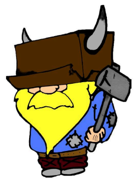


Forseti Digital

www.forsetidigital.com

Remote Desktop/Terminal Service Integrated Client for Android & BlackBerry PlayBook

The Application

Terminal Services (also known as Remote Desktop Protocol or RDP) is the Microsoft Windows protocol and feature that lets you remotely log-on and work at a computer as if you were seated at your local machine. Well now you can do that from your Android device - even your phone! Imagine this - you can work with your applications, manage files, perform various administrative tasks, develop world shaking software (probably not!) - all while away from your computer using this application.

Benefits

- ***** NEW *** Gateway bug fixes, UI changes, .rdp files and more (version 2.7)**
  - Bug fixes in TS Gateway support
  - revamped UI look and feel
  - support for .rdp file use for connection parameters
  - trace files saved on external storage for ease of retrieval
- ***** NEW *** Color & Screen support (version 2.2)**
  - 8, 15, 16 (default) & 24 bit Colour for crisp, beautiful display
  - screen orientation (portrait or landscape - the default) and size support
  - special tablet support to make screen size the same as the tablet's size (see screen=display below)
  - default parameter set that is retrieved on startup to make access to common systems simpler.
- ***** NEW *** TS Gateway & Data Integration (version 2.0)**
  - **Data integration**: (aka file virtual channel support): map your device disk and have it appear as a mapped network drive on Windows - move files back and forth, print device files on your Windows printers etc. etc. See below.
  - **Terminal Services Gateway:** connect seamlessly through a TS Gateway.
- **security**: All data sent from the device to the server is protected by 128 bit encryption. In addition you are using a native, built in capability of Windows not some 3rd party

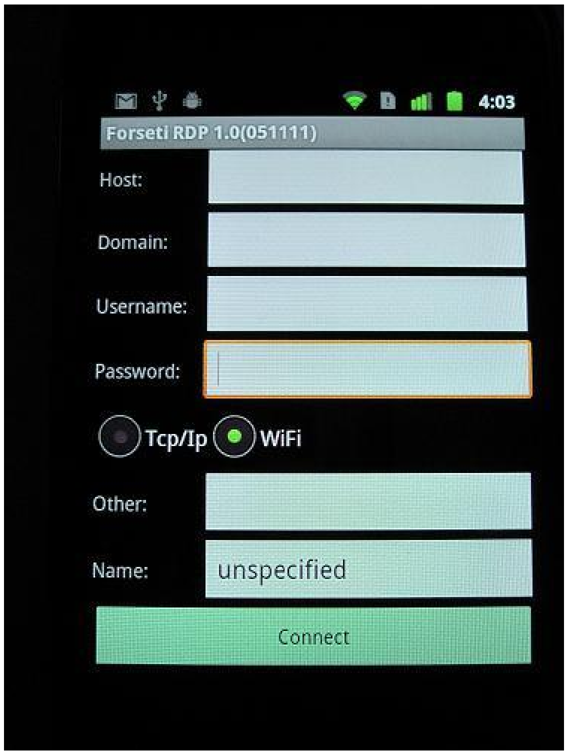
software that you have to install on your computer.

- **access**: You get easy remote access to your servers and computers and their business applications anywhere, anytime. Windows XP, 7, Server 2003, Server 2008 R2 anyone?!
- **customer satisfaction**: You can fix problems on servers and office computers remotely without having to be on site. For example you could start or stop services, examine log files all from your device. This allows you to provide timely and quality support to your internal and external customers.
- **productivity**: Enhance your and your employees' productivity and mobility.
- **cost**: At this very affordable price why would you want to be without this capability?! You never know when it could come in handy.
- **multi-language support**: uses the Locale of the device to establish the language. See list below.

***BlackBerry Playbook* users please consult the specific PlayBook FAQ section at the end of this document.**

Getting connected

When you start the app you will see something like this: (as of 2.7 the UI look and feel has changed)



Host:
Domain:
Username:
Password:
Other:
Name: unspecified

- **Host** is the host name of your remote computer. It can be a simple IP address too.
- **Domain** is the Windows domain name you need. Sometimes you don't need to supply

anything. Contact your System Administrator for help in this area.

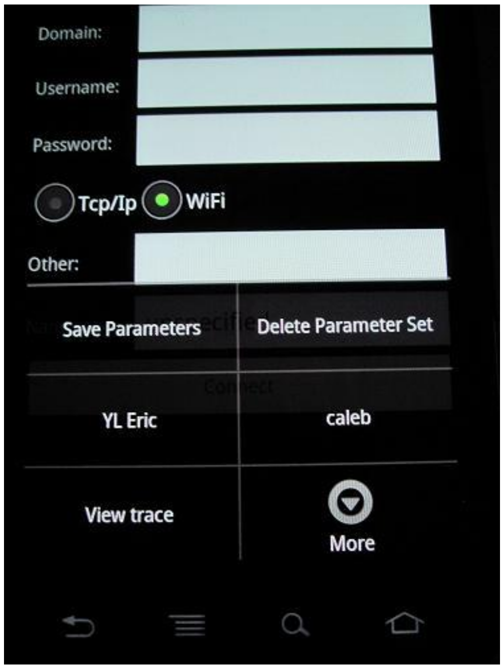
- **Username** is the username or account name! Duh!
- **Password** is what you would expect it to be - the password for this account.
- The radio buttons refers to the kind of connection you would like to attempt. Since the protocol is supported over TCP network connections, you have a number of options for this. Only the ones available for your device will occur in this choice/drop down menu. The available options are:
    - **Tcp/Ip** means you can make direct connections using TCP/IP.
    - **WiFi** means your device is WiFi/wireless enabled and you would like to connect over WiFi.
- **Other** refers to some optional collection of parameters you might want to supply. We've done it this way to reduce the clutter on the screen. Often nothing needs to be specified here. The parameters are usually keyword/value pairs and the minimum length of the keyword that needs specifying is indicated by the underline. Parameters are separated by blanks. Parameters you can specify here include:
    - port=<port number> The default TCP port used for the protocol is 3389. Suppose your remote computer has been configured to support RDP over port 3333 you could specify port=3333 here.
    - portrait or landscape to specify the orientation of your device. The default is landscape.
    - screen=WxH where W is the screen width in pixels and H is the screen height. The minimum size is 800x600.
    - screen=display to automatically match the remote server screen size to the size of your tablet. This saves you having to use "fling" gestures to move your display window over the virtual server screen. Very useful for tablets such as Playbooks.
    - DEPRECATED height=n n is the height of your virtual screen. See Screen Real Estate below.
    - gateway=gateway address
    - So for example if your environment was configured for RDP access on port 6534 with a screen sized the size of your tablet connecting via a gateway at mygateway.com, you would specify *"port=6534 screen=display gateway=mygateway.com"* in this field. Of course don't use the quote characters!
- **Name** refers to the name of this set of parameters. You wouldn't want to type all this same information in every time you accessed this remote machine would you? Not! So there's the ability for you to save and persist this set of connection parameters under a name. In the screen shot above no name has been specified. That's why the name is **unspecified**!
    - if you save a set of parameters under the name "default" then the app will automatically load that set of parameters at startup. Useful for cases where you basically only log into one remote Windows machine most of the time.

Once you've filled in the parameters you're ready to press the Connect button.

Connection Parameters

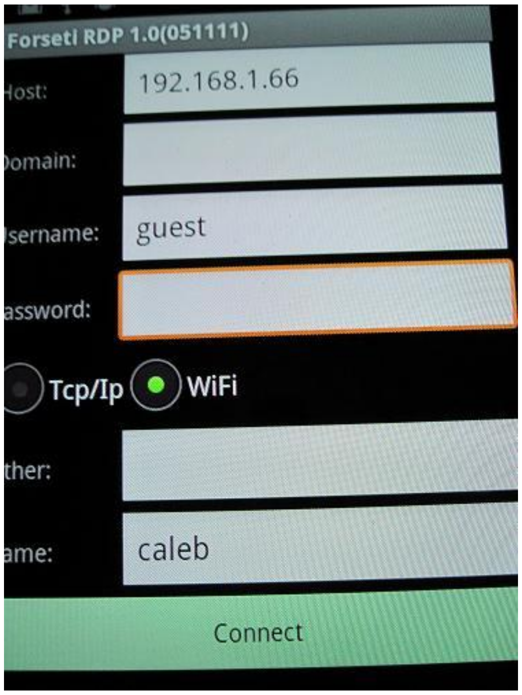
You want to be able to save the connection parameters you use to connect to your stable of machines! Or maybe just one! To do that you would do the following:

- Type in the parameters of interest for your connection.
- Give the parameter set a name and save the connection parameters using the **Save Parameters** menu item which looks like this:



The above menu also illustrates how you see the set of saved parameters already saved on this device. In this case they are called **caleb** and **YL Eric**.

So when you start up the app and want to connect to **caleb** you simply select **caleb** (or whatever your chosen name for this parameter set is) and you will see the screen like this:


Forseti RDP 1.0(051111)

lost: 192.168.1.66

omain:

sername: guest

assword:

Tcp/Ip [✓ WiFi]

ther:

ame: caleb

Notice a few things about this:

- The parameters you saved under the name **caleb** are there.
- Except for the password!! The cursor is positioned for you to enter the password. Why? This is by design for security purposes. All we need is yet one more place for security to be compromised right? Not! Just imagine someone finds your device with this app and your set of connection parameters INCLUDING passwords. Not a happy thought. Sure we could password protect this but then you have ONE MORE PASSWORD to remember. Wouldn't that be sweet?! Besides, you probably have more secure ways of remembering passwords. All this to say that we don't think saving the password with the other parameters is a good idea. So we don't do it.

Screen Real Estate, density and other stuff

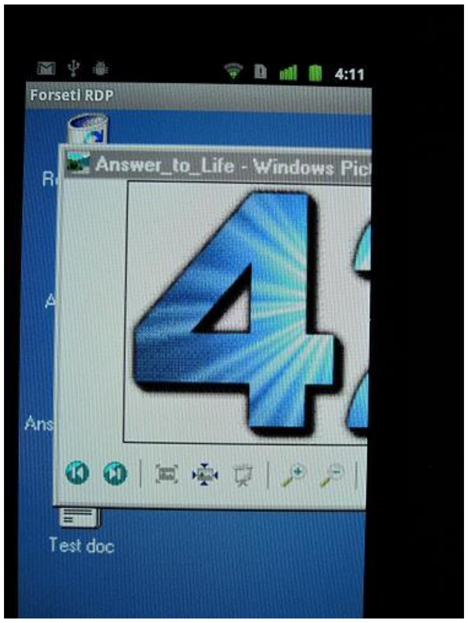
So we're not talking about real estate as in Vancouver real estate which is way over priced. We know. We live here. Back to the topic. We're talking in computer jargony terms about the space available to display your remote computer's screen. Let's face it: there's not a lot. It is never going to be like using your big flat monitor back home or in the office.

Note the following:

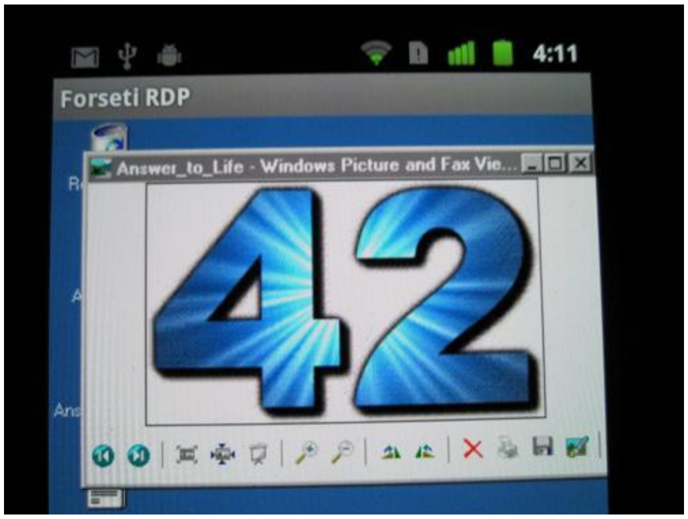
- **Screen size** for the Windows server can be specified giving a width and height.
    - screen=WxH where W & H are the width and height. The default is screen=800x600.
    - screen=display automatically sizes your screen to that of your device. Useful in

tablet situations where you want to avoid using “fling” to move your real window around the server’s virtual window. See below.

- **View window**: this means your actual display (e.g. your device screen) is a window on a larger virtual screen which you move about using gestures (see below).
- **Density**: aka dots-per-inch (dpi) Since viewing screens at the default screen density (or dpi) can make them almost unreadable (unless you happen to carry a larger magnifying glass with you!) the application initially scales the density to something considered ‘readable’. In our case that is 140dpi. Of course one can use the pinch/spread gestures to zoom the screen display (see below).
- **Orientation**: the application supports setting the orientation of the device. The default is landscape. For example if one was in portrait mode displaying a screen like the following:



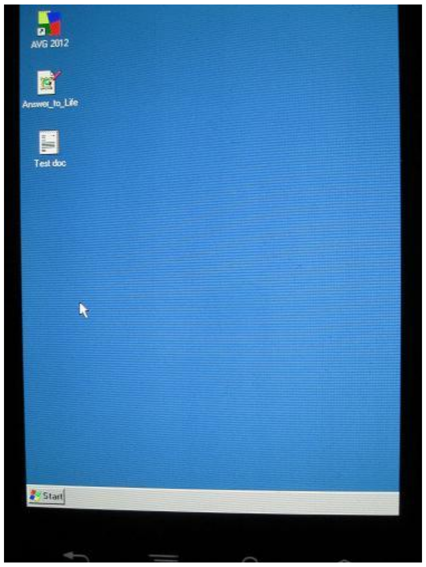
Specifying landscape orientation (simply use “landscape” on the other field) would display the following:

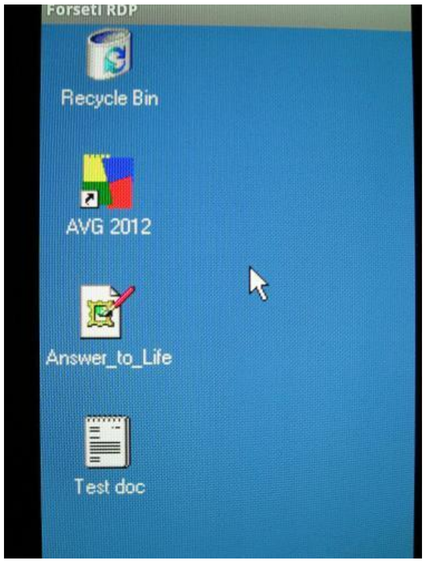


Touch Mode & Gestures

One of the appeals of the touch screen nature of modern Android devices is that we can use those features naturally in our interaction with the server.

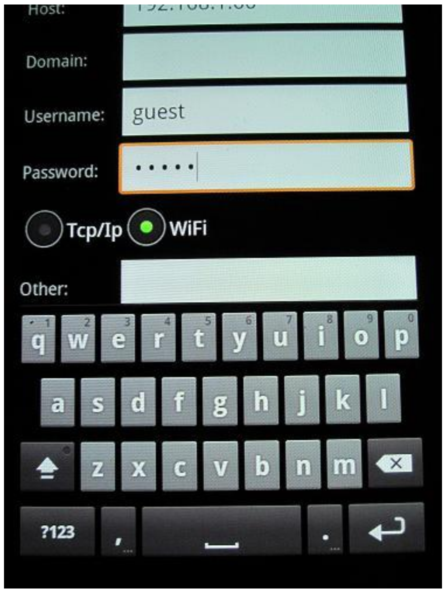
- A “single tap” is the same as clicking the mouse at a particular location.
- A “double tap” is the same as double clicking the mouse.
- A “long press” is as if you had “right clicked” the mouse. Sometimes it is more useful to use the “Right Click” menu that performs a right click action at the current cursor location.
- A “long press” followed by a move is as if you pressed down the mouse and moved it. e.g. selecting text, clicking in the title bar and moving the window.
- Moving or scrolling your finger is like moving the mouse.
- “Flinging” your finger (scrolling quickly) iscrolls the screen - moving the window on the larger real “screen” in the x, y directions of your “fling”, leaving the mouse position unchanged.
- **Multi-touch**: the application implements two multi-touch gestures:
    - pinching/squeezing your fingers together is like zooming in
    - squeezing them apart is like zooming out as the following two images illustrate.



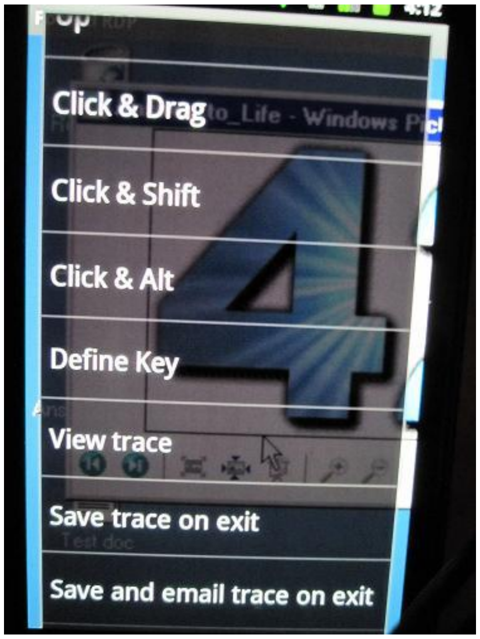


Keyboard input

- For touch screens the touch events allow you to move the mouse.
- For non-touch devices the application recognizes that moving the trackwheel is moving the mouse. Pressing it is like pressing the mouse left button.
- Obviously for devices with "soft" keyboards you have to be able to enter "key strokes". To do that one uses the menu **keyboard** option. It will show something like:


Host:
Domain:
Username: guest
Password: •••••
Tcp/Ip [✓ WiFi]
Other:

- A limited set of mouse click modifier actions are also supported. The menu illustrates the following click "modifiers":



- **"Click & Drag"** provides the common drag function. **You won't need this for a touch enabled device** (see above) Click once in the title bar of a window to drag, use the menu to select "click & drag" and then click where you would like the drag to occur.
- **"Click & Shift"** makes the *next* mouse click be interpreted as if that had been done holding down the shift key.
- **"Click & Alt"** is similar to above except for the "alt" key modifying the next click.

The Menu explained

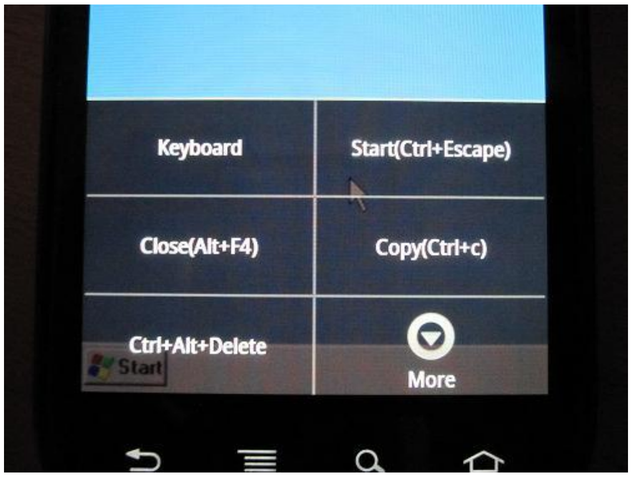
If you scroll down to the bottom of the menu options you will see something like this:

- **State** gives you some interesting state of the application that you may or may not be interested in.
- **Key Definition** and **Delete Key** are described in the next section.
- All the ones with **trace** in them have to do with tracing and problem reporting (see later).
- **Set Debug on/off** is something that should only be used when problem reporting since it slows things down a lot.
- The others should be self explanatory**.**

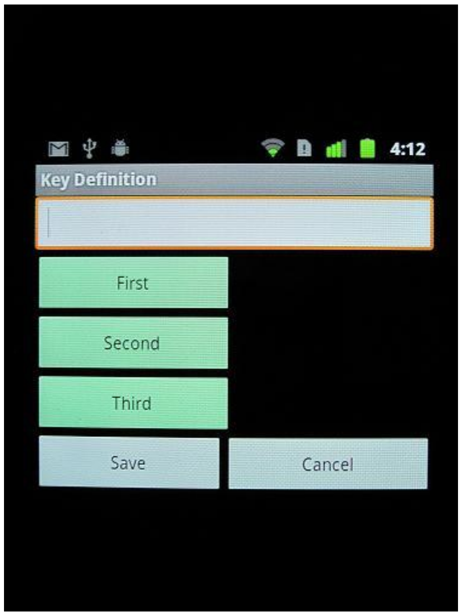
Defining Keys

How do you enter certain key combinations on a standard PC keyboard from your device? We could define some funky mapping (e.g. well this key is the ESC key and you get the F1 key by some other contortion of your fingers) We've decided that adds too much complexity. So you enter those funky keys simply using the Menu. We’ve put some of the standard built in combinations (e.g. Copy, also know as Ctrl+c) and then created a facility for you to define your own funky key combos that will then also appear on the menu. To further ease of use, we’ve made the keys, use count sensitive. Which means the more you use them the closer to the top of the menu they will appear and the less menu scrolling you will have to do to use them.

- The built in keys are at the top of the menu. They include the usual suspects sorted by name.

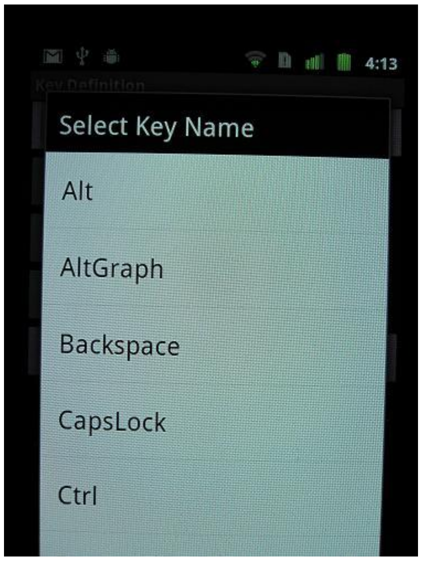


- The user defined keys are managed using the **Key Definition** and **Delete Key** menu items mentioned above. The **Delete Key** is self explanatory: it lets you delete your key definitions (if any). The **Key Definition** is where the action is. It gives you something like this:

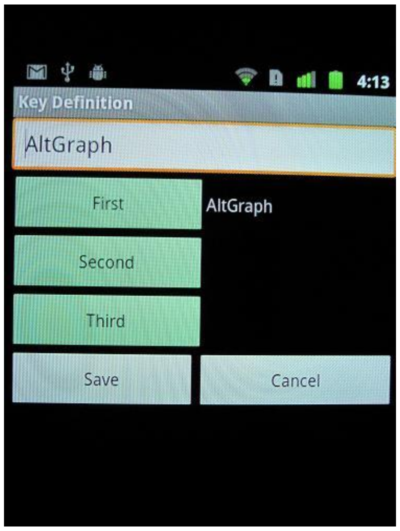


There are some things to note about this:

- There are a maximum of three keys that make up a combination, appropriately called **First**, **Second** and **Third**. How original!
- Clicking one of these buttons allows you to select a key from the standard PC keyboard. For example:



- The **Name** field is automatically chosen with the default name being the names of the individual keys separated by “+” signs. You can however choose your own name for this combo. After selecting your key combination the screen can look like:



- Press the **Save** button and you now have your own combo that will show up on the Menu bar.

Licence stuff

The RDP protocol allows the optional use of licences during its licence negotiation phase. Not every server does this. In essence: when the server makes use of licences, the server requires the client to supply a licence in order for the connection to proceed. Some things that are useful to know:

- If the server does optional licence negotiation, a licence is acquired for the duration of the session.
- When the session is finished (For example when you log off or forcibly close the app) all licences are removed from the device. This makes sure unnecessary data is not left on the device.
- You can manage acquired licences during your session using the menu “**List licences**”, “**Delete all licences**” or “**Delete a licence**” options. The licences are named like the host. Note: if you have acquired no licences, these menu options will not appear - naturally.

Sound/Audio: *Audio Output Virtual Channel Extension*

The RDP protocol allows audio to be rendered on the local device. The Android device will then try and play the audio it receives on the local device. Some things to be aware of:

- the format the audio data takes is dependent on what the server chooses. The supported audio formats include PCM (*Pulse Code Modulation*), ADPCM (*Adaptive differential pulse-code modulation*), ALAW (*A-Law telephony companding algorithm, from ITU-T G.711*) and MP3(*MPEG Layer 3*)
- In our experience this is most useful for the short sounds produced (standard logon

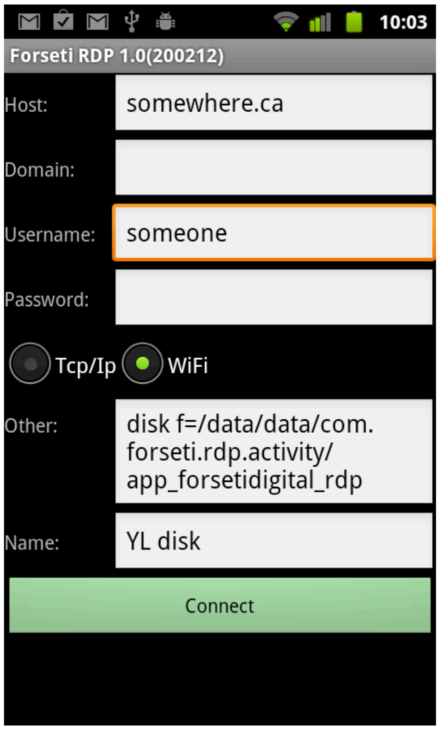
sound, clicks etc.) Playing music audio requires that the server & network provide the data to be played at sufficient rate. e.g. in our testing of playing an mp3 music track the server chose ALAW (why this instead of MP3 we know not!!) at 88200 bytes/second. **If sound data is not delivered at this rate you will notice hiccups in the sound**.

- In our testing, playing a music track for about 3-4 minutes sends from 50 to 100 short sound files and 7 - 13 Megabytes of sound data! You have been warned/illuminated!
- There are a couple of commands available via the “other” field on the connect screen for sound specific issues (these are entered without the quotes!):
    - “sound=true/false” allows you to turn sound on/off for this connection. The default is true.
    - “debugs=true/false” enables you to turn on sound specific tracing as the various sounds are played. The default is false.
- You have to configure RDP on the server to enable audio to be sent to the client. For example we have found that even though the log on sound plays, the log off sound does not play even though the log off screen says the sound is playing! This is not a fault of the app. There is no audio data being sent to the device. Perhaps checking the server RDP settings can help. Good luck!

Data integration: *File System Virtual Channel Extension*

The RDP protocol suite offers a whole suite of what are known as virtual channels. One of those is used for audio (see above). Another one is called *Remote Desktop Protocol: File System Virtual Channel Extension*. In essence this allows you to expose some portion of your device (phone/tablet) disk storage as an external drive to Windows. This allows an amazing level of integration between your Windows system and your device: move files back and forth using standard Windows tools like Windows explorer; print device local files on Windows attached printers; etc. etc. In essence, treat your local device file system as if it was local to your Windows machine.

To do that you need to: define the portion of your disk space you want to expose and choose the drive letter. You specify that in the “Other” field of the connect screen. In the example below drive “f” is mapped to the directory on the local device called “/data/data........”
using the syntax “disk <drive letter>=<local directory specification>” Use this same syntax for each “directory” you want to expose. Most often you only need one such definition but you never know!



10:03

Forseti RDP 1.0(200212)

Host: somewhere.ca

Domain:

Username: someone

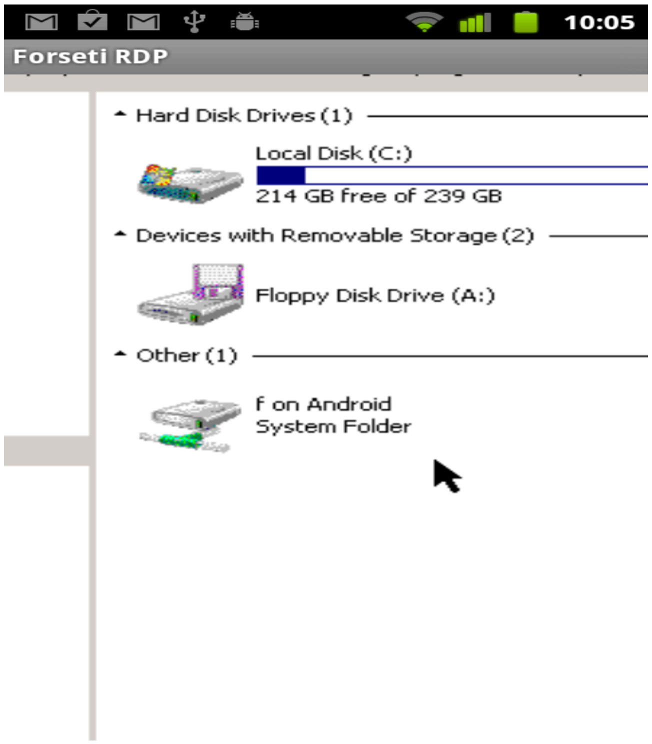
Password:

Tcp/Ip [✓ WiFi]

Other: disk f=/data/data/com.forseti.rdp.activity/app_forsetidigital_rdp

Name: YL disk

Once you connect you will see your local drive seamlessly integrated into your Windows environment as the screen shots below illustrate. Notice the “f on Android”



10:05

Forseti RDP

Hard Disk Drives (1)

Local Disk (C:)

214 GB free of 239 GB

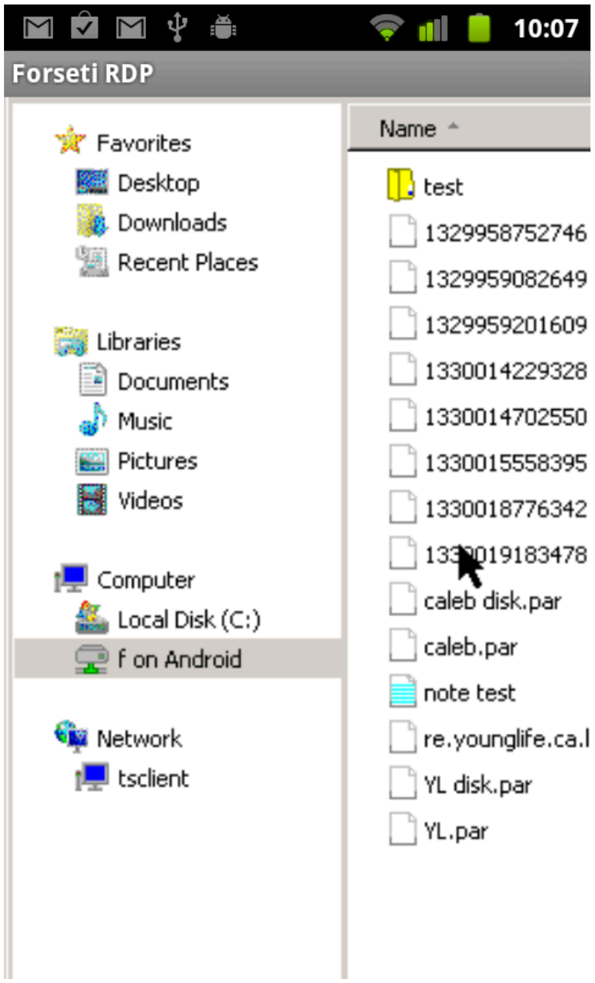
Devices with Removable Storage (2)

Floppy Disk Drive (A:)

Other (1)

f on Android
System Folder

Double click on drive “f’ and voila - the files on your local device are visible on Windows.



<u>TS Gateway integration:</u> *Terminal Services Gateway*

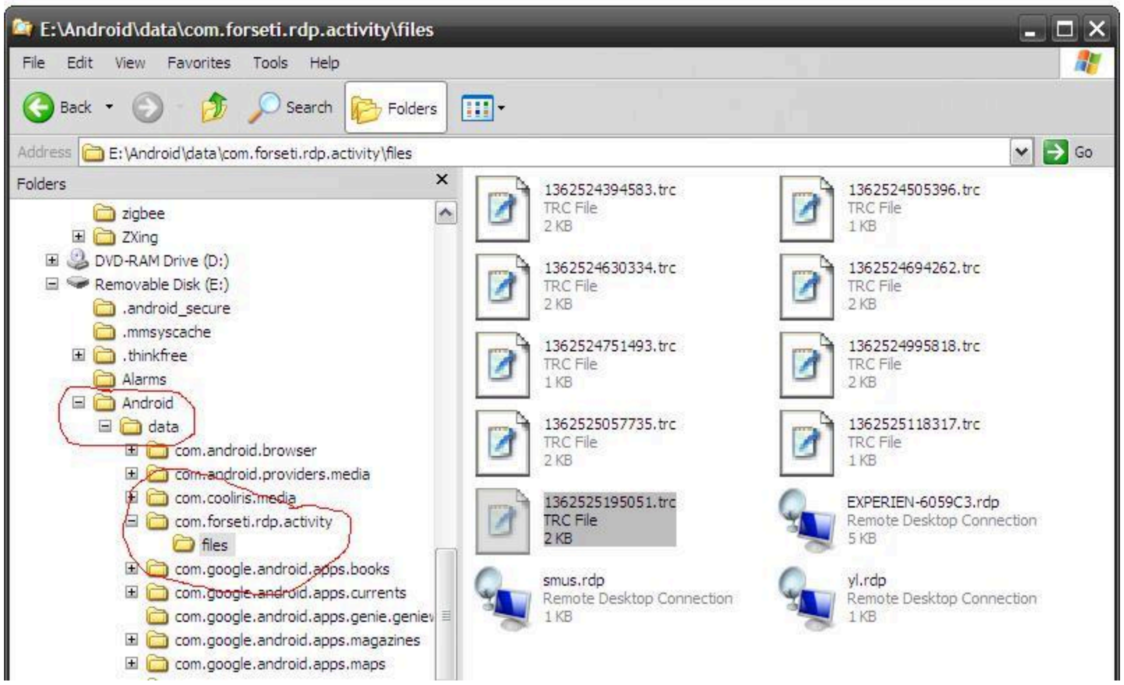
Often Windows systems are not directly accessible but instead accessed through a gateway. This provides even greater levels of security beyond the normal RDP security. Connecting this way is a snap. Simple specify "gateway=mytsgateway" in the "other" field on the connect screen. For example, if your TS Gateway is at "mygateway.abc.def" simply have "gateway=mygateway.abc.def" in the "Other" field. Of course without the quote signs!

<u>.rdp file support and Trace file retrieval:</u>

Remote desktop protocol settings on Windows systems are often contained in an .rdp file. It is often convenient to use the same file for both your Windows environment and your Android environment. The protocol settings are described by Microsoft here.

- The .rdp elements supported are:
    - **address:s:xxxx** is the same as "host=xxxx"
    - **username:s:username** is the same as "user=username"
    - **domain:s:mydomain** is the same as "domain=mydomain"

  - **gatewayhostname:s:gatewayhost** is the same as specifying “gateway=gatewayhost” in the other fieldon the connect screen. This is as long as **gatewayusagemethod:i:1** is also specified.
  - **MCSPort:i:n** is the same as specifying “port=n” in the other field on the connect screen.
- To place .rdp files and to be able to access trace files generated, one must look at the standard application public data directory. This is located in the **Android/data/com.forseti.rdp.activity/files** directory. This directory path is created if necessary when the app runs for the first time.
- The easiest way to access this directory is to make your USB storage available on the device when it is attached to your personal computer. For example on Windows one can use Windows explorer to drag and drop files to and from this directory. Use this in much the same way you would use to put music onto your device. **NB**: some USB drivers are known not to display ALL the directories (e.g our current Motorola Xoom). However a free file explorer such as *ES File Explorer* will show that the above directory exists!
- The picture below illustrates an Android device with three .rdp files and a bunch of trace (.trc) files.



E:\Android\data\com.forseti.rdp.activity\files

Address: E:\Android\data\com.forseti.rdp.activity\files

```
zigbee
ZXing
DVD-RAM Drive (D:)
Removable Disk (E:)
  .android_secure
  .mmsyscache
  .thinkfree
  Alarms
  Android
    data
      com.android.browser
      com.android.providers.media
      com.cooliris.media
      com.forseti.rdp.activity
        files
      com.google.android.apps.books
      com.google.android.apps.currents
      com.google.android.apps.genie.geniev
      com.google.android.apps.magazines
      com.google.android.apps.maps
```

```
1362524394583.trc  TRC File  2 KB
1362524505396.trc  TRC File  1 KB
1362524630334.trc  TRC File  2 KB
1362524694262.trc  TRC File  2 KB
1362524751493.trc  TRC File  1 KB
1362524995818.trc  TRC File  2 KB
1362525057735.trc  TRC File  2 KB
1362525118317.trc  TRC File  1 KB
1362525195051.trc  TRC File  2 KB
EXPERIEN-6059C3.rdp  Remote Desktop Connection  5 KB
smus.rdp  Remote Desktop Connection  1 KB
yl.rdp  Remote Desktop Connection  1 KB
```

Reporting Problems

Problems? What problems?! We’ve been around the block long enough to know that with a complex protocol like RDP and varieties of devices and networks, things can and will go wrong. So we’ve designed this app so that you can get the information to us that will help us fix the problems and make life better. Instead of just ranting to support@forsetidigital.com you can do some or all of the following:

- The application always automatically saves its current trace file when it ends abnormally. So the easiest first step is to send us this file if you think the app ended abnormally. Of course the more detail the better so the debug options below are even more helpful for us.
    - ***The easiest way*** to send this trace to us is to select the menu "**Save & email trace on exit**" prior to connecting. This will attempt to email the trace directly to Forseti support. You will be asked to approve the sending of the email. A followup email with other info including useful info in the device's event log will be very helpful
    - One can also use the method above (versions 2.7 and above) to get the .trc files from the device when attached via a USB cable.
- Use the **State** menu option and record what it says there. By the way there's some generally useful info there about how much data you have sent and received etc.
- If you can narrow down where the problem occurs and can easily reproduce it, write down the steps necessary to do it.
- Since giving us your access credentials will probably not happen, and even if you did we might not be able to access the machine anyway. You can help us by giving us as much detail about the situation. This involves send us a trace file. Use the following steps.
    - get to the place in your work where the problem occurs.
    - Using the menu enable debug mode using **Set Debug on**.
    - Do what causes the problem.
    - Turn debug mode off using **Set Debug off**. This avoids cluttering the trace with too much information.
    - Save the trace using the menu **Save trace** option. NB: if the error is of a nature that causes the application to terminate (oh dear we are sorry), then at the point you **Set Debug on**, also enable **Save trace on exit**. This will make sure something gets saved.
    - Get the trace file. This involves using the **View trace** menu option. Trace files are produced with a timestamp. View the trace file of interest. The real file name is at the start of the trace. Get this file off your device and send it to us in the email.
    - We hope you never have to do this but we thank you in advance for taking the time to make our product better.

- bundle this info up and email it to us. Thank you for taking the time to do this.

Languages supported

The particular language context for your device is called the Locale. For example if your language is English and of the US variety, the Locale would have a string representation of en_US.

If that's too techie a concept, the basic idea is this: you want your RDP connection to present itself the same way as your device does. If you are a Danish speaker setting up your device to be like you - Danish - you want your RDP session to know you have a Danish "keyboard".

Here's the list of Locales currently supported:

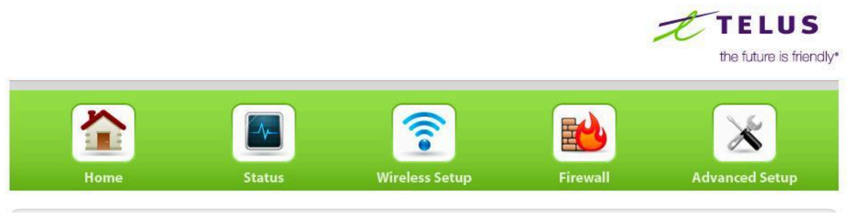
- Arabic - Saudi Arabia
- Danish
- German - Germany
- English - United Kingdom
- English - United States
- Spanish - Spain
- Finnish
- French - France
- French - Belgium
- Croatian
- Italian - Italy
- Japanese
- Lithuanian
- Latvian
- FYRO Macedonian
- Norwegian (Bokmål)
- Polish
- Portuguese - Portugal
- Portuguese - Brazil
- Russian
- Slovenian
- Swedish
- Turkish
- Turkmen

How cool is that!

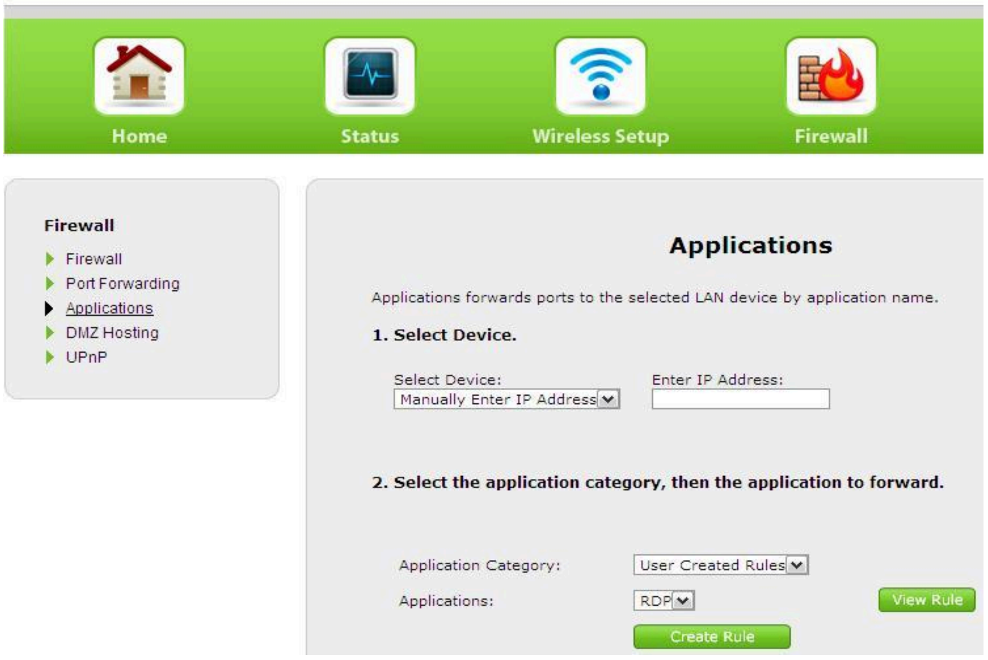
Helpful hints (or FAQ)

Some useful information that might come in handy.

- Software versions. Our app internal version you will see in the "About" menu, in the trace file output and on the initial connect screen. Our "version" is a combination of a version and a build date (the date we last generated a new version of the app)
- Connection issues during log on. If you find that you get connected and then get

disconnected while trying to log on you can:
    - Check the automatically saved trace file (and ideally send that to us please). If you see a message like “Connection terminated: No information available” or “Connection terminated: deactivated” these are normal disconnect messages sent from Windows.
    - In order to dig into why this is happening you might have to look at the Windows. Start->Control panel->Administrative tools->Event Viewer. If you find a reason there send that to us and that might help us pin down whether this is an app issue or a Windows issue.
- Multiple logons via RDP. If your Windows machine is a server then you can have multiple simultaneous users (e.g. see http://support.microsoft.com/kb/814590). If you don’t then alas you are stuck with having to log out first or force the log out. You will a screen to this effect when you connect. This is a Microsoft limitation for non Windows server boxes. Mr. Google indicates a number of “options” (e.g. see http://social.technet.microsoft.com/Forums/el-GR/w7itproinstall/thread/41e9e500-714a-443b-bff2-55f0d500d3d1) All we can say is: user beware :-)
- Sometimes it appears that you should specify a domain in the domain field. e.g. user=myid domain=MYDOMAIN We have found that if that does not work, leave the domain field blank and instead specify user=MYDOMAIN\myid
- Getting through firewalls & routers.
    - This article is useful for both setting up RDP on XP and configuring your home router for access: http://www.online-tech-tips.com/windows-xp/how-to-setup-remote-desktop-on-windows-xp/
    - If your machine is behind a wireless router (for example at your home) you will need to enable TCP access on port 3389. Consult your router documentation or carrier for how to do this. See the example below for the general idea.
    - In addition you might have to do the same sort of thing on the machine itself. For example, if you are running Zonealarm it will block this kind of access so you might have to stop it (or buy the non-free version which would allow this selective enabling!)
- Example enabling of a home router: Actiontec V1000H supplied by Telus (ISP provider)
    - from your vendor get access to the router login with id & password. e.g. http://192.168.1.254 in my case
    - you’ll see a screen like:
    - 

    - Click on firewall and get to the applications tab (you could maybe also do this with port forwarding I suspect)

- 

Applications

Applications forwards ports to the selected LAN device by application name.

1. Select Device.

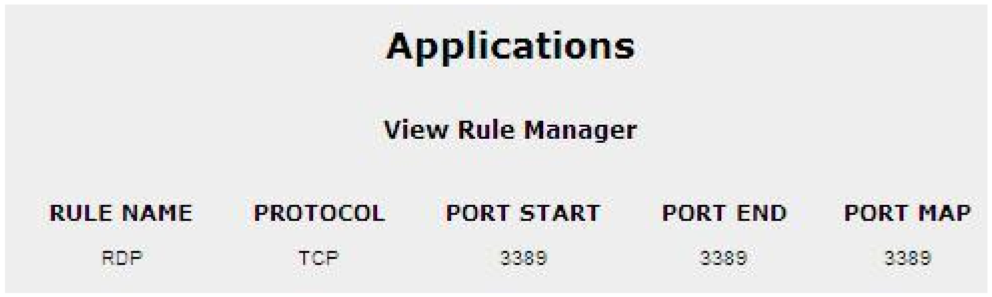
Select Device: Manually Enter IP Address

Enter IP Address:

2. Select the application category, then the application to forward.

Application Category: User Created Rules

Applications: RDP

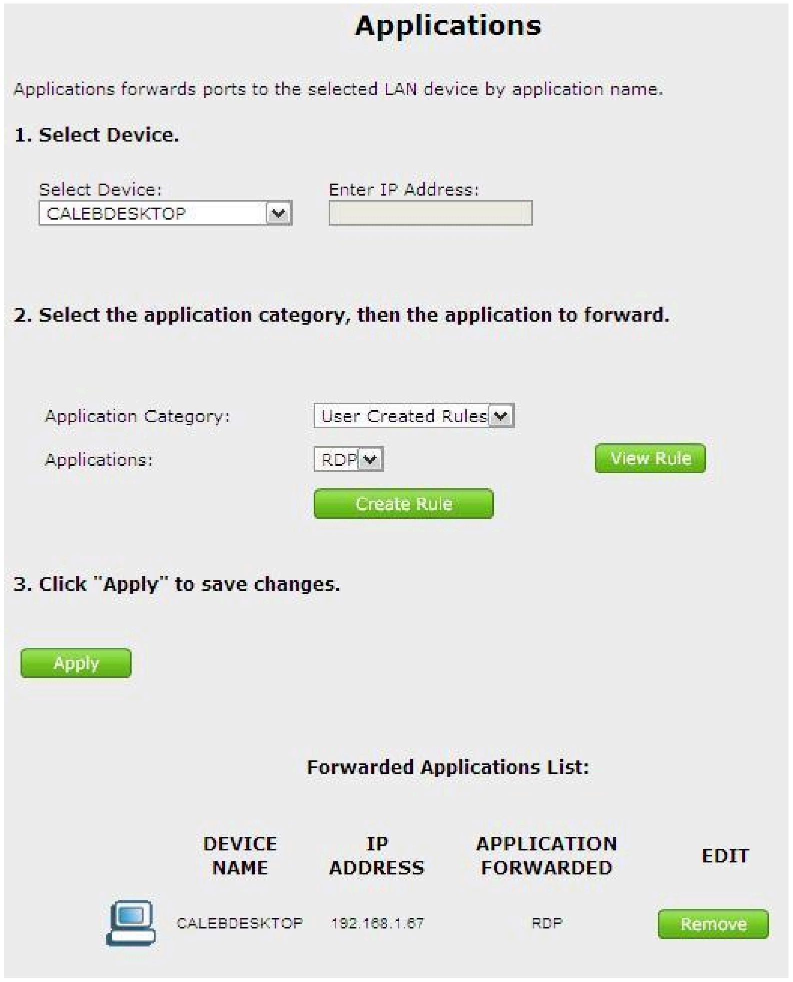
- Check the Application category for an Application predefined for RDP. I didn't find one so I create a new one using the "User Created Rules" one. It looks like this.
- 

Applications

View Rule Manager

RULE NAME	PROTOCOL	PORT START	PORT END	PORT MAP
RDP	TCP	3389	3389	3389

- Give it a name - RDP, select the protocol TCP and have the same port number in the Port Start, End and Map. 3389 is the TCP port that the protocol uses.
- Security tip: if you want to make things more opaque you can use port mapping which involves selecting a port on the router (e.g. 65432) and mapping it to 3389. If you do this then you will need to use port=65432 in the "other" field of the connect screen.
- Once you have a rule then bind it to the computer on your LAN of interest. Here I select Caleb's desktop (which is its Windows machine name). Click Apply in all the right spots.The screen then looks like:



Applications

Applications forwards ports to the selected LAN device by application name.

1. Select Device.

Select Device: CALEBDESKTOP

Enter IP Address:

2. Select the application category, then the application to forward.

Application Category: User Created Rules

Applications: RDP

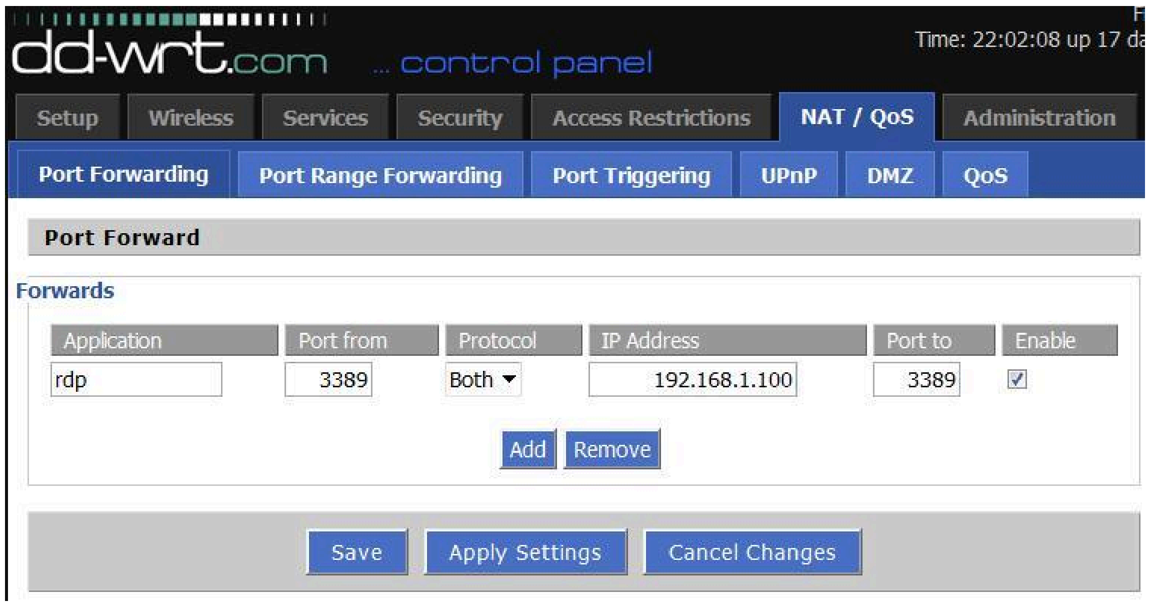
3. Click "Apply" to save changes.

Forwarded Applications List:

DEVICE NAME	IP ADDRESS	APPLICATION FORWARDED	EDIT
CALEBDESKTOP	192.168.1.67	RDP	Remove

- Now the router is setup.
- From the machine you enabled (in my case Caleb's desktop) you need the public IP address. You can do via (a) going to http://www.whatismyip.com/ on that computer or (b) using the router's public IP address (from the router home page)
- This is the address to enter in the Host field of the connect screen. Note: if the router's address changes (e.g. its DHCP renews with a different address) you'll need to adjust this.
- In my case the final step was disabling Zone Alarm (firewall) on the desktop since that was blocking RDP port access.
- Even though routers differ, the basic principles apply. With Mr. Google's help you should be able to adjust your home/business router in a similar way!

● Another example for a dd-wrt router.
  - You can use the stock setting on the router except for the port forward found in NAT/Qos -> Port Forward
  - Set something up like:

- ○ 

Port Forward

Forwards

Application	Port from	Protocol	IP Address	Port to	Enable
rdp	3389	Both	192.168.1.100	3389	✓

- Licence issues: as noted above in the Licence section, if you suspect you are having troubles related to licences, delete the specific licence or all of them and try again.
- One important encouragement: if you have troubles and can provide even just a publicly accessible IP address for us to get to and test, that will make a HUGE difference in getting to the root of your troubles. We realize that sometimes this is not possible but if you can provide it (with even a time limited guest account!) so much the better for all of us.

BlackBerry PlayBook FAQ

The current PlayBook app has been tested on the Beta 2.0.0.6149 OS release of the Android Runtime. The features that do not translate/work well have been customized specifically for the PlayBook version of the App. The following are helpful things to know about the PlayBook version.

- You need to be running the Beta 2.0.0.6149 OS on your PlayBook. You can start this process from here: https://bdsc.webapps.blackberry.com/android/beta/bbtablet20/register/
- Once this is installed you are ready to run the app. If you have not installed this release your app will not launch.
- **Sometimes we have noticed the first initialization of the Android runtime claims to have trouble** (this can happen when you run the App for the first time if this is the first Android app to run on the PlayBook). Whether this is a bug or feature of the Beta release we do not know! Do not be alarmed. Subsequent attempts go very smoothly.
- Since there is one networking option on the device, the connect screen does not offer the choice of Wifi/Networking illustrated above. This makes this screen simpler.
- The standard PlayBook gestures apply. For example, swiping from the top border down,

exposes the context specific menu - ours.

- You do not need to use the “show keyboard” menu item. The existing PlayBook gesture for bringing up the keyboard (swipe diagonally from bottom left of the tablet) works equally well.
- The standard pinch/zoom, tap, long tap, double tap, move all work. The long press gesture which translates into a Windows “right click” also is there. As noted above, there is a menu option “Right Click” which you might prefer.
- There is no audio support. We have not been able to get this to work well enough to include in the app at this point.
- Menu items that are not enabled in the PlayBook version include:
    - “Save & email trace on exit” There is no email support to capture the Android email intent.
    - Similarly the “Help” menu (which requests a viewer to open this online doc) does not work. Instead consult the online doc via the PlayBook browser directly!
- If you have trouble with the app you might want to do a complete restart of the PlayBook. i.e. power it down and up. This will re-initialize the tablet including the Android runtime.
- On termination of the connection the App does not appear to go away but minimizes. We suspect this is a “feature” of the interraction between the Android runtime on QNX. Simply close (click the “x”) the window.

We hope you enjoy this application and that it adds value to your life.

We are changing the app to fix issues and address user’s requests. Check out the release notes here: Release Notes

You can be in touch with us at support@forsetidigital.com