

**INTERNAL AFFAIRS**



Te Tari Taiwhenua

National Dog Database  
Ta Interface Connection Guide

# **National Dog Database**

## **TA Interface Connection Guide**

**Version 1.2**

**December 2011**



## Revision History

Date	Version	Description	Author
February 2006	1.0	First release	Equinox
May 2010	1.1	Standardised SSH user ID and host examples	Equinox
December 2011	1.2	DIA Logo Update	Equinox



## Table of Contents

1.	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	NDD Interface System Availability	4
1.4	NDD Interface System User IDs and Passwords	4
2.	TA Extract File Transfer	5
2.1	Background	5
2.2	File Permissions	5
2.3	Interactive File Transfer Steps	5
2.4	Automatic (Unattended) File Transfers with Public-Key Authentication	7
2.4.1	Create Private and Public Keys	7
2.4.2	Creating the Private and Public keys using PuTTY and PuTTYgen	8
2.4.3	Add the Public Key to the TA's NDD Server <code>authorized_keys</code> File and Test	8
2.4.4	Automate the Running of the SCP Command at a Predetermined Time	9
2.4.5	Retrieval of the XML To-Do List Report using SCP	9

# TA Interface Connection Guide

## 1. Introduction

The Dog Control Act 1996 requires the territorial authorities (TAs) in New Zealand to provide information on dog registrations within their TA to the National Dog Database (NDD). The information is to be provided in XML format as described in the TA Interface Schema and the associated TA Interface Guidelines.

Two types of file may be provided, Incremental or Full, both of which have the same format, but which contain different selections of records.

- Incremental files - these are provided on a regular (probably daily) basis, and contain only those records which have been added, changed, or deleted on the local dog control system since the last extract.
- Full files - these are provided on an as required basis (as arranged with the NDD National System Administrator) and contain all "current data" from the local Dog Control System.

Incremental files are scheduled for automatic extraction and transfer. Full files are extracted and transferred by specific request.

### 1.1 Purpose

The purpose of this document is to describe how a TA sets up the connection to transfer Incremental and Full files to the NDD.

### 1.2 Scope

This document only covers the access and use of the NDD file transfer interface. For all other information about the NDD, TAs should refer to their nominated NDD local administrator. This document describes the steps involved in using a secure copy (scp) client application to transfer a TA extract file to the NDD server. It also provides some background technical information.

### 1.3 NDD Interface System Availability

The NDD server is generally available 24 hours a day 7 days a week, except when taken down for administration or maintenance purposes. Support for the interface is only available on business days during normal business hours.

### 1.4 NDD Interface System User IDs and Passwords

Each TA will be supplied with a file transfer user ID and password, which are to be used for all NDD file transfers from that TA.

## 2. TA Extract File Transfer

### 2.1 Background

The interface site is using **Secure Shell (SSH)** server to allow secure transfer of the TA XML files to the NDD. Two methods of user authentication are available, these being password authentication for interactive transfers, and public-key authentication for non-interactive usage.

The interactive (password) authentication is a manual means of sending a file, and is provided for use during the initial set up of connections, or for troubleshooting, to prove that a file can be sent between the TA and the NDD. It is not the method that will be used for the regular sending of files once the system is in "production", as it is expected the regular transfers will be automated.

The public-key authentication can be used for unattended file transfer (i.e. no user present to enter a password). The public-key authentication requires the TA to create a private/public key pair and to provide the public key to the administrators of the NDD server for installation on the server. This is the method expected to be set up for regular transfers of data to the NDD.

Each TA has their own SSH userid and upon authentication will be placed into their own home directory (using the UNIX convention referred to as "~"). Each TA's home directory will also have a .ssh subdirectory ( ~/.ssh) which is used to store the TA's public key to allow non-interactive file transfer (this is described in detail in section 2.4)

The extract file should follow the naming conventions described in the TA Interface Guidelines.

When the file is processed by the NDD it is moved out of the TA's home directory.

### 2.2 File Permissions

As SSH preserves file access permissions when a file is transferred, if the file does not allow other users to read the file the NDD application will be unable to open the file to process it. To ensure the file is readable the `chmod` command (in UNIX) can be used to allow the NDD Application to read it before it is sent. For example in UNIX the command:

```
chmod o+r extractfile.xml
```

makes the file `extractfile.xml` readable by the NDD application.

### 2.3 Interactive File Transfer Steps

This manual interactive file transfer method is provided for use during initial set up to prove the connection, or for troubleshooting.

The **Secure Copy (scp)** command can be used to transfer the extract file.

#### ***Interactive Command Line SCP Transfer***

The basic format of a scp command to upload to remote host from your local machine is:

```
scp localfilespec myusername@remotehost:remotefilespec
```

For example the command:

```
scp extract.xml <ssh-user-id>@ssh.ndd.govt.nz:001-20050630-1.xml
```

Is used to copy the file `extract.xml` to the TA's upload directory on the server `ssh.ndd.govt.nz` with the name `001-20050630-1.xml`. *Where <ssh-user-id> is the TA's SSH user name (as advised separately).*

When transferring a Full file, the same process is used. The only difference is that the file name will contain the word "FULL" in the place of the batch sequence number (as well as the file type in the file header being FULL, not INCR).

After the above is entered the password will be requested and if it is correct the transfer will occur. For example:

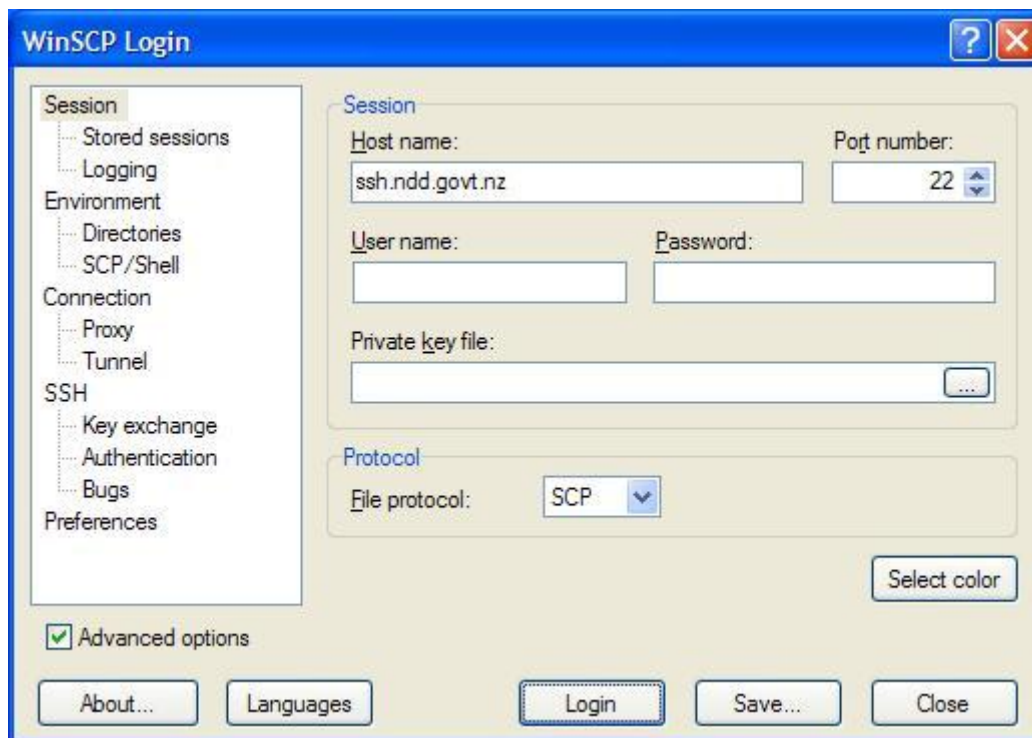
```
<ssh-user-id>@ssh.ndd.govt.nz 's password:*****
extract.xml                100% 5067    5.0KB/s   00:00
```

Where `<ssh-user-id>` is the TA's SSH user name (as advised separately). The scp uses SSH to securely transfer files between computers on a network. SCP and SSH are normally always included with Linux and Unix distributions. In a windows environment a command line SCP is available by installing the free Cygwin environment which provides Linux / Unix programs for the windows environment. (see <http://www.cygwin.com/> )

### Interactive Graphical Interface SCP Transfer

Another option for the Windows environment is the WinSCP program. WinSCP is an open source SCP client for Windows. Its main function is the secure file transfer between a local and a remote computer. Beyond this, WinSCP offers basic file manager functionality. It uses Secure Shell (SSH) and supports the SCP protocol (see <http://winscp.net/eng/index.php> ).

As shown in the figure below the WinSCP login dialog box is used to configure the session parameters, which can be saved and reconnected later.



The settings are:

Host name: ssh.ndd.govt.nz  
 Port number: 22  
 Username: The TA's SSH user name (as advised separately)  
 Password: The TA's SSH user password (as advised separately)  
 Protocol: SCP



## 2.4 Automatic (Unattended) File Transfers with Public-Key Authentication

In the operational environment, the extract file uploads will occur automatically after the end of the business day, so an unattended SSH SCP command needs to occur.

The use of an unattended SCP command requires the use of public-key authentication to avoid the need for a user to enter a password.

To keep the instructions consistent the following definition of terms used in this section:

- *TA client system*: the system at the TA that is transferring the file to the NDD.
- *NDD server*: the NDD SSH Server.

Setting up the unattended transfer involves 3 major steps:

1. Creating the private and public keys
2. Add the public key to the TA's NDD server `authorized_keys` file and test
3. Automate the running of the SCP command at a predetermined time each business day

The details of each of these steps are described in the following sections.

### 2.4.1 Create Private and Public Keys

The actions shown are based on using the OpenSSH implementation of SSH. Key generation may vary under different implementations of SSH.

To confirm that OpenSSH is the SSH software installed on the TA client system. The `ssh -V` command should print a line beginning with OpenSSH, followed by other details.

```
$ ssh -V
OpenSSH_4.1p1, OpenSSL 0.9.7g 11 Apr 2005
```

A RSA key pair must be generated on the TA client system. The public portion of this key pair will be transferred to the NDD server, while the private portion needs to remain on a secure local area of the TA client system, by default in `~/.ssh/id_rsa`, i.e. in the `.ssh` subdirectory under the home directory of the user the transfer will be performed under.

The key generation can be done with the OpenSSH `ssh-keygen` utility. The following commands create a `.ssh` directory under the root directory, make the directory non-public, and generate an RSA key file.

```
client$ mkdir ~/.ssh
client$ chmod 700 ~/.ssh
client$ ssh-keygen -f ~/.ssh/id_rsa -t rsa
Enter passphrase (empty for no passphrase): ...
Enter same passphrase again: ...
```

The meaning of the parameters passed to the `ssh-keygen` are:

- f <filename> filename of the key file, in this case "id\_rsa"
- t <type> the type of key to create, in this case an "rsa" key



When the passphrase is requested do not enter anything<sup>1</sup>, just press enter twice.

The file permissions should be locked down to prevent other processors or users from being able to read the key pair data. OpenSSH may also refuse to support public key authentication if the file permissions are too open.

```
$ chmod go-w ~/
$ chmod 700 ~/.ssh
$ chmod go-rwx ~/.ssh/*
```

#### 2.4.2 *Creating the Private and Public keys using PuTTY and PuTTYgen*

A popular Windows FTP tool is PuTTY, (see <http://www.putty.nl/>) which has an associated key generation utility called PuTTYgen which can be used to generate keys. However if the PuTTYgen tool is used there is an additional step required to get the key in the correct format for OpenSSH.

Keys can be in one of two formats, the SECSH format or the OpenSSH format. PuTTYgen uses the SECSH format for public keys, but its own unique format for private keys.

However PuTTYgen can read and write private and public keys for both OpenSSH and SECSH formats. To get the SECSH formatted key into OpenSSH format using PuTTYgen follow these steps:

1. Click the Load button and select the key file (you need to set file types to "All Files (\*.\*)" for non-PuTTY keys to show up).
2. Use the Conversion menu and select "Export OpenSSH key".
3. To write an OpenSSH public key, the user needs to copy and paste the key shown at the top of the PuTTYgen Window into the id\_rsa.pub file.

#### 2.4.3 *Add the Public Key to the TA's NDD Server `authorized_keys` File and Test*

To add the public key to the NDD Server just email the public key (i.e `id_rsa.pub`) to the NDD National System Administrator who will install it on the NDD SSH Server and advise the TA when it has been done.

Once you have been advised your public key has been installed on the server you are in a position to test it.

Many different things can prevent public key authentication from working, so be sure to confirm that public key connections to the server work properly. Try logging onto the NDD server again using SSH. This time SSH should connect without any request for a password or passphrase.

```
client$ ssh <ssh-user-id>@ssh.ndd.govt.nz
server$
```

*Where <ssh-user-id> is the TA's SSH user name (as advised separately).*

---

<sup>1</sup> If you wish to protect the private key with a passphrase, then use will need to be made of ssh-agent or similar utility to provide the passphrase when the automated SCP command is run. The configuration of this can be complex and will differ significantly for different Territorial Authority IT environments.



If there are issues at this stage using the “-v” can be passed to the ssh command to provide verbose output that may be useful in troubleshooting the issues.

SSH is strict about file and directory permissions and may refuse to use a private key file if the permissions to access that file are too lenient.

The SCP command can now be run without requiring the user to enter a password, for example:

```
scp extract.xml <ssh-user-id>@ssh.ndd.govt.nz:001-20050630-1.xml  
extract.xml 100% 5067 5.0KB/s 00:00
```

Where <ssh-user-id> is the TA's SSH user name (as advised separately). When transferring a Full file, the same process is used. The only difference is that the file name will contain the word “FULL” in the place of the batch sequence number (as well as the file type in the file header being FULL, not INCR).

#### 2.4.4 Automate the Running of the SCP Command at a Predetermined Time

How the SCP command is automated depends on whether scheduling functionality is built into the vendor's dog control system, or the scheduling facilities of each TA's IT environment is used - such as the cron facility in UNIX / Linux environments, or the Task Scheduler in the Microsoft Windows environment.

The main consideration would be that when the SCP command is called it has rights to access the private key file (`id_rsa` in our example) to allow the authentication to occur. The process that runs the SCP should be the owner of the private key file.

#### 2.4.5 Retrieval of the XML To-Do List Report using SCP

If the TA has elected to receive the period processing To-Do List Report as XML, then the XML file will be placed in a subdirectory called `\out\` under the TA's home directory on the SSH Server. The TA also receives an email with a link to another copy of this XML file kept on the NDD Application Web Server.

The copy in the SSH `\out\` directory is overwritten each processing period, but the copies available on the NDD Application Web Server are available for an extended period of time (when they are removed is at the discretion of the NDD operations staff, and is only limited by disk storage).

The file on the SSH Server will always be called `ToDoListMessagesReport.xml`. The SCP command to retrieve the file is:

```
scp <ssh-user-id>@ssh.ndd.govt.nz:out/ToDoListMessagesReport.xml .
```

Where <ssh-user-id> is the TA's SSH user name (as advised separately).

Note the space and full-stop at the end of the command – these are necessary.