxr-xr-x 1 root
root 10236 Jan 23 12:02 /etc/init.d/certifsub
Certifier seems to be cleanly stoped for role certifsub on this host
Starting Insta certifsub Version 5.0.0 (build 62) [OK]
Jan 23 12:02:30 pki-fe1 etc-profile test-started certifsub and received exit 0 from
start script
...waiting for BE connection or timeout
CONNECTION to BE tcp://192.168.1.81:7001/ is up: tcp 0 0 192.168.1.91:51039
192.168.1.81:7001 ESTABLISHED 28522/ssh-ca-server
Stopping Insta certifsub Version 5.0.0 (build 62) [OK]
Jan 23 12:02:42 pki-fe1 etc-profile in future you must use /etc/init.d/corosync to
control certifier

CERTIFIER IS HALTED NOW, START COROSYNC TO ENABLE HA RESOURCES!!
Jan 23 12:02:42 pki-fe1 etc-profile restored original file /etc/profile: "-rw-r--r--
1 root root 1793 Jan 23 12:02 /etc/profile"
Jan 23 12:02:42 pki-fe1 etc-profile disabled init-start of certifsub : "certifsub
0:off 1:off 2:off 3:off 4:off 5:off 6:off"
Jan 23 12:02:44 pki-fe1 etc-profile rc-(re)start of corosync requested - script
returned "0": "corosync (pid 28630) is running..."
Connection to 192.168.1.91 closed.
[12:02:41][root@pki-adm ~]#

```


- 6) After successful enrollment, the HA software (corosync/pacemaker) will be restarted and the FE will become operational when the FE-HA-pair is completely operational.
- 7) That the “FE-HA-pair is completely operational” means, that after completing the procedure on FE1/FE3 only, your cluster pair is still incomplete and will run without FE2/FE4. After the procedure has been completed for the first FE in a pair, repeat it for the HA peer in order to get the SW operational.

Both FEs in a HA pair have the “/usr/local/certifsub” filesystem mounted, but HA will allow only one FE to run active. However, it is possible, that you use the init-script to start Insta also on the standby peer and the connection will become active. This has been allowed for convenience in maintenance and is at no point a good idea unless you closely monitor what is happening, or have the clustering functions manually disabled! The clustering Software gets into an undefined state if it is active while you simultaneously run Insta FE functions on both peers!

4. Initialize Intrusion detection

HAPF2.0 comes with a preconfigured setup of AIDE (Advanced Intrusion Detection Environment) that is contained in the default RedHat Enterprise Linux. The system will run cron-jobs that provide reports about changed files and they should be inspected carefully on a weekly basis just as all other system logs. They can be found in the directory `/var/log/aide`.

In order for AIDE to function properly, it must be initialized to build a database of the system baseline. This process is also automated, but must be triggered manually as soon as a certain static level is reached after the installation. It is on the owner of the system to define, when this point is reached. The system will come up with “reminders” as long as AIDE initialization has not been done. Those reminders can be disabled by removing the last line from `/root/.bashrc`:

```
/root/.do_aide_anoy
```

However, it is the purpose of the reminder, that the initialization should be executed and after this has been done, the reminders will disappear naturally without further action required. In order to initialize AIDE, the system normally requires about 2 minutes:

```
PLEASE INIT AIDE ADVANCED INTRUSION DETECTION ENVIRONMENT
Run the following command to complete AIDE initialisation:
    "/root/init_aide.sh --interactive"
!!!!!!!!!!!!!!
[12:17:15][root@pki-be2 var]# /root/init_aide.sh --interactive
Your system is prepared to run AIDE advanced intrusion detection
environment. AIDE needs to re-initialize to reflect changes done during
installation phases. After its first database was built. You must allow this
script to continue in order to use AIDE without obtaining a high number of
false positives.
The initialization will take approximately 2-4 minutes. It is executed only
this one time and requires your patience!!!
You can also init aide at any time by running "/root/init_aide.sh --
interactive"
MAKE SURE THAT ALL UNNECESSARY FILESYSTEMS (USB, /boot/efi, ...) ARE
UNMOUNTED!
=====
type [OK] to start (re)-initialising AIDE now
type [NO] to skip init (asks again on next login)

Starting to (re)-initialize at 12:17:41, please be patient
done at 12:18:17, AIDE database re-initialized!
[12:18:17][root@pki-be2 var]#
```

If you allow the above dialogue to continue by typing “OK” and pressing [RETURN], the initialization will complete and reports will be generated.

5. Disabling root login from SSH after installation

In order to allow a smooth installation process, root-login via SSH is allowed with password-interactive authentication. This setting might not be compliant for most operator security policies and is in every case recommended to be disabled after the installation is finished.

In `/etc/ssh/sshd_config`, go to the end of the file to disable root-logins and consider also to disable password interactive login in general.

```
# We recommend you comment the following lines
# to enforce keybased authentication
Match User nsn
    PasswordAuthentication yes
Match User root
    PasswordAuthentication yes
    PermitRootLogin yes
#
```

To disable root login, comment the 3 last lines with the “Match User root” entry ! This will not allow at all an initial login for user root, but still allows any user to do “su – “ to become root also in remote sessions.

To disable Password Authentication in general, comment also the “Match User nsn” entry.

In the upper part of the file (not shown in the figure), all logins via password interactive authentication are disabled, and the “Match User” entries do only set exceptions. For example, user “certfier” that is also generated during system setup will not be able to log directly in because password-login is disabled and no ssh-keys are (yet) generated! The same applies to all users that might be added later via regular “adduser” commands on the command line.

To disable this setup for all users, change the line 39 in the `sshd_config` to

```
PasswordAuthentication yes
```

According to good security practice, this is discouraged at this point and use of SSH-keys and blocking of password login for all users is highly recommended! Proper management of keys and users are left to further discussion with the operator if needed.

After changing the SSHD-config. Changes become active by issuing the command

```
service sshd restart
```

Appendix A: Handling of Installation Media

As install media for HA setup, a USB pendrive is required. Servers will boot from this pendrive, hence the drive partitioning must be prepared accordingly.

In order to allow easy delivery of this boot-media, NSN CSI Sec will usually provide a binary image file instead of a physical USB drive. This file needs to be “dumped” to a physical USB pendrive media which can afterwards directly be used for installation.

This Appendix describes, how the Pendrive can be created from a binary image file that NSN will provide. Using the binary image as described below will not require to use fdisk and apply manual changes to the USB drive’s partitioning.

However, in order to verify correct outcome of the media creation and to encounter common issues around such pendrive-based installations, it seems worth a few introductory comments:

- ☐ It is clear that you should use a quality pendrive, the minimum size is 2GB, larger drives up to 16GB have been tested and the size was no problem. Most tests have been conducted using different types of “SanDisk” Cruzer USB pendrives.
- ☐ The drive will be partitioned! You do this manually with fdisk by creating ONE partition on the drive. Some exotic drives have 2 partitions and you can not change that using fdisk. This no problem, if you specify the correct device file under which RH linux sees the main partition!
- ☐ The partition must be a “primary” partition. The boot flag must be set
- ☐ When the drive is partitioned, Linux will always see the device file of a partition! Like /dev/sdb1. If the drive is NOT partitioned and you only formatted it, then you will have a working pendrive but it is seen with the device file of the entire drive (like /dev/sdb) and not as a partitioned drive! This is not OK! You need a drive with one partition!
- ☐ Format the partition with a “vfat” (under linux) oder “fat32” (under windows) filesystem! Do not use NTFS, ext3 or other advanced filesystems !
- ☐ In order to create a bootable USB pendrive from the binary file received through your local NSN CT, follow the steps described in the following pages, for MS-Windows or Linux OS, respectively.

Important Notice

Mind, that the verification of SHA checksums of the delivery media (i.e. bin-file) as described is a necessary part of a “secure delivery and deployment” process for the operator CA/PKI! Because trust is only enabled directly among NSN and the customer, while any transport path that is not direct hand-to-hand between those parties, as e.g. email, ftp transfers etc, must be verified via out of band means!

The checksum should therefore be taken directly from the person who created the archive (e.g. PF10Insta44.bin.gz file) and compared with the checksum that can be reproduced by the archive used at the customer to create the install media. It does of course not make much sense, to verify a checksum that was received on the same channel as the media file itself, as this procedure would only unveil damages to the bin file, but not any malicious modifications!

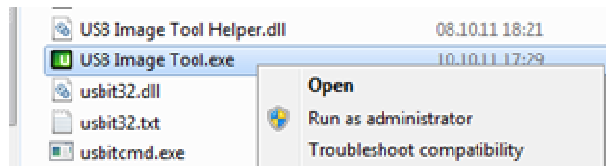
You can use the MD5 checksum to verify that the file has not been corrupted during transport. For secure deployments in a production environment, refer to the SHA checksum and verify it with NSN CSI Security before deploying the SW.

Mind further: We are describing a method to handle the images under MS-Windows because of the wide appreciation MS-Windows as in office environments, and the resulting great availability of MS-Win PCs. However: We strongly recommend for various reasons, that you should prefer the linux-way for this task. At least one reason should be most clear: Using dd under windows is cryptic, and if you use the wrong “of=”, you may destroy existing data volumes. Besides, each OS has its strengths and primary purposes: It took us 15 min to verify a correct result as described under linux, and 3h to do the same under MS-windows This document is written using MS Office.



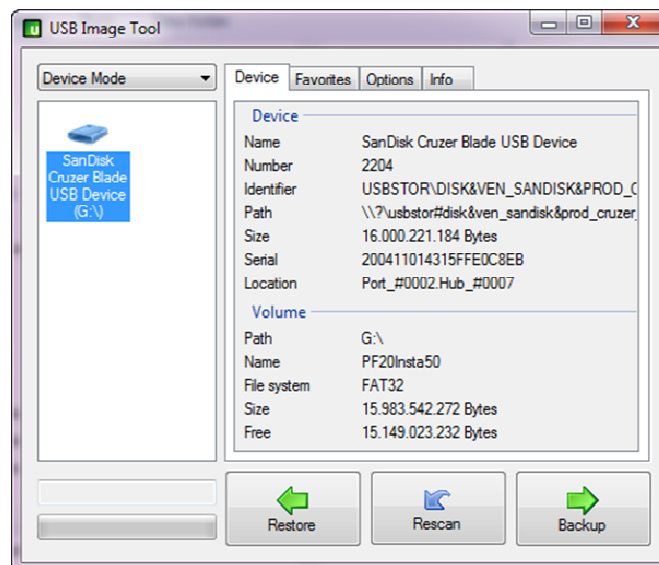
Creating the USB media using MS Windows:

1. You will need additional Software, since many tools natively available in Linux are not in the standard MS-Windows installation. We recommend to use “usbit”:
 - a. Download the tool from <http://www.alexpage.de/>
 - b. Run the tool with administrative rights to restore the image. In Windows 7, you right-click on the tool and choose “Run as administrator”.



If you can not run the usbit tool as administrator, you will not be able to create bootable drive images.

2. Verify that the checksum of the “.bin” image file your received from NSN is the same as the checksum that NSN provided you, to make sure there have not been any modifications or corruptions to the image. You can download a tool to verify file integrity e.g. from <http://support.microsoft.com/kb/841290>
3. Restore the image to your USB drive using usbit tool button “Restore” which will open a file-chooser where you can chose the image to restore. In order to update also partition table and boot-record of your USB media, make sure to run the application in “Device Mode”(needs admin privileges on your PC)



Creating the USB media using Linux:

1. You will normally not be required to install any additional Software, since recent linux versions do include all needed tools. However, you must make sure the following functions are available:
 - a. Checksum utility sha-1, as in “sha1sum” command. For example on EPEL type linux, it is contained in package coreutils-8.5.
 - b. Create raw disk-dumps using the linux “dd” command. For example, on EPEL type linux, it is contained in package coreutils-8.5.

Both abilities must be available, independent of your preferred tool!

2. Verify that the checksum of the “.bin” image file your received from NSN is the same as the checksum that NSN provided you, to make sure there have not been any modifications or corruptions to the image.

```
[root@samson Install_Media]# ll
total 571M
-rw-r--r--. 1 root root 571M Jul 15 12:34 PF20I50RH62-1.bin.gz
[root@samson Install_Media]#
[root@samson Install_Media]# sha1sum PF20I50RH62-1.bin.gz
673ce4a7bff4577e25a38b3f38692660eec4e3d5 PF20I50RH62-1.bin.gz
[root@samson Install_Media]#
```

3. Now you can dump the image file to a USB drive. Make sure that the USB drive has enough space for the uncompressed image file. (If the drive is bigger, it won't matter. But it can't be smaller). Plug in the USB-drive and run the dump:

```
[root@samson]# dd if= PF20I50RH62-1.bin.gz |gunzip -c > /dev/sdg
1167547+1 records in
1167547+1 records out
597784357 bytes (598 MB) copied, 184.253 s, 3.2 MB/s
[root@samson]#
```

Device files are of course specific to the computer in use. After the above command has finished, the pendrive (here as sdg) can be plugged into the PKI servers and used for installation.

It has shown important in our tests, that you do not use a “bs=” parameter for the dd command. Eventhough that would speed up the process, it has negative impact on the successful outcome. On most systems, it will take around 15-20 minutes to restore the usb-drive. You can use the “iostat” command under linux to check that the

Appendix B: Description of the Installation Media

PF20Insta50

name of the actual media, visible when USB drive mounted

ldlinux.sys
menu.c32linux OS loader from syslinux package, needed to boot
boot-menu chooser, needed to display install menu**inst_params.xlm****MS-Excel Macro-Sheet to define installation parameters**mkks61.exe
mkks61.plWindows standalone exe to extract parameters into kickstart files
native perl code of the mkks61.exeks-*.cfg
configure.sh
syslinux.cfg
stampfile.txtDynamic: Kickstart files for activated roles (created by excel-macro)
Dynamic: ENV file for customer parameters (created by excel-macro)
Dynamic: boot-menu definition file, needed by menu.c32 (created by excel-macro)
Dynamic: File to mark installation medias main parameters (created by excel-macro)RHEL62
RHEL62/images
RHEL62/images/README
RHEL62/images/TRANS.TBL
RHEL62/images/product.img
RHEL62/images/efidisk.img
RHEL62/images/efiboot.img
RHEL62/images/install.img
RHEL62/images/pxeboot
RHEL62/images/pxeboot/vmlinuz
RHEL62/images/pxeboot/TRANS.TBL
RHEL62/images/pxeboot/initrd.img
RHEL62/isolinux
RHEL62/isolinux/vmlinuz
RHEL62/isolinux/memtest
RHEL62/isolinux/boot.msg
RHEL62/isolinux/boot.cat
RHEL62/isolinux/isolinux.cfg
RHEL62/isolinux/vesamenu.c32
RHEL62/isolinux/grub.conf
RHEL62/isolinux/isolinux.bin
RHEL62/isolinux/initrd.img
RHEL62/isolinux/TRANS.TBL
RHEL62/rhel-server-6.2-x86_64-NSNCustom.isodirectory where bootable kernel for USB media is located
subdirectory for x86_64 boot support images
README file describing the boot images in this directory
ISO 9660 long file names transcript file
RedHat product specific information
Unified Extensible Firmware Interface CD/DVD boot image
Unified Extensible Firmware Interface USB boot image
RHEL installation image
Directory for PXE boot images
kernel for PXE boot
ISO 9660 long file names transcript file
initrd driver images for PXE boot
directory for ISOLINUX boot
kernel for booting ISOLINUX
binary for RAM test under ISOLINUX
File to set the boot message
El Torito boot catalog
Isolinux boot config
vesa based boot menu
default grub.conf
isolinux bootable image
isolinux initrd driver images
ISO 9660 long file names transcript file
customised RHEL media to install linux on PKI serversinsta
insta/ssh-ca-setup-nsn-certifsub.sh
insta/ssh-ca-setup-nsn-certifier.sh
insta/init-script.sh
insta/ICertifier
insta/certifier-5.0.0.x86_64.rpm
insta/certifsub-5.0.0.x86_64.rpm
insta/license-data.licdirectory for Insta specific scripts and SW
customized setup routine for insta certifier on FE
customized setup routine for insta certifier on BE
LSB compliant rc-init script for FE and BE
OCF HA Resource Agent for Insta
Insta SW for BE or SingleServer
Insta SW for FE only
Insta license file, to be supplied by NSNrpms
rpms/kernel-2.6.32-220.4.1.el6.x86_64.rpm
rpms/kernel-firmware-2.6.32-220.4.1.el6.noarch.rpm
rpms/cluster-glue-1.0.5-2.el6.x86_64.rpm
rpms/cluster-glue-libs-1.0.5-2.el6.x86_64.rpm
rpms/clusterlib-3.0.12.1-23.el6.x86_64.rpm
rpms/corosync-1.4.1-4.el6.x86_64.rpm
rpms/corosync-libs-1.4.1-4.el6.x86_64.rpm
rpms/pacemaker-1.1.6-3.el6.x86_64.rpm
rpms/pacemaker-libs-1.1.6-3.el6.x86_64.rpm
rpms/pacemaker-cluster-libs-1.1.6-3.el6.x86_64.rpm
rpms/pacemaker-cli-1.1.6-3.el6.x86_64.rpm
rpms/resource-agents-3.9.2-7.el6.x86_64.rpm
rpms/drbd-8.4.0-31.el6.x86_64.rpm
rpms/drbd-kmdl-2.6.32-220.4.1.el6.x86_64-8.4.0-31.el6.x86_64.rpmdirectory of non-standard rpms (HA and updates)
latest kernel update available at media-freeze
kernel firmware as required by the kernel
corosync/openais cluster-glue
corosync/openais cluster-glue libraries
RedHat specific cluster-lib
corosync HA
corosync HA libs
pacemaker resource manager
pacemaker resource manager libs
pacemaker - redhat integration libs
pacemaker crm cli
default resource agents
DRBD Software from LinBit
DRBD kernel module for RHEL 62script
script/config_functions.sh
script/setnsnenv.sh
script/config_security.sh.bak
script/config_network.sh
script/phase2.sh
script/config_drbd.sh
script/config_openssh.sh
script/config_insta.sh
script/config_corosync.sh
script/config_netsnmp.sh
script/config_ntpclient.sh
script/init_aide.sh
script/wrap_procfix.sh
script/watch_snmpd.sh
script/watch_fs.sh
script/startup_workaround.shHAPF2.0 installation and platform scripts
common function library used by HAPF2.0 scripts
script to setup installation environment (ENV)
script to configure platform security during installation
script to configure networking during installation
script to initiate PHASE2 of the installation after ANACONDA
script to configure DRBD during installation phase 2
script to configure openssh during installation phase 2
script to install and configure insta during installation phase 2
script to configure corosync during installation phase 2
script to configure net-snmp during installation phase 2
script to configure ntp-client during installation phase 2
script to initialize AIDE after installation (manual triggr)
wrapper for HAPF2.0 process supervision via SNMP
script for HAPF2.0 process supervision of SNMPD
script for simple, basic supervision of most critical FS security parameters
temporary workaround for a pacemaker/drbd issues at cluster startup

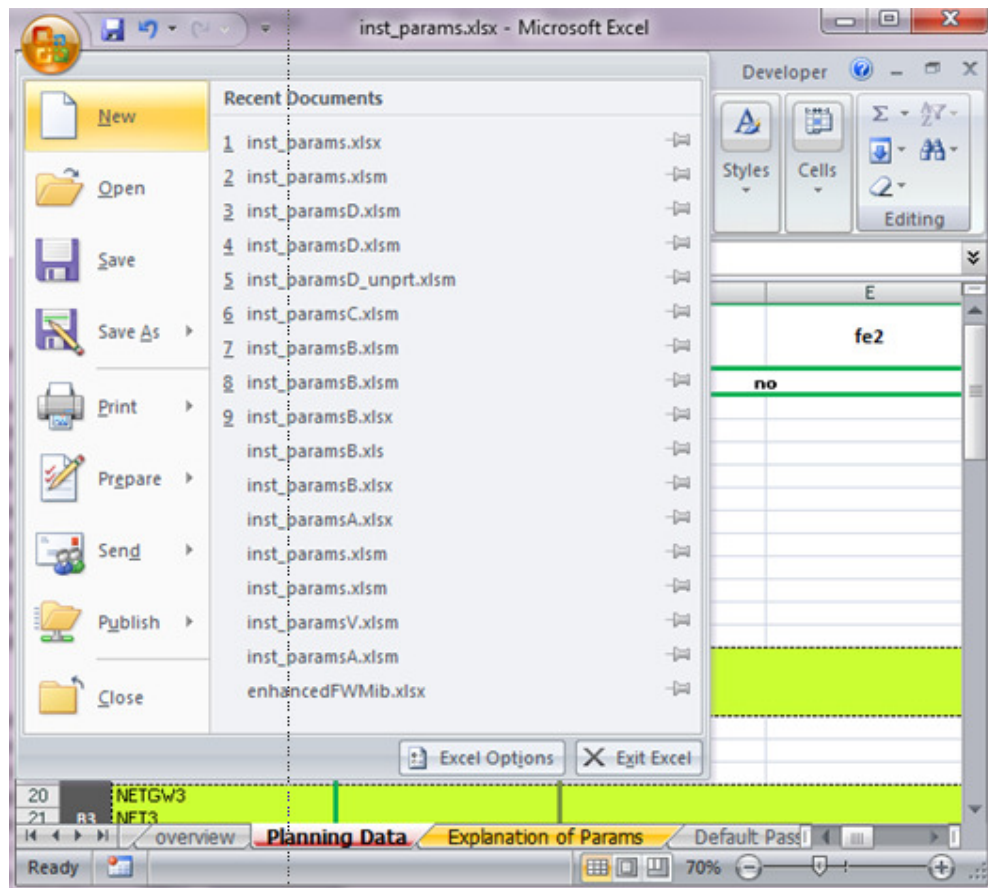


Appendix C: Enable Excel Macro execution

In order to use the inst_params.xlm macro sheet, execution of VBA macros in MS-excel must be enabled on your PC.

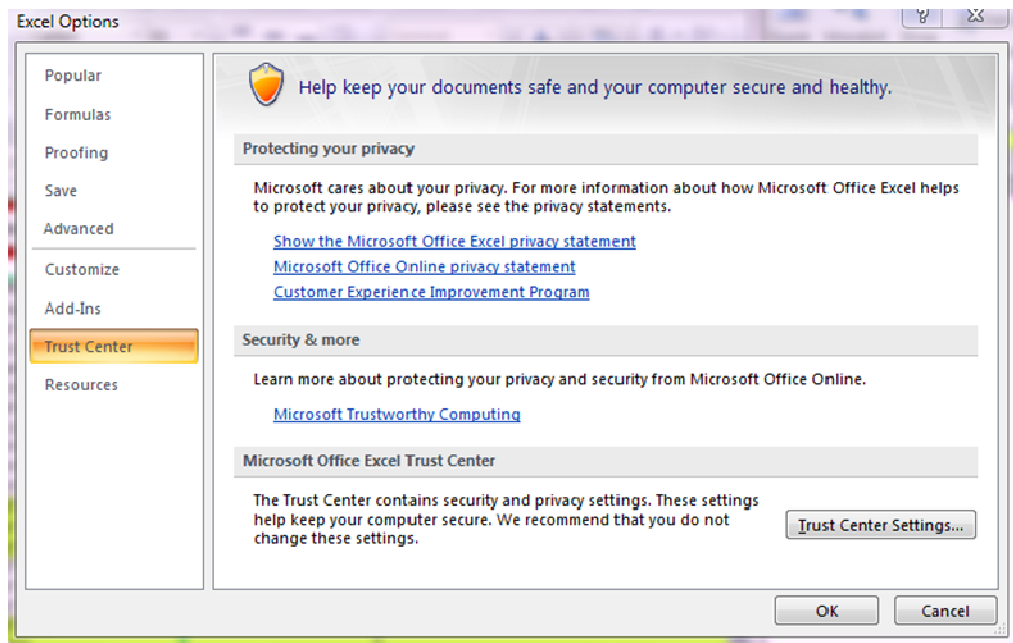
The following describes one of several possible ways to enable excel-macros:

- 1) Press the office-ribbon. By any way, you should see an “Excel Options” button in the lower right corner of the selection that pops up.

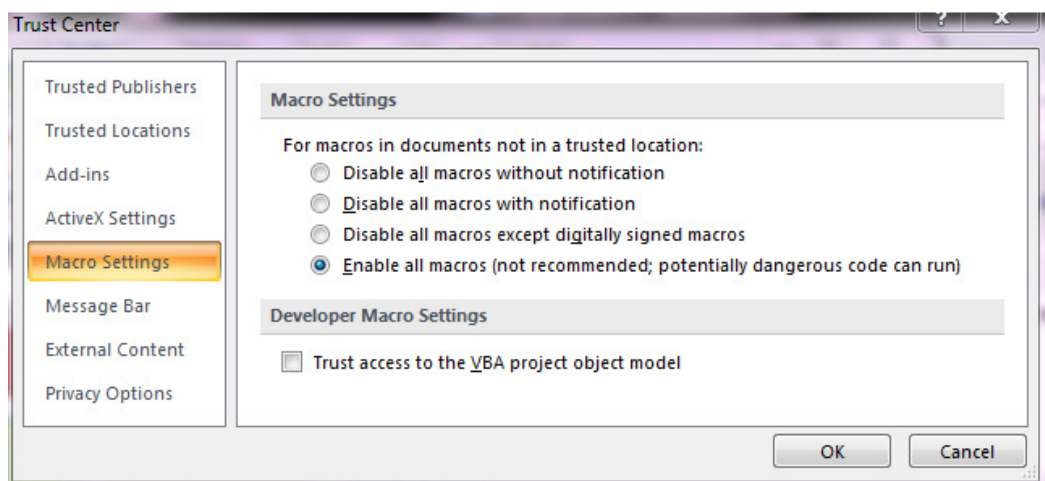




- 2) Press “Excel Options” and you will be able to select “Trust Center” (left side)



- 3) Select “Trust Center Settings” -> “Macro Settings” and then enable “Enable all Macros”. This setting is referred by MS as insecure and so it might be. Just open the Excel sheet to fulfill your task, reset Excel to it's previous security settings and close. The Macro delivered by NSN is not digitally signed but has been virus checked beforehand and is tracked by checksums, e.g. for the download. Hence it should be handled only on secure workstations as soon as it arrives at the customer!



- 4) Restart Excel and edit the Macro sheet.

Appendix D: References

Ref	Description
/1	NSN Insta PKI Installation Guideline for HAPF2.0
/2	LTE Transport Security Sizing & Ordering Guideline
/3	Radio Transport Security Solution - Technical Solution Description
/4	PKI Configuration Quick-Start Guide
/5	HAPF FM PM Reference
/6	HAPF20 Admin Notes (Collection of independent Leaflets)