Stand-Alone Integration

RBS 6402

OPERATING INSTRUCTIONS

Copyright

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Contents

1 Introduction 1

1.1 Limitations 1

2 Prerequisites 2

2.1 Documents 2

2.2 Tools 2

3 Setting up Client Laptop 3

4 Local Super User Account 4

4.1 Creating Client Side SSH Certificate 4

4.2 Preparing Combined File Template with Local Super User Account Details 5

4.3 Enabling Local Super User Account 5

5 Creating Certificates 7

5.1 Creating Certificates Flow Chart 7

5.2 Trusted CA Certificate 9

5.3 Operator Certificate and Credential for SEG 9

5.4 Operator Certificate and Credential for RBS 11

5.5 Preparing the ICF 12

6 Integrating with Stand-Alone 14

6.1 Preparations 14

6.2 Integrating using ORI 14

7 Disabling Stand-Alone Integration 18

7.1 Disabling the Local Super User Account 18

1 Introduction

This document describes the Stand-alone integration for the Pico Radio Base Station (Pico RBS). Stand-alone integration makes it possible to integrate without the use of Operations Support System – Radio and Core (OSS-RC).

This integration method is intended for temporary use during small projects like trials and demonstrations, with a limited number of nodes.

Without the connection to OSS-RC, normal live monitoring and remote connection will not be available through the OSS-RC.

An alternative remote support connection to the RBS must be provided by the local team responsible for the network setup.

The remote support connection is mandatory in order for Ericsson to deliver Customer Support services.

1.1 Limitations

For Stand-alone integration, the following limitations exist:

- No monitoring or supervision through OSS-RC
- No data collection through OSS-RC
- Only for a limited number of RBSs

2 Prerequisites

This section describes documents and tools that need to be at hand before performing Stand-alone integration.

2.1 Documents

The contents of the following documents are known:

- *Personal Health and Safety Information*
- *System Safety Information*

2.2 Tools

The following tools are available:

- A client laptop with the following installed:
  - Ubuntu Linux 12.04 LTS operative system installed
  - Secure File Transfer Protocol (SFTP) server
- Ethernet cables to connect the following:
  - Client laptop to the RBS
  - RBS to the Local Area Network (LAN)

3 Setting up Client Laptop

Ubuntu Linux 12.04 LTS operating system must be used since OpenSSH version 5.4 or later, is needed to support Secure Shell (SSH) certificate based authentication. The Linux operating system also simplifies the whole Stand-alone integration process.

Ubuntu Linux 12.04 LTS can be obtained from *http://releases.ubuntu.com/12.04/*

To use Ubuntu Linux in a laptop with Windows operative system, **VirtualBox** can be used. **VirtualBox** is downloaded from *https://www.virtualbox.org/*

After finished Ubuntu installation, the openSSH server can require a manual installation. If a manual installation is needed, use the following syntax:

```
sudo apt-get install openssh-server
```

4 Local Super User Account

The Stand-alone integration feature introduces the possibility to create a local super user account for the node. If O&M access to the node is required, it is mandatory to create a local super user account.

The local super user account uses SSH certificate based authentication. This means that the clients SSH RSA public key is signed with the private key of another key pair created by the operator. The later key pair acts as CA. The public key of the CA is imported into the RBS during the Stand-alone integration. During the integration process the public CA key makes it possible for O&M to create and enable the local super user account.

For more information regarding certificate handling, see *Security for O&M Node Access* and *IP Security*.

4.1 Creating Client Side SSH Certificate

A client laptop that uses local super user account to connect to the RBS, must be prepared before the integration.

Note: Certificate based authentication support is introduced in OpenSSH version 5.4.

To create a Client Side SSH Certificate, use the following procedure:

1. 1. Create a CA key pair. Its private key is used to create and sign the client certificate:

   ```
   ssh-keygen -f users_ca
   ```

2. If file `~/.ssh/id_rsa` does not exist, create a client key pair:

   ```
   ssh-keygen
   ```

3. Create a client certificate by signing the client public key (`~/.ssh/id_rsa.pub`) with earlier generated private CA key, that is, `users_ca`.

   ```
   ssh-keygen -s users_ca -I <user_full_name> -n
   <loginname> ~/.ssh/id_rsa.pub

   # where <user_full_name> can be any text identifying
   the certificate
   # <loginname> shall be set to desired user/ login name
   for super user account
   ```

4.2 Preparing Combined File Template with Local Super User Account Details

After the certificates are created on the client laptop, account details must be added to the Combined file template before the file is stored on the client laptop.

The Combined file consists of an Initial Configuration File (ICF) and an optional Site installation file. The Site installation file is not necessary if a DHCP provides the following IP addresses:

- For trusted Network: O&M IP addresses
- For untrusted Network: Public network IP addresses and Security Gateway (SEG) address

The content of public CA key created in Section 4.1 on page 4, that is, `users_ca.pub` is added to the Combined file template. The information is added to the `<superUserAccount>` tags and the `<usersCa>` attribute.

The variable between the `<superUserName>` tags must match the **loginname** given in Section 4.1 on page 4:

```
<combinedConfigurationFileType>
      <RbsSiteInstallationFile>
      ...
      </RbsSiteInstallationFile>
      <rpc>
      ...
      </rpc>
      <superUserAccount>
      <superUserName>oam</superUserName>
      <usersCa>ssh-rsa AAAB3NzaC1yc/
       4Yexample@sed652403</usersCa>
      </superUserAccount>
</combinedConfigurationFileType>
```

Example 1 Combined File Example

Note: The content of SSH-rsa public key is purposely shortened in Example 1

4.3 Enabling Local Super User Account

After the preparations described in Section 4.2 on page 4 are done, the account will later be created during the first phase of the Stand-alone integration. See Section 6 on page 14.

During this phase, O&M stores the account information to a permanent file system partition. The local super user account is not allowed to be active when the integration still runs in the Basic Software Package.

Note: Golden Software is another name for Basic Software Package, often used in non-CPI documentation.

O&M enables the certificate based authentication for the provided user account during the start of the second phase of selected integration procedure.

The following syntax is used to log in from the laptop with the local super user account:

```
ssh -v loginname@pRBS.ip.address
```

With SSH verbose enabled, the following prints in the console show that the certificate is used in the authentication:

```
debug1:  identity file ~/username/.ssh/id_rsa-cert type 4

debug1:  Offering RSA-CERT public key:  ~/username/.ssh/
id_rsa
```

5 Creating Certificates

This section describes how to create certificates. The method used to generate the certificates can be different, yet the result will be the same.

5.1 Creating Certificates Flow Chart

The flow chart in Figure 1 illustrates an overview over the certificates creation for the RBS and the SEG.

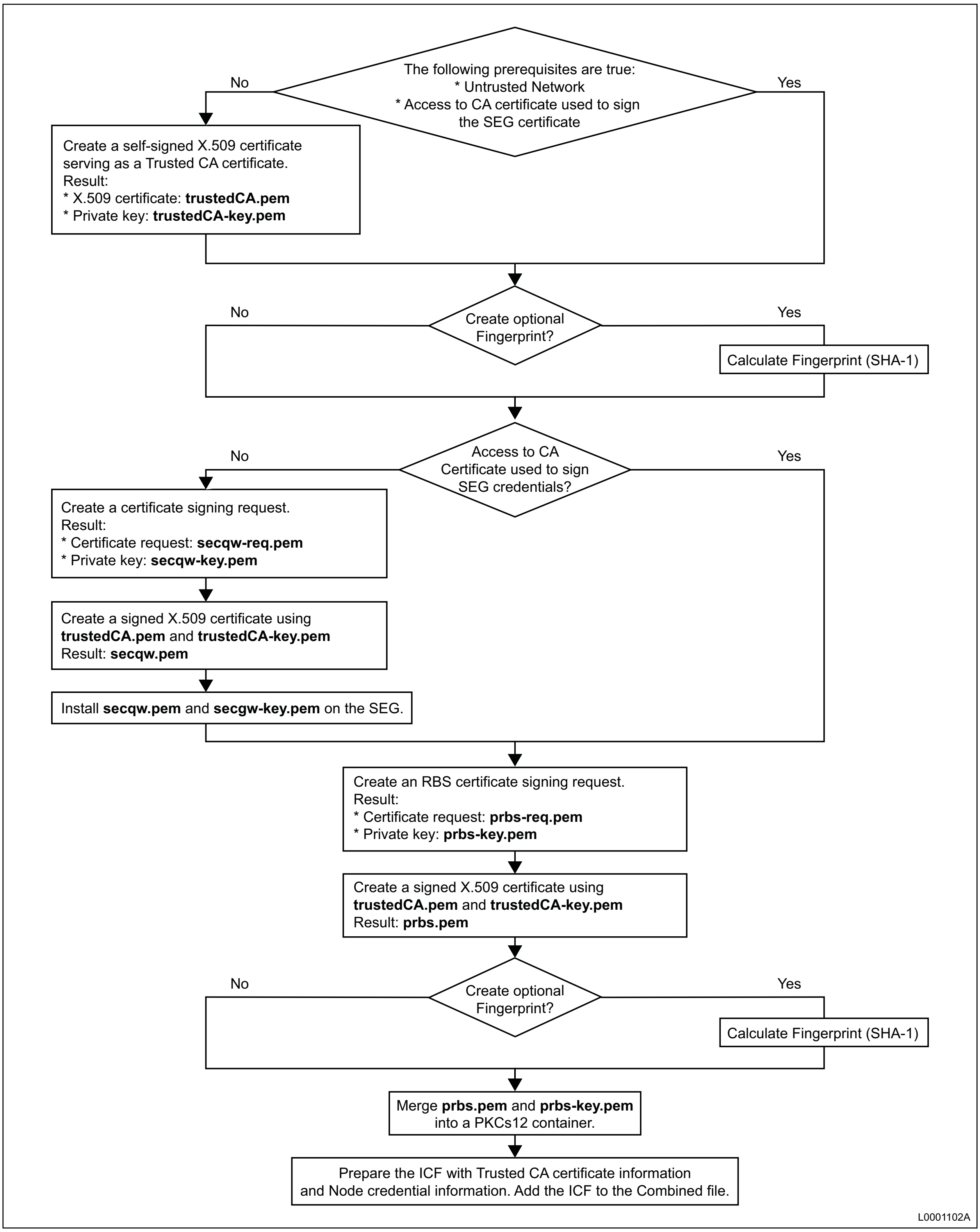

The following prerequisites are true:
* Untrusted Network
* Access to CA certificate used to sign
the SEG certificate
No
Yes
Create a self-signed X.509 certificate
serving as a Trusted CA certificate.
Result:
* X.509 certificate: trustedCA.pem
* Private key: trustedCA-key.pem
No
Create optional
Fingerprint?
Yes
Calculate Fingerprint (SHA-1)
No
Access to CA
Certificate used to sign
SEG credentials?
Yes
Create a certificate signing request.
Result:
* Certificate request: secqw-req.pem
* Private key: secqw-key.pem
Create a signed X.509 certificate using
trustedCA.pem and trustedCA-key.pem
Result: secqw.pem
Install secqw.pem and secgw-key.pem on the SEG.
Create an RBS certificate signing request.
Result:
* Certificate request: prbs-req.pem
* Private key: prbs-key.pem
Create a signed X.509 certificate using
trustedCA.pem and trustedCA-key.pem
Result: prbs.pem
No
Create optional
Fingerprint?
Yes
Calculate Fingerprint (SHA-1)
Merge prbs.pem and prbs-key.pem
into a PKCs12 container.
Prepare the ICF with Trusted CA certificate information
and Node credential information. Add the ICF to the Combined file.
L0001102A

Figure 1   Certificates Flow Chart

5.2 Trusted CA Certificate

If the following is true, step 1 to 3 in the procedure can be left out:

- Stand-alone integration is performed in an untrusted network
- The user has access to the CA certificate that was used to sign the SEG existing certificate

To create a CA certificate, perform the following procedure:

1. Use openSSL to create a self-signed X.509 certificate. This will serve as Trusted CA certificate.

   The `-days` attribute sets the validity period for the certificate:

   ```
   openssl req -x509 -nodes -newkey rsa:2048 -keyout
   trustedCA-key.pem -out trustedCA.pem -days 5000
   ```

2. Fill in the subject of the certificate, for example:

   ```
   Country Name (2 letter code) [AU]:SE
   State or Province Name (full name) [Some-State]:Stockholm
   Locality Name (eg, city) []:Kista
   Organization Name (eg, company)
   [Internet Widgits Pty Ltd]:Ericsson
   Organizational Unit Name (eg, section) []:Support
   Common Name (e.g. server FQDN or YOUR name) []:trustedCA
   Email Address []:no.name@example.com
   ```

3. As a result, two files are created. The X.509 certificate `trustedCA.pem` and its private key `trustedCA-key.pem`

   The content of the certificate can be printed as text to check that given parameters are correct:

   ```
   openssl x509 -noout -text -in trustedCA.pem
   ```

4. As an option, fingerprint (SHA-1) can be calculated for the certificate:

   ```
   openssl dgst -c -hex -sha1 trustedCA.pem
   ```

5. The certificate file `trustedCA.pem` is stored on the client laptop. Later, the certificate file can be installed to the RBS through SFTP.

   Optional fingerprint information must also be stored if fingerprint validation is to be used during certificate installation.

5.3 Operator Certificate and Credential for SEG

This section describes how to create certificate and credential for a SEG in an untrusted network.

Note: This section is not applicable if the user already have access to the CA certificate which was used to sign the SEG existing credential.

Also, it is not applicable for a trusted network where no SEG is needed.

If the user have no CA certificate access, a new credential must be created for the SEG. It must be signed with the self-signed CA certificate created in Section 5.2 on page 8.

To create a SEG certificate, perform the following procedure:

1. Use openSSL to create a certificate signing request:

   ```
   openssl req -newkey rsa:2048 -nodes -keyout
   secgw-key.pem -out secgw-req.pem -days 5000
   ```

2. Fill in the subject of the certificate.

   Note: The same information can be used as for the CA certificate, just different **Common Name** needs to be defined.

   ```
   Country Name (2 letter code) [AU]:SE
   State or Province Name (full name) [Some-State]:Stockholm
   Locality Name (eg, city) []:Kista
   Organization Name (eg, company)
   [Internet Widgits Pty Ltd]:Ericsson
   Organizational Unit Name (eg, section) []:Support
   Common Name (e.g. server FQDN or YOUR name) []:SecGW
   Email Address []:no.name@example.com
   ```

3. As a result, two files are created. The certificate request `secgw-req.pem` and the private key `secgw-key.pem`

4. Use the CA certificate `trustedCA.pem` and its private key `trustedCA-key.pem` to create a signed X.509 certificate:

   ```
   openssl x509 -req -in secgw-req.pem -CA trustedCA.pem
   -CAkey trustedCA-key.pem -set_serial 1234 -out
   secgw.pem -days 5000
   ```

5. The resulted certificate `secgw.pem` can be verified with:

   ```
   openssl verify -CAfile trustedCA.pem secgw.pem
   ```

6. Finally, the generated credential `secgw.pem` and `secgw-key.pem` can be installed on the SEG. For strongSwan based SEG implementations in lab environments, just copy the files to the strongSwan installation folder. For real hardware based SEGs, follow the manufacturer’s instruction for certificate installation.

5.4 Operator Certificate and Credential for RBS

To create an RBS certificate, perform the following procedure:

1. Use openSSL to create a certificate signing request:

   ```
   openssl req -newkey rsa:2048 -nodes -keyout
   prbs-key.pem -out prbs-req.pem -days 5000
   ```

2. Fill in the subject of the certificate:

   Note: The same information can be used as for the CA certificate, just different common name needs to be defined.

   ```
   Country Name (2 letter code) [AU]:SE
   State or Province Name (full name) [Some-State]:Stockholm
   Locality Name (eg, city) []:Kista
   Organization Name (eg, company)
   [Internet Widgits Pty Ltd]:Ericsson
   Organizational Unit Name (eg, section) []:Support
   Common Name (e.g. server FQDN or YOUR name) []:C827309086
   Email Address []:no.name@ericsson.com
   ```

3. As a result, two files are created. The certificate request `prbs-req.pem` and private key `prbs-key.pem`

4. Use the CA certificate `trustedCA.pem` and it's private key `trustedCA-key.pem` to create a signed X.509 certificate:

   ```
   openssl x509 -req -in prbs-req.pem -CA trustedCA.pem
   -CAkey trustedCA-key.pem -set_serial 5678 -out prbs.pem
   -days 5000
   ```

5. The resulted certificate `prbs.pem` can be verified with:

   ```
   openssl verify -CAfile trustedCA.pem prbs.pem
   ```

6. As an option, fingerprint (SHA-1) can be calculated for the certificate:

   ```
   openssl dgst -c -hex -sha1 prbs.pem
   ```

7. The operator node credential must be archived in the PKCS12 container. This is due to that the credential contains the private key, which must not be exposed.

   The PKCS12 is the only supported certificate format for Stand-alone integration.

   The certificate and the private key must be merged into one file and then password encrypted. The user will be prompted for the desired password. The file is stored in the PKCS12 container.

```
cat prbs-key.pem prbs.pem > combined_prbs.pem

openssl pkcs12 -export -in combined_prbs.pem -out
prbs_credential.pkcs12
```

8. The generated PKCS12 container `prbs_credential.pkcs12` is stored on the client laptop. Later, the PKCS12 container can be installed to the RBS through SFTP.

   Optional fingerprint information and container encryption password must also be stored in a safe location.

5.5 Preparing the ICF

To perform a Stand-alone integration on a node, the On-Site RBS Integrator (ORI) is used. This is the Graphical User Interface (GUI) in the RBS software.

Stand-alone integration is only possible with ORI and not during an autointegration. That is, Stand-alone integration is triggered by MOM actions that the user needs to define in the ICF. Before a Stand-alone integration can be initiated, necessary MO actions need to be added to the ICF. The ICF then needs to be added to the Combined file template inside the `<rpc>` tags.

The following templates are used:

- ICF - Trusted network. See *ICF Template Trusted Semi-AI*
- ICF - Untrusted network. See *ICF Template IPSec Semi-AI*
- Combined file. See *Combined File Template*

The ICF templates must be filled with values corresponding to the actual network and test environment.

Note: An exception is the definitions for registration authority servers under `<CertM>` object. The certificate enrollment is left out in the Stand-alone integration. Therefore, the server URIs, that is, `%TLSenrollmentServerAddressUri%`, must be defined with dummy IP addresses.

5.5.1 Trusted CA Certificate Installation Information

It is assumed that the user has stored the trusted CA certificate and the node credential created in Section 5.2 on page 8 in the `/var/ftp-` folder in the client laptop.

For the `<CertM>` object, the following lines must be added starting from `<certMId>1</certMId>`:

```
<CertM xmlns="urn:com:example:ecim:MSRBS_V1_CertM">
<certMId>1</certMId>
<installTrustedCertFromUri>
```

```
<uri>sftp://example@10.1.1.12/var/ftp/trustedCA.pem</uri>
<uriPassword>barfoo</uriPassword>
<fingerprint>69:5a:ee:57:ad:4f:b1:f1:5e:b3:ad:9c:45:70
</fingerprint>
<//installTrustedCertFromUri>
</CertM>
```

If the user wants to execute fingerprint check, fingerprint must first have been calculated for `trustedCA.pem` in Section 5.2 on page 8. It then needs to be added in the `<fingerprint>` attribute. If no fingerprint check is needed, the attribute is left undefined. `<uri>` always needs to point to the RBS local access port, that is, address range 10.1.1.0/24

5.5.2 Node Credential Installation Information

For the `<NodeCredential>` object, the following lines must be added starting from:

- `<nodeCredentialId>2</nodeCredentialId>` - Untrusted network
- `<nodeCredentialId>1</nodeCredentialId>` - Trusted network

```
<NodeCredential xmlns="urn:com:example:ecim:MSRBS_V1_CertM"
xc:operation="create">
<nodeCredentialId>2</nodeCredentialId>
<installCredentialFromUri>
<uri>sftp://example@10.1.1.12/var/ftp/prbs_credential.pkcs12</uri>
<uriPassword>barfoo</uriPassword>
<credentialPassword>12345</credentialPassword>
<fingerprint>49:f5:73:cb:a7:36:a5:cc:25:31</fingerprint>
</installCredentialFromUri>
</NodeCredential>
```

If the user wants to execute fingerprint check, fingerprint must first have been calculated for `prbs.pem` in Step 6 in Section 5.4 on page 10. It then needs to be added in the `<fingerprint>` attribute. If no fingerprint check is needed, the attribute is left undefined. `<uri>` always needs to point to the RBS local access port, that is, address range 10.1.1.0/24

5.5.3 Adding ICF to Combined File

After the ICF updates have been completed, the file content must be placed inside the `<rpc>` tags in the Combined file. Thereafter, the Combined file is stored on the client laptop.

6 Integrating with Stand-Alone

This section describes how to use ORI when performing a Stand-alone integration.

6.1 Preparations

Prior to integration, the following preparations must have been performed:

- The operator must have prepared the required node credential and trusted CA certificate.
- For integration in an untrusted network, it is assumed that the operator has installed corresponding SEG certificates. That is, the SEG certificate must be signed with the same trusted CA certificate that the operator is about to install in the RBS.
- The files to be installed in the RBS has been stored on the client laptop.

6.2 Integrating using ORI

To start the Stand-alone integration procedure, use ORI to download the Combined file created in Section 5.5 on page 12 from the laptop to the RBS.

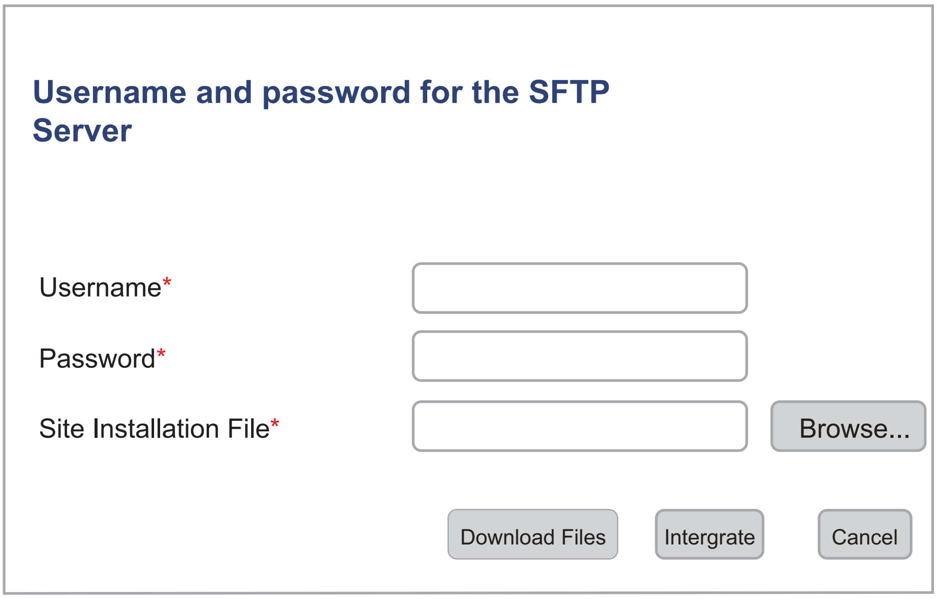
Perform the following procedure:

1. Use an RJ-45 cable to connect the client laptop to the RBS **WAN B** port.

   Note: To connect the cable to the WAN B port, an SFP module is used.

2. In the browser, enter the local access IP address `10.1.1.11`

   The ORI dialog box opens. See Figure 2.



Username and password for the SFTP Server

Figure 2 ORI Dialog Box

3. Enter the Username `admin` and the Password `webui`

4. In **Site Installation File**, enter the Combined file name. The file name with full path can be entered manually or with the use of the **Browse...** key.

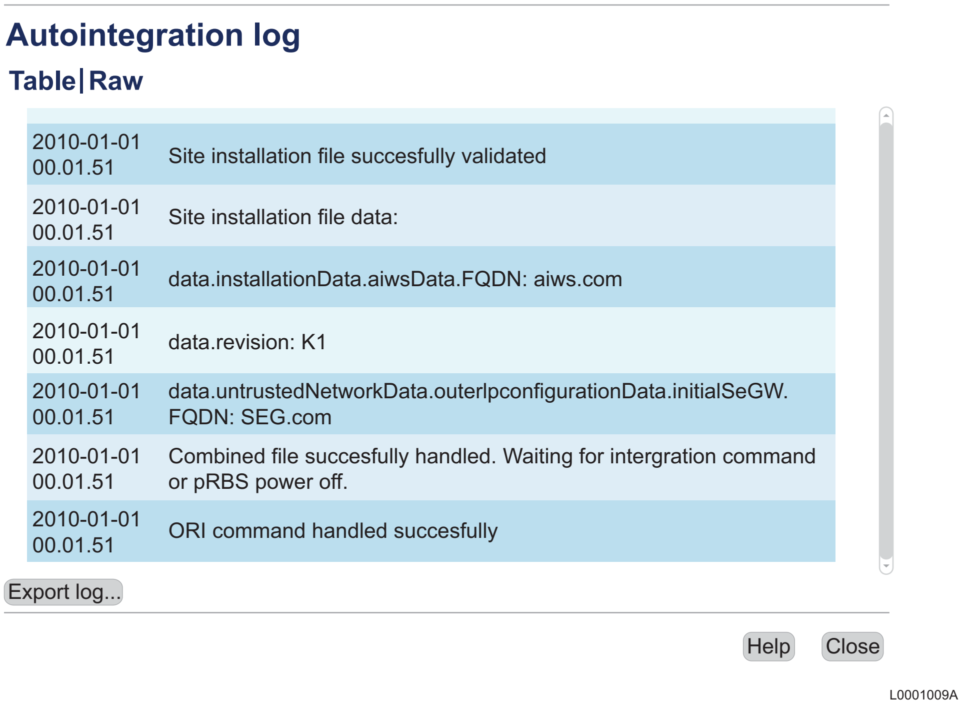
   Note: If the **Browse...** key is used, the Combined file needs to be located in the root folder of the SFTP server. **Browse...** only locates the file in laptop’s file system and does not return the full file path.

   The **Download Files** key becomes visible after all the details have been filled in.

5. Click **Download Files**. The download procedure starts.

   After the Combined file is successfully downloaded to the RBS, a confirmation message is printed in the **Autointegration log**. See Figure 3.

   The **Intergrate** key becomes visible.



Autointegration log

2010-01-01 00.01.51	Site installation file succesfully validated
2010-01-01 00.01.51	Site installation file data:
2010-01-01 00.01.51	data.installationData.aiwsData.FQDN: aiws.com
2010-01-01 00.01.51	data.revision: K1
2010-01-01 00.01.51	data.untrustedNetworkData.outerIpconfigurationData.initialSeGW. FQDN: SEG.com
2010-01-01 00.01.51	Combined file succesfully handled. Waiting for intergration command or pRBS power off.
2010-01-01 00.01.51	ORI command handled succesfully

Figure 3 Autointegration Log - Example

6. Click **Integrate** to initiate the integration.

   The integration starts. During the integration process, the RBS performs a restart.

   To confirm that the certificates are successfully installed on the RBS, the following transcript will be visible in the **Autointegration log**:

```
Stand alone integration.  Installing node credentials.

...

Installing node credential certificate for ipsec.

...

Node credential certificate downloaded successfully.

Trusted certificate downloaded successfully.

...

Reading of node certificate container successful.

Certificate fingerprint verification successful.

Trust chain is valid for certificate.
```

```
Node certificate written to secure storage.

...

Trusted certificate written to secure storage.

...

Certificates written to secure storage.

Node credential and trusted certificate installed
successfully.

ipsec certificate installation OK.
```

7. No further user interaction is needed. Wait a few minutes and check that the green optical indicator is lit and not blinking. When the green indicator is lit, the integration is finished and the RBS is up and running.

   Disconnect the client laptop cable from the RBS port.

7 Disabling Stand-Alone Integration

If a new integration of the RBS is planned, now with another integration method than the Stand-alone, the ICF must first be updated. The local super user account updates in the ICF need to be removed.

7.1 Disabling the Local Super User Account

To delete the local super user account information from the permanent file system partition, a factory reset must be performed. For information about the reset procedure and how to locate the reset button, see *Recovering a Node on Site*.

If a new Stand-alone integration attempt is planned, the Combined file must first be edited. The `<superUserAccount>` tags and the contents between the tags must be removed from the file. If not, O&M will use the information from the Combined file to create the local super user account once again.