1. **Log Files**

1.1 Location of Host Log Files

1.2 Location of Viewer Log Files

1.3 Structure of log file lines

1. **Communication Architecture**

2.1 HTTP & Streaming Communications

2.2 GoToMyPC & Proxies

2.3 Structure of log file lines

1. **Connection Wizard**

3.1 Connection Wizard Architecture

3.2 Connection Wizard Troubleshooting

1. **GoToMyPC Host Communications Out of Session**

4.1 Connection Test

4.2 Connection Test Troubleshooting

4.3 Persistent Connections

4.4 Get Options

4.5 Connection Probing

1. **GoToMyPC Starting a session**

5.1 Overview of communications

5.2 Troubleshooting: Determining whether the problem is at the host or the client

5.3 Troubleshooting: Client Connection problems

5.4 Troubleshooting: Host Communications to the Comm Server

1. **In session communications**

6.1 In session communication architecture

6.2 Troubleshooting in session communications

1. **Remote Printing**

7.1 Architecture: GoToMyPC Universal Printing

7.2 Troubleshooting: Missing GoToMyPC Printer

7.3 Troubleshooting GoToMyPC Print Jobs don’t print out

7.4 Architecture: Native Printing

7.5 Troubleshooting: Native Printing

1. **In session features**

8.1 Architecture: Screen Blanking

8.2 Troubleshooting: Screen Blanking

8.3 Troubleshooting: In session features unavailable

8.4 Troubleshooting: File Transfer

8.5 Troubleshooting: Remote Sound

8.6 Troubleshooting: CTRL ALT DEL

1. **Host computer offline**

9.1 Troubleshooting host offline

1. **Direct Connections**

9.1 Direct Connections Architecture

9.2 Troubleshooting: Direct connections unsuccessful in the same network

9.3 Troubleshooting: Direct connections unsuccessful in different networks

1. **GoToMyPC Installation**

10.1 Architecture: GoToMyPC Installer

10.2 Troubleshooting: Installation

# 1 Log Files

**1.1 Location of Host Log Files**

Every time the GoToMyPC software starts up, it creates two log files. It makes a file named g2svc.log and a file named g2host\_00.log. The file named g2host\_00.log is the one that we are interested in because this file has information that is critical for troubleshooting.

Windows 2000/Windows XP

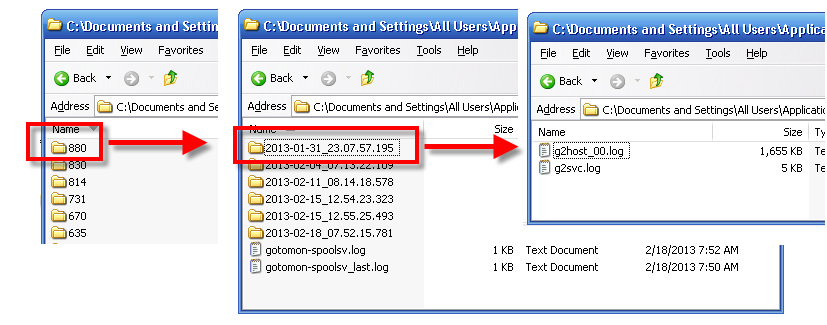
**%allusersprofile%\application data\CitrixLogs\GoToMyPC\**

Windows Vista/7/8

**%ProgramData%\CitrixLogs\GoToMyPC\**

GoToMyPC organizes the folders by build number. Within each build’s folder, the log files are organized into subfolder’s containing the host log and g2svc log. Most of the information that’s used to troubleshoot is contained in the host log. The g2svc log may also be used to track GoToMyPC for other events like user log off, log on, CTRL ALT DEL, or shut down events.

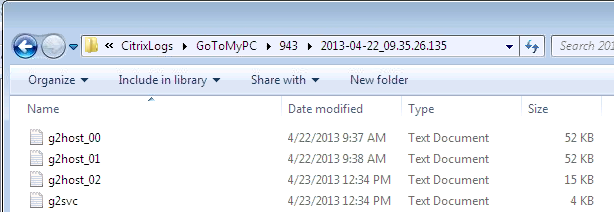
Starting with build 622 the subfolder is named based on the date and time that the GoToMyPC folder started.



You might notice that sometimes the folder where the host log file is supposed to be only has the g2svc.log. There is a recurring problem where sometimes the host log file ends up in the GoToMyPC folder instead of the build folder. In this case, you will find the file in the GoToMyPC folder instead of the build folder. They can get stacked up pretty far and so you’ll have to use the date modified to make sure that you have the correct file.

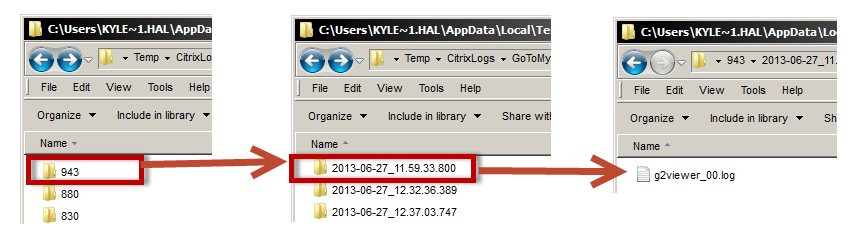


If you see multiple log files within the same folder, this usually happens because some part of the GoToMyPC program crashed and GoToMyPC had to relaunch itself.

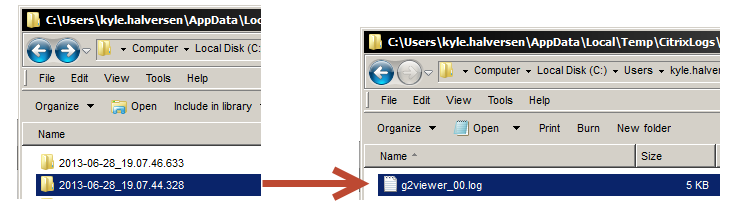


**1.2 Location of Viewer Log Files**

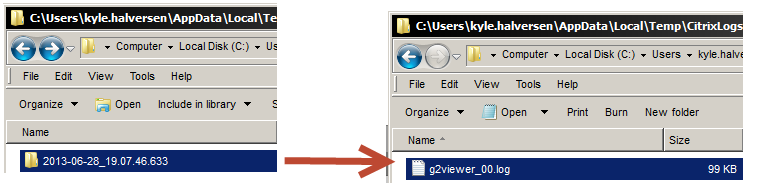
Every time the GoToMyPC viewer software is launched a log file is created. The viewer organizes its log files by build number. Each build number has its own folder and within this is a subfolder that has the viewer log file in it. The name of the subfolder is the timestamp of when it was created.



A minor annoying bug is that if you use a desktop shortcut, two viewer files are created, each with their own folders. The first log file describes the process of the GoToMyPC shortcut program “g2quick” calling up the session. Once the session is started, the log file ends.



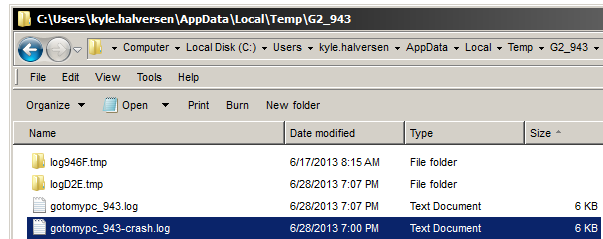
The next log file actually describes what happens during the session:



Sometimes the GoToMyPC viewer software experiences a crash when opening. The crash may happen before the software can create a log file in the CitrixLogs folder. You can find this crash in

%temp%\G2\_<build number>

For example: %temp%\G2\_943



**1.3 Structure of Log File lines**

The first thing to understand about log files is that they are a bit chaotic. There are multiple parts of the GoToMyPC software that run at the same time. Each part of the GoToMyPC software logs information as things happen and they talk over each other.

2013-04-17 17:24:53 PST i: [g2audioh] <G2AudioThreadH> AudioHost::initialize() - Skype installed=false

2013-04-17 17:24:53 PST i: [g2mainh] <G2HostAgentThread> CPluginProxy::CPluginProxy() - G2Printing

2013-04-17 17:24:53 PST i: [g2comm] <CPluginManagerIpcAdapter::getPlugin()> Creating process: "C:\Program Files (x86)\Citrix\GoToMyPC\g2printh.exe" "StartID={CC79D7F8-2A04-4B81-AE23-8AB2993EC440}&Debug=Off&Stat=On&StatDb=Off&Index=0"

2013-04-17 17:24:53 PST i: [g2audioh] <G2AudioThreadH> ReceiveThread::connect\_()

2013-04-17 17:24:53 PST W: [g2comm] <948/ISession.newChannel> CSession::\_createPingChannelMgr() - ping channel already exists.

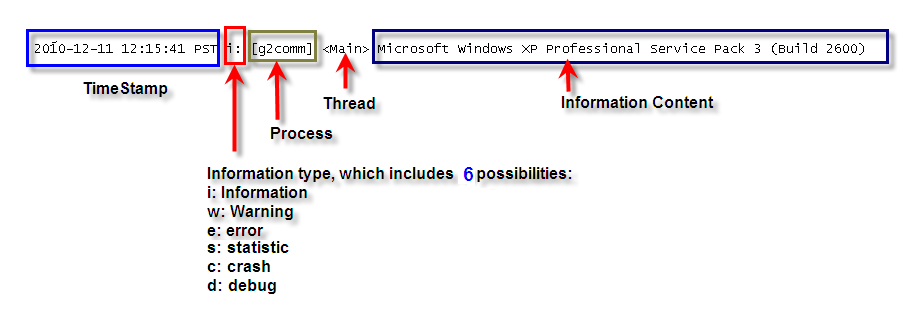
2013-04-17 17:24:53 PST i: [g2comm] <AU\_hiWR13881500> Reuse flag set to true

2013-04-17 17:24:53 PST i: [g2comm] <AU\_hiWR13881500> JEDI connect: Creating SSL socket

2013-04-17 17:24:53 PST i: [g2comm] <SS\_hiRD13658516> SS Retransmission: drain sees SYN packet

2013-04-17 17:24:53 PST i: [g2host] <G2PScreenCaptureThread> CChannelUtils::createAndJoinChannel() - ScreenSharing Channel joined successfully.

In the sample sbove, within the same millisecond, our sound plugin is checking to see whether or not Skype is installed to help it figure out what codecs are available. G2Comm is trying to call up the process for printing. G2Audio is trying to open up a communication channel to the comm server for remote sound and g2host has managed to open up a channel for screen sharing. All of these processes are logging entries at the same time. To make sense of it, we want to look just one line and figure out how the information is organized.



**TimeStamp**

Just like it sounds, the timestamp tells you when the information was written to the log file. The log file is always kept in Pacific Standard Time. It does not adjust for the user’s time zone and it does not adjust for daylight savings time. It’s important to keep your times straight when you are trying to match up issues to specific lines in the log file.

**Information type**

Information in the log files is divided into certain types, represented by a letter followed by a colon

**i:** Information – this is the most common type of information to be found in a log file. This line just gives you some information about a particular event. The hard thing is that it’s not always clear what information relates to.

**w:** Warning – this line is to draw attention to the fact that the software has run into something that it did not expect. Warnings are usually precursors to errors and sometimes crashes and may give an indication so to what the cause was.

**e:** Error – This means that the software encountered an error with what it was attempting to do. Sometimes an error isn’t what you are expecting.

**s:** Statistic – Just like it sounds this line is used to keep performance statistics. Across all products, statistics keep track of Fast Failover (FFO) which is the process by which our products calculate and can switch over to different Internet Service Providers to reach a Citrix Server for a session. We will talk about FFO in more detail later.

Examples:

**c:** Crash lines in the log file literally mean that some part of the software experienced a crash. This is not the same as an error line in the log files. Crash lines mean that some part of the software has run into a problem that it was not expecting and had to close. Usually you will see a dump of the data that was loaded into memory and you may see the modules that the software loaded during the crash

**d:** Debug lines are only visible when you have turned on debugging for the account the GoToMyPC software is setup under. This is to make you aware that this line includes additional information that would not otherwise be visible. For example if you were troubleshooting

**Processses**

Each line in the log file is reported by a particular process or exe file of the GoToMyPC software. Some of these are very straightforward. G2printh handles printing related things and g2filh handles file transfer related things.

Processes that run when a user is not connected to a host computer

G2Svc – This is the GoToMyPC service file. This file is responsible for launching the rest of the GoToMyPC software. You will not see many entries from the g2svc in the host log files folder

G2comm – This process does the heavy lifting in terms of opening communication to Citrix servers. It is also involved in launching other components of the software.

G2pre – This process is the prelauncher for GoToMyPC. It calls up g2tray

G2tray – This is the part of the GoToMyPC software that the user interacts with. It’s responsible for the system tray icon and any prompts or information that is visible to the user on the host computer.

Processes that run once a user connects to a host computer

G2host – This is the part of GoToMyPC that is responsible for screen sharing

G2print – This is the part of GoToMyPC that’s responsible for printing. It attempts to intercept

G2mainh – This is the part of the GoToMyPC software that is responsible for screen capture

G2simpleft – The name means simple file transfer. When connecting to a Windows host computer from a Mac, g2simpleft (not g2fileh) handles file transfer. G2Simpleft only runs

Processes that run when they are needed in session

G2fileh – This file is responsible for file transfer. It is called up whenever a file transfer is initiated and closes when

Processes that run on the client Computer

G2Viewer – This is the only process on the client computer. All the log file lines on the viewer will show g2viewer. This process handles all of the different parts of the viewer software.

# 2 Communication architecture

**2.1 HTTP & Streaming Communications**

There are two kinds of requests that GoToMyPC sends out, HTTP Requests and JEDI Requests.

**HTTP Requests** - HTTP is a standard communication protocol that you use all the time when you open up a web browser and go to a website. Other applications besides browsers make these requests. When a request to a server is sent using HTTP protocol, the server will receive the request, respond to the client that made the request and then the channel will be closed. GoToMyPC uses HTTP to send short specific requests. Some of these include requests to start and end sessions, the connection test, authorization for and use of a desktop shortcut, etc.

**JEDI Requests** – JEDI is Citrix’s proprietary communications protocol. GoToMyPC uses it for sending real time data which includes screen sharing, keyboard and mouse input, file transfer data, printing, etc. A JEDI connection is similar to an HTTP request, but it stays open for an extended period of time.

**8200** – The JEDI request is left untouched

**443** – The JEDI request is wrapped up in SSL (Fake SSL in GoToMyPC’s case)

**80** - The JEDI request is wrapped up in HTTP (Also known as HTTP Tunneling). This was mainly available for use with older proxies that required all communications to use the HTTP protocol. This restriction is virtually nonexistent today.

Why do we like port 8200?

If port 8200 is allowed, we don’t have to do any kind of modifications to the JEDI request

If we use port 443 we wrap up our communications in SSL. GoToMyPC wraps up its communications in what I’ll call “Fake SSL”. By that I mean that the SSL encryption it uses to encrypt packets does not adhere strictly to standard SSL protocol. GoToMyPC communications are encrypted, but the SSL encryption that communication is wrapped up in when it uses port 443 is not real SSL encryption.

With port 80, all of the JEDI request have to be wrapped up into HTTP protocol. This increases the size of the packets. This means that we have to more requests to capture transmit the same amount of information. This causes a slight performance hit.

**2.2 GoToMyPC & Proxies**

What’s a proxy? Proxies can get a bit complex and so before we talk about them, let’s go through a few examples. Your internet configuration is that you’ve got a cable modem and the Ethernet cable is plugged directly into your computer. You want to get to google’s website. To get to google’s website, a request is sent from your computer directly over to google’s server at 74.125.128.138 and google’s servers respond back.

Here’s a more complex setup.

You’re in a corporate environment where all of the computers are connected to a network. You want to get to google’s website and so a request needs to be sent from your computer to google’s servers. Your company has placed restrictions on the network. You can’t send a request directly from your computer to google’s servers. Instead your request must be sent to the company’s server first. Your company server will examine the traffic and if it allows it, the request will be forwarded to google and google’s response will be forwarded back to you.

In the more complex setup, the company server that we’re talking to is a proxy. Proxies help establish connections to different servers in a restricted environment so you don’t want to just try and get around them. Instead it’s good to try and use them.because they “know” how to navigate out to the internet from a company’s firewall.

**Explicit HTTP Proxies**

**Transparent HTTP Proxies**

**SSL Inspecting Proxies**

**Explicit HTTP Proxies**

An explicit HTTP proxy has a specific IP address and a port. A specific port and IP address would be 192.168.1.1 on port 8080. If a client wants to talk to a server, the client first establishes a connection to the proxy and then asks the proxy to send the request to the server. Listed below is a line where the host software sends a request to the poll server through the HTTP proxy.

2012-11-30 12:31:35 PST i: [g2comm] <cda1> Verified: HTTP 66.151.158.177:80 [192.168.1.1:8080(HTTP)], method=Proxy stage=Manual source=[U]

Client sends a request to 192.168.1.1 on port 8080. Once connected to the proxy, the client tells the proxy to send the request to gotomypc.com. With explicit HTTP proxies, we first establish an HTTP connection to the proxy. Then for streaming communications we’ll send JEDI requests in a fake SSL wrapper directly to Citrix servers instead of going through the proxy. We don’t send a request that uses the JEDI protocol directly to a proxy server. The proxy server will not understand what JEDI protocol is and will probably block the request. Instead what GoToMyPC does is it wraps up the JEDI request in SSL encryption. A proxy server doesn’t know what JEDI is, but it knows what SSL is and when it sees SSL requests it understands that it’s an encrypted request and will normally forward it along.

**Transparent HTTP Proxies**

Transparent proxies, like the name suggests are running in the background, but they are completely invisible to the user. The client doesn’t even need to be aware of the proxy at all. A transparent proxy is different from the explicit HTTP proxy. It doesn’t have a specific IP address and port like the transparent proxy does. Instead whenever the user makes a connection to the internet the proxy will intercept outgoing connection request. The transparent proxy pretends to be the target server and accepts the connection request. The proxy then reads the request and will either forward it to the real server or reject it. Most transparent proxies are configured to only allow HTTP and SSL requests. To get a better understanding of how it works, you can refer to the log lines below to see how it behaves. In this situation GoToMyPC is connecting for the first time. It ties ports 8200, 443, and 80. The HTTP communications were successful on all three ports. GoToMyPC also didn’t find any other proxies in the user’s networks.

2013-05-18 23:30:46 PST i: [g2comm] <cda1> Detect: HTTP PING to 66.151.158.177:443 (proxy: None=0.0.0.0:0<ip>) [\_] stage 2/2 >OK<

2013-05-18 23:30:46 PST i: [g2comm] <cda1> Detect: HTTP PING to 66.151.158.177:80 (proxy: None=0.0.0.0:0<ip>) [\_] stage 2/2 >OK<

2013-05-18 23:30:46 PST i: [g2comm] <cda1> Detect: HTTP PING to 66.151.158.177:8200 (proxy: None=0.0.0.0:0<ip>) [\_] stage 2/2 >OK<

2013-05-18 23:30:46 PST i: [g2comm] <cda1> Detected 3 connections to 66.151.158.177, source: IE;Firefox;Netscape;Registry;WPAD script

If there aren’t any proxies, then we’d expect GoToMyPC to be able to connect successfully when it sends the streaming communications request. It fails.

2013-05-29 14:48:22 PST i: [g2comm] <PollThread> JEDI connect: Creating Direct socket

2013-05-29 14:48:22 PST i: [g2comm] <PollThread> JEDI connect: Connected to address[0] 66.151.158.177:8200

2013-05-29 14:48:22 PST i: [g2comm] <RemoteLogging(SetRemoteDest)(0)> comm::jinet::JSpecProviderBroker::getHttpProvider(): Matched the singleton connection spec provider

2013-05-29 14:48:23 PST i: [g2comm] <RPCThread:IPersistentPollConnection.waitForNotification> CPersistentPollConnection::waitForNotification - report notification: -2

2013-05-29 14:48:23 PST W: [g2tray] <OnPersistent(ECMethodTask)(0)> CCommunication::poll()

(4004) "ECNetworkError::eHTTPError"

HTTP request failed: 403 Forbidden

So if there’s not a proxy, why would the request be forbidden? The answer is because there is a transparent proxy. The transparent proxy intercepted the request that was sent over 8200 and rejected it. For transparent proxies the recommended setting to use is

**HTTP Communications** – Use port 80

**Streaming Communications** – Use port 443

HTTP communications are simple requests sent to our servers and they can use port 80 just like normal traffic from an internet browser. Streaming communications should use port 443. This will wrap up the request in SSL and this should prevent the transparent proxy from recognizing that the request is a JEDI request and rejecting it. Here is the same case with port 443.

-05-29 15:30:58 PST i: [g2comm] <cda2> Verifying: Jedi 66.151.158.177:443 [0.0.0.0:0(None<ip>)], method=SSL stage=Manual source=[U]

2013-05-29 15:30:58 PST i: [g2comm] <RemoteLogging(SetRemoteDest)(0)> comm::jinet::JSpecProviderBroker::getHttpProvider(): Matched the singleton connection spec provider

2013-05-29 15:30:58 PST i: [g2comm] <cda2> Verified: Jedi 66.151.158.177:443 [0.0.0.0:0(None<ip>)], method=SSL stage=Manual source=[U]

2013-05-29 15:30:58 PST i: [g2comm] <PollThread> JEDI connect: Creating SSL socket

2013-05-29 15:30:58 PST i: [g2comm] <PollThread> JEDI connect: Connected to address[0] 66.151.158.177:443

**SSL Inspecting Proxies**

An SSL inspecting proxy is a proxy that will attempt to examine all SSL traffic to ensure that the traffic adheres to SSL protocol. In most environments, if SSL inspection is enabled, communications over port 8200 will not be allowed. Requests over port 80 will be rejected because the proxy will not be able to understand the JEDI request that is wrapped in HTTP. Port 443 becomes the only option. GoToMyPC uses fake SSL when it connects over port 443.

**The Fake SSL Problem**

When GoToMyPC sends out JEDI requests over port 443, it wraps up the request in SSL. The problem is “fake SSL” or rather the wrapping it uses doesn’t strictly adhere to SSL protocol. It’s wrapped up in such a way as to make most proxy servers think that it is. In most cases, the SSL wrapper that GoToMyPC uses is able to “convince” a proxy server that its communications are standard SSL communications and the proxy server forwards them along like normal. Some proxy servers will actually inspect the SSL traffic to make sure that it conforms to SSL protocol. When this happens, the inspecting proxy server finds out that GoToMyPC isn’t using real SSL and the traffic can get rejected.

In the lines below, note how GoToMyPC is using port 443 and is creating an SSL socket. The response it gets is unexpected. This is because a transparent proxy at the host inspected the SSL packet and rejected it after it determined the packet did not conform to the SSL protocol. The line about the comm server refusing it is interesting because in this case a transparent proxy was the likely recipient of the request and it was responsible for refusing it.

2013-02-28 07:16:40 PST i: [g2comm] <MG\_hiWR13582204> JEDI connect: Creating SSL socket

2013-02-28 07:16:41 PST i: [g2comm] <MG\_hiWR13582204> JEDI connect: Connected to address[0] 216.115.209.52:443

2013-02-28 07:17:02 PST E: [g2comm] <MG\_hiWR13582204> JConnection: JEDI response read error

2013-02-28 07:17:02 PST E: [g2comm] <MG\_hiWR13582204> join channel error (-1)

2013-02-28 07:17:02 PST i: [g2comm] <MG\_hiWR13582204> MG createConnection: commServer refused us, so close channel

2013-02-28 07:17:02 PST W: [g2comm] <RPCThread:IChannel.join> reconnector's waitUntilConnected returns FAIL.

**Workaround**: **No workarounds**. The only workaround is to convince a network administrator to disable SSL inspection or open up communications through port 8200. GoToMyPC does not support real SSL at this time and there are no workarounds.

**2.3 Architecture of GoToMyPC’s connection logic**

When GoToMyPC (Host & Client) run for the first time, a connection detection algorithm (CDA) is run to determine what the best connection settings are. CDA is basically a tool that looks for all of the possible connection settings and by trial and error decides which one is the best.

The first thing CDA will do is try to connect directly to Citrix servers using ports 8200, 443, and 80. When we say direct connection, we mean that the request is being directly from the user’s computer to a Citrix server and not going through a proxy. If a direct connection is successful, it is preferred. CDA will also look for proxies. It will see what settings Internet Explorer and Firefox use. If it finds any proxy settings it will store them in the registry so that GoToMyPC will use them. The idea is that if the web browser can get out to the internet with these settings then GoToMyPC should be able to as well. CDA can also search the user’s registry for some additional proxy settings. Some proxies may ask a user for authentication before they will accept incoming connections. In this case, when GoToMyPC connects to the proxy a user will be asked to supply a username and password. After the user enters this, GoToMyPC will store these credentials in the registry and the user should not be asked to give them again.

The specific registry folders that connection settings are stored is:

Client Computer:

**HKEY\_CURRENT\_USER\Software\Citrix\GoToMyPC\ConnectionInfo**

Host computer (32 bit system)

**HKEY\_LOCAL\_MACHINE\SOFTWARE\Citrix\GoToMyPC\ConnectionInfo**

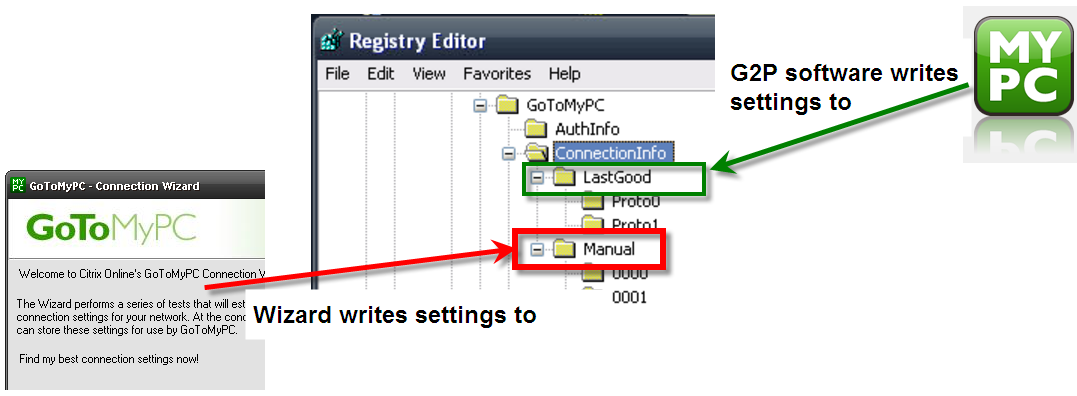
Host computer (64 bit system)

**HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Citrix\GoToMyPC\ConnectionInfo**

The connection folder is divided into two subfolders.

**LastGood** – After the software connects to our servers successfully once, it will remember this setting by storing it in the registry. The next time the software needs to connect to the internet it will try this remembered setting first. If the stored setting is successful, GoToMyPC will not attempt any of the other settings. If a connection using the stored setting is not successful, the other connection settings will be tried.

**Manual** – A manual folder is created in the registry whenever you run the GoToMyPC connection wizard. The registry entry here contains the set of instructions designate when running the GoToMyPC connection wizard. This folder’s settings override the LastGood folder settings. When you run the connection wizard, you have the choice of giving a connection method a certain priority. Try always means that the GoToMyPC software will always try the connection method you selected when running the connection wizard. If GoToMyPC is unable to connect to a session using the connection method you assigned, it will attempt other connection methods. If you tell the wizard to force a connection setting, the GoToMyPC software will only try your connection setting. If the connection setting fails the GoToMyPC software will not try a different connection method. There are two Proto folders, one for HTTP communications and one for JEDI communications.



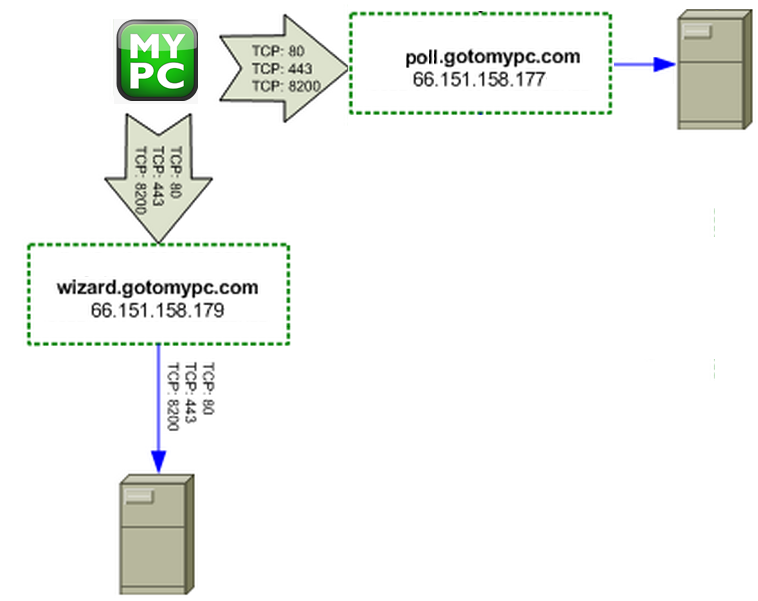
# 3 GoToMyPC CONnection wizard

**3.1 Connection Wizard Architecture**

The GoToMyPC Connection Wizard is a tool that attempts to locate and store the best connection settings for the GoToMyPC Host and viewer software to use. The connection wizard accomplishes this by aggressively looking for all possible connection methods of communicating with Citrix servers and by methodically trying each one. If the wizard is run in automatic mode it will automatically select what it determines to be the best connection method. If it is run in custom mode it will allow the user to select the best connection method.

The connection wizard reaches out to a server that is dedicated for connection testing. The server is accessible at wizard.gotomypc.com. The wizard tests both HTTP and Streaming Communications to this server. These tests are designed to simulate the kinds of connections that GoToMyPC will do within a session. The wizard will also test connectivity to the poll server. The wizard performs a DNS lookup to see if the url [www.gotomypc.com](http://www.gotomypc.com) resolves to 66.151.158.183. The Wizard also sends a long message. The long message is sent to determine whether or not GoToMyPC will be able to send out packets of information at its normal size. It also measures whether or not in session requests that use the JEDI protocol are likely to be successful.This is an example of the long message that is sent.

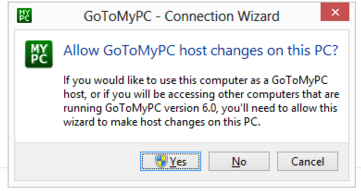
GET http://66.151.158.179:80/Jedi?request=ping&jedir\n

****

**3.1 Using the Automatic Wizard, Custom Wizard, and manual tools**

The wizard is capable of setting connection settings for the client computer as well as the host computer. Storing settings for the host computer requires administrator privileges.

When you first run the wizard, you may be prompted to authorize the tool to run with administrator privileges.

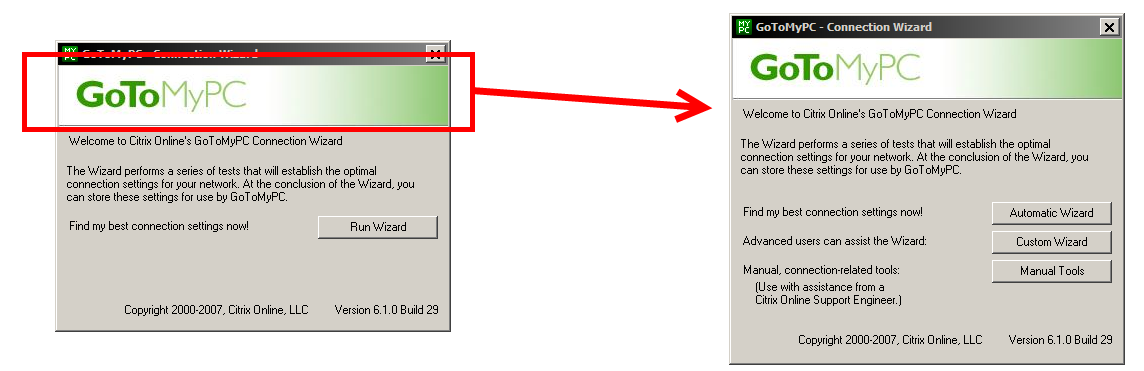


**Yes** – Changes will affect both host and client connection settings (requires administrator privileges)

**No** – Changes to settings will affect only client connection settings

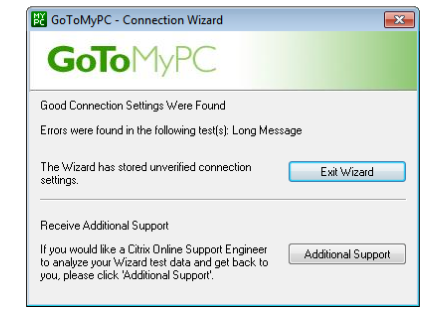
Clicking **Run Wizard** runs the automatic connection Wizard

Right click on the GoToMyPC logo banner to access the custom wizard and manual tools



Connection Wizard Troubleshooting Wizard

**Long Message**



*Explanation*: The connection wizard is sending a large amount of information using the JEDI protocol. This is done to try and recreate the actual request that GoToMyPC would send in a session. This test will either fail because the request was rejected because it was using the JEDI protocol or the size of the request was too large to handle.

*Resolution for MTU Threshold*: Decrease the size of the packets that the GoToMyPC software sends during sessions by adjusting a user’s flow control packet size in the GoToMyPC Internal Admin.

*Resolution for JEDI response rejected*: Force Streaming communications to use port 443. If this is not successful, the user may be in an environment where there is an SSL inspecting proxy. IT may need to be contacted to

**No options after running the custom wizard**

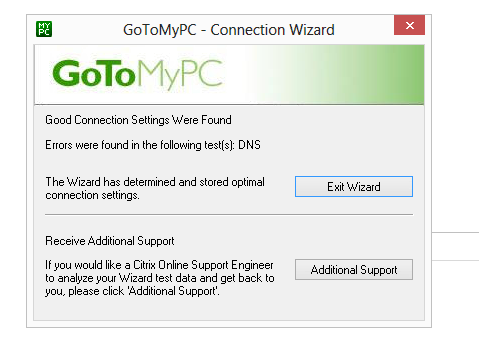
****

This means that the connection wizard tested all available connection methods and none of them were successful.

Resolution: Verify that the connection wizard has not been blocked by any security software

If the customer is in a corporate environment and uses a proxy, click add and try and add the proxy IP address and port. If the customer does not know, contact the network administrator with the Citrix Online IP Range Document.

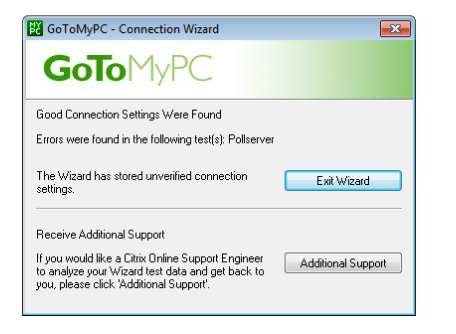
**DNS Error**



*Explanation:*The connection wizard performs a DNS lookup on [www.gotomypc.com](http://www.gotomypc.com) If the address resolves to 66.151.158.183 then the test is considered a success. If the url does not resolve to this IP address then the test fails

*Resolution:* A network administrator either needs to enable DNS for gotomypc.com or store a DNS entry for the address [www.gotomypc.com](http://www.gotomypc.com) in the hosts file on the user’s system.

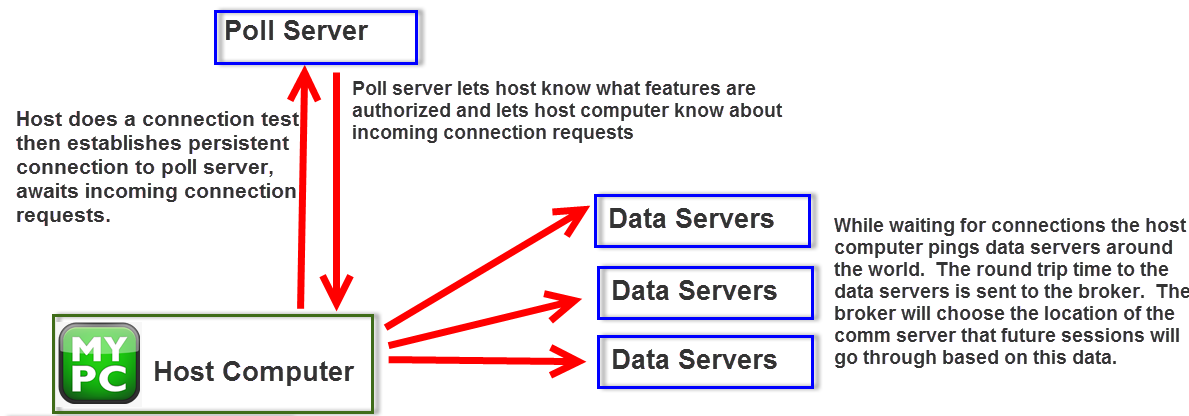
**Failed to connect to Poll Server**



Explanation: This means that the wizard could not reach poll.gotomypc.com at 66.151.158.177.

Resolution: Make sure that you can ping the poll server.

# 4 GoToMyPC Host Communications out of session



**4.1 Connection Test**

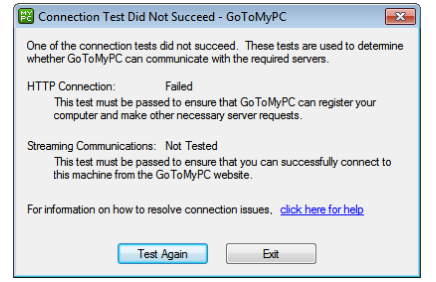
Each time the GoToMyPC host software starts up it performs a connection test. The connection test has two parts, one is the HTTP Request and the other part is the Streaming Communications.

**HTTP Request - > poll.gotomypc.com 66.151.158.177**

GET /conn\_test?nc=8527 HTTP/1.1Host: 66.151.158.177Content-Length: 0HTTP/1.0 200 OKContent-Type: text/plainCONNTESTIP=66.151.158.181

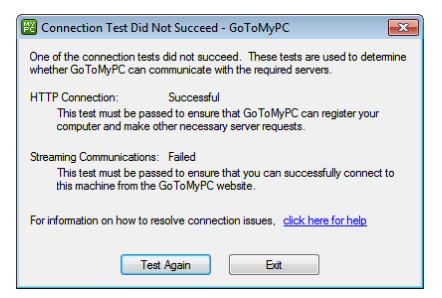
If this test fails, the GoToMyPC icon will show a red X  and a warning will appear

If the request is not successful, the user will be given a pop up. Clicking Exit will quit the GoToMyPC software.



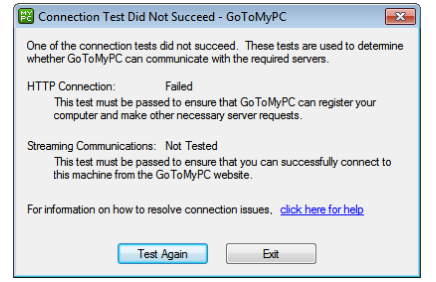
**JEDI Request -> connecttest.gotomypc.com 66.151.158.187**

JEDI request=ping&jedi=100 HTTP/1.0JEDI response=0&proxy=1009&sequence=0& HTTP/1.0



**4.2 Connection Test Troubleshooting**

**HTTP Connection Failed**



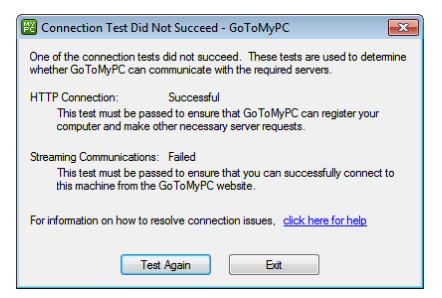
Explanation: This could mean one of three things. Either the user is completely offline, the GoToMyPC program may be blocked by a security program, or the request may be blocked by network security

Resolution for offline: Restore internet connectivity at the host computer

Resolution for security program blocking: Follow the instructions in the support center to ensure that GoToMyPC has full access to the internet

Resolution for network security: Run the connection wizard to try and ensure that if the user has a proxy that GoToMyPC is sending HTTP requests through the proxy. You may need to contact a network administrator to ensure that the IP addresses that GoToMyPC uses are whitelisted.

**Streaming Communications**



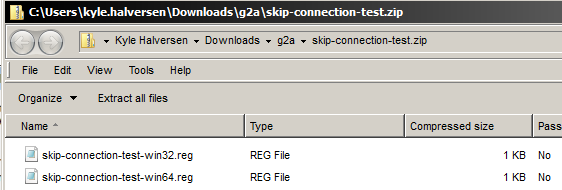
Explanation: This means that GoToMyPC was able to establish a connection to the test server, but the JEDI request did not complete successfully.

Resolution for security program blocking: Follow the instructions in the support center to ensure that GoToMyPC has full access to the internet

**Note:** In some rare cases a user may be able to successfully connect to a computer even though the connection test may fail. The root cause of this is still under investigation. As a temporary workaround, it is possible to force GoToMyPC to skip the connection test using a .reg file.

Download the file skip-connection-test.zip: <https://citrix.sharefile.com/d/s829b475dd8742948>

Extract the zip file



Depending on your version of windows, double click on either the 32 or 64 bit .reg file to add it to the registry.

**4.4 Persistent Connections**

The GoToMyPC software establishes what’s called a persistent connection to the poll server. This request is sent using the JEDI protocol. For JEDI connections, after the poll server responds to the request the communication channel is still open. This means that the GoToMyPC host software does not need to continually send pings to the poll server.

Here is how the persistent connection looks in the log files.

2013-04-09 17:22:08 PST i: [g2comm] <PollThread> Establishing persistent connection to Poll server

013-04-09 17:22:08 PST i: [g2comm] <cda2> Verifying: Jedi 66.151.158.177:443 [0.0.0.0:0(None<ip>)], method=SSL stage=CDA source=[R]

2013-04-09 17:22:08 PST d: [g2comm] <cda2> The last log message in this thread was repeated 2 time(s)

"Add ping stage: JEDI PING"

The channel will continue to stay open unless it is interrupted due to internet connectivity issues

2013-07-08 09:45:06 PST i: [g2comm] <81474/IPersistentPollConnection.waitForNotification> CPersistentPollConnection::waitForNotification - connection broken

(2010) "ECError::eIOError"

Socket error, err=10054 in , jsocket.cpp:876

2013-07-08 09:45:11 PST i: [g2comm] <PollThread> Establishing persistent connection to Poll server

2013-07-08 09:45:11 PST d: [g2comm] <PollThread> JEDI connect: Start connect to 66.151.158.177 (index=0)

2013-07-08 09:45:11 PST i: [g2comm] <PollThread> JEDI connect: Creating SSL socket

2013-07-08 09:45:11 PST d: [g2comm] <PollThread> JInet: Use Fake SSL connector

2013-07-08 09:45:11 PST i: [g2comm] <PollThread> JEDI connect: Connect to 66.151.158.177 failed

(4002) "ECNetworkError::eConnectError"

Connect failed

2013-07-08 09:45:16 PST W: [g2comm] <PollThread> JEDI connect: Connect failed

If the connection test is successful, the persistent connection to the poll server should also be successful.

**4.3 Get Options**

The host computer sends a special request to the poll server to verify what settings it should be using. This is called GetOptions. By settings, we mean things like whether or not the user has the ability to use remote printing, file transfer, etc. If for example the user has a corporate account and the user’s administrator has disabled remote printing, the poll server will tell the host computer this when it responds to the get options request. The request is an HTTP request. It has the following formatting:

GET /erc/GetOptions?build=830&platform=win32&machinekey=12060630&random=

In this example, the host computer is registered to a corporate user’s account. The corporate administrator has disabled the remote printing feature.

2013-03-20 17:01:21 PST d: [g2tray] <Options> SnapUpdOptionsIn ::=

ACExpirePeriod=-1

ACHardLockoutAttempts=-1

ACHistorySize=-1

ACNotificationPeriod=-1

ACSoftLockoutMinutes=5

AudioFlowCtlPacketSize=2048

AudioFlowCtlTokens=64

PerformanceMonitoring=true

PersistentConnection=true

Probe=false

RemotePrinting=false

If this request fails, the GoToMyPC icon will show up as being “Not authorized.”

**Troubleshooting:** The GET options request is a really long HTTP request. It’s uncommon for this request to fail if the connection test succeeds. It’s usually recommended that you run the connection wizard again and try another port for HTTP communications. If the user is in a corporate environment and a proxy is being used, make sure that HTTP connections use this proxy.

**4.5 Connection Probing**

When GoToMyPC is not in session it will periodically talk to different Citrix data servers around the world. This type of communication is called a connection probe in the log files. The probe is a ping that is designed to measure how long it takes to reach each data server. Connection probes happen every 5 minutes.

2013-07-07 08:23:24 PST d: [g2comm] <74438/IINet.probe> Probe connect: Start connect to 216.219.117.244

2013-07-07 08:23:24 PST d: [g2comm] <74459/IINet.probe> Probe connect: Start connect to 140.207.108.250

2013-07-07 08:23:24 PST d: [g2comm] <74461/IINet.probe> Probe connect: Start connect to 202.173.25.200

2013-07-07 08:23:31 PST d: [g2comm] <74459/IINet.probe> Probe connect: Start connect to 78.108.117.250

2013-07-07 08:23:32 PST d: [g2comm] <74459/IINet.probe> Probe connect: Start connect to 216.219.117.244

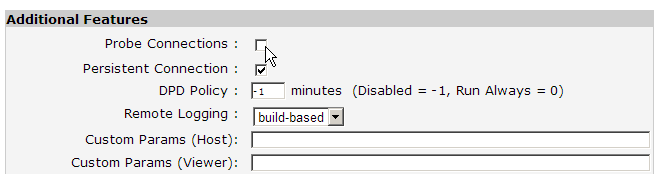
2013-07-07 08:23:33 PST d: [g2comm] <74459/IINet.probe> Probe connect: Start connect to 140.207.108.250

2013-07-07 08:23:33 PST d: [g2comm] <74461/IINet.probe> Probe connect: Start connect to 202.173.25.200

2013-07-07 08:23:33 PST d: [g2comm] <74458/IINet.probe> Probe connect: Start connect to 78.108.117.250

2013-07-07 08:28:24 PST d: [g2comm] <74483/IINet.probe> Probe connect: Start connect to 68.64.4.249

By request, connection probing can be turned off in the GoToMyPC Internal Admin.

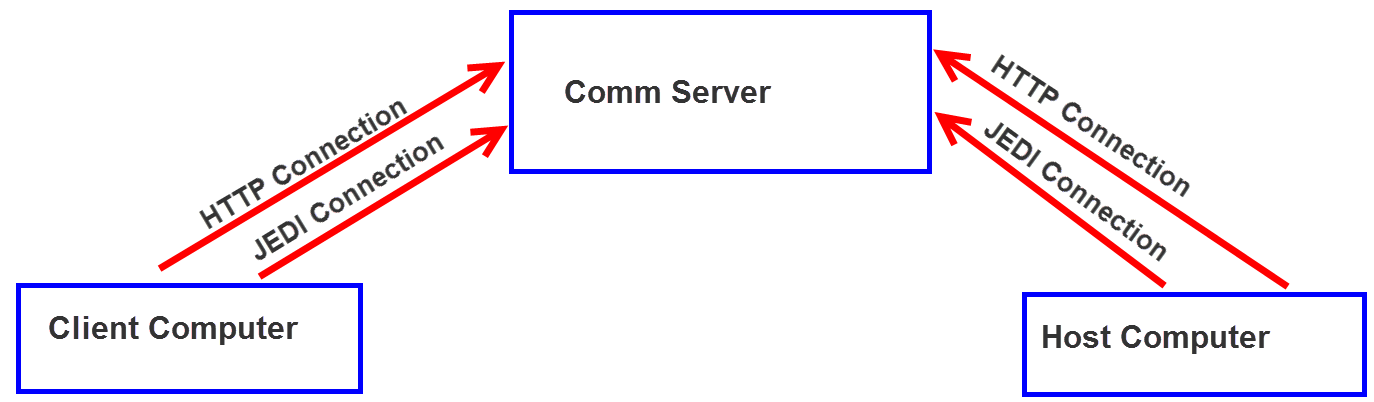


.

# 5 gotomypc starting a session

**5.1 Overview of communications**

The Broker selects a Comm Server that will be hosting the session. Both the host computer and the client computer try and establish their own separate connections to the comm server. Each side must establish both an HTTP connection and a JEDI connection. If any one of these connections fails, the entire session will not be able to start successfully.

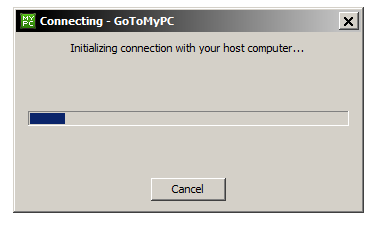


**SESSION START** means GoToMyPC is initializing the connection

2013-02-15 12:56:41 PST i: [g2tray] <Launcher-main> \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2013-02-15 12:56:41 PST i: [g2tray] <Launcher-main> --- SESSION START ---

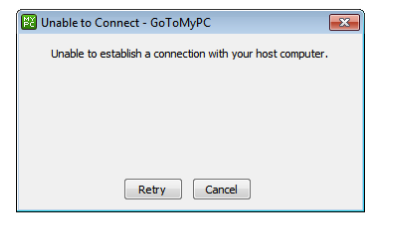
2013-02-15 12:56:41 PST i: [g2tray] <Launcher-main> \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



If both sides are able to connect to the comm server, the host software will let the client know to ask for the access code.

2013-02-23 07:53:15 PST i: [g2mainh] <G2HostAgentThread> Remote agent appeared, now authenticating

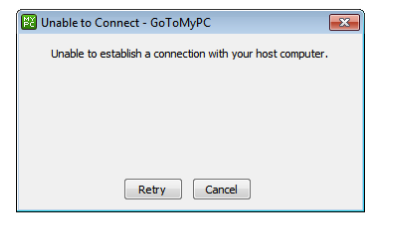




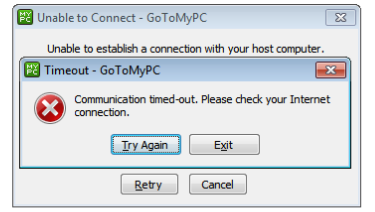
**5.2 Determining whether the problem is at the host or the client**

**Customer Symptom:**

If the connection to the host computer fails, the customer will see the following message on the client side:

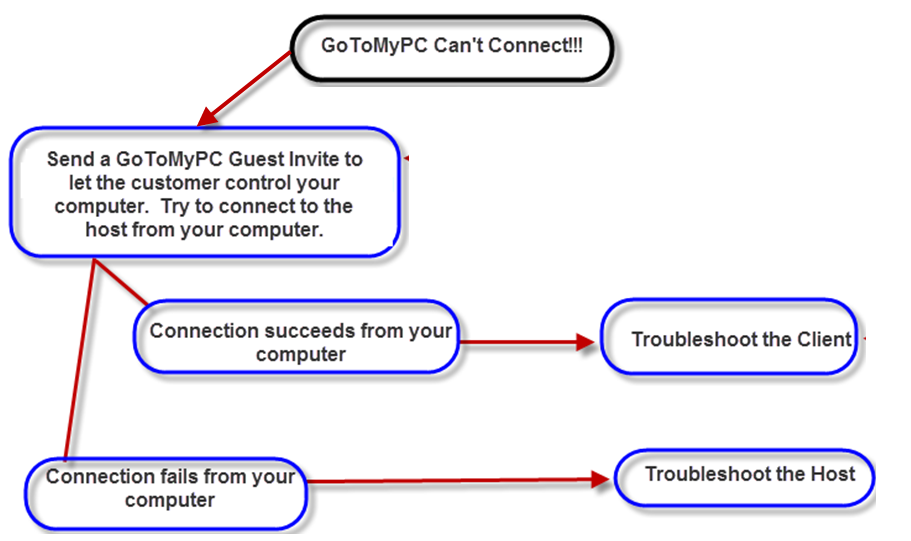


If the client is unable to connect to the Comm Server, an error message will appear if you wait a few moments.



If the connection timeout message does not appear, it is likely that the problem is at the user’s host computer. You can confirm this with additional troubleshooting.

**Troubleshooting to find out if the problem is at the host or client:**



**Using the log files:**

You can search a log file for **Connected to comm server**

If the connection to the comm server is successful on the client side you will see:

2013-07-11 12:00:57 PST i: [g2viewer] <G2ViewerAgentThread> Connected to comm server, agent state -> eStateInit

If the connection to the comm server is successful on the host side you will see:

2013-07-09 06:44:53 PST i: [g2mainh] <G2HostAgentThread> Connected to comm server, \_agent state -> eStateInit

If everything is going well on the host side it should connect to the comm server even if the client fails to do so. If everything is going well on the client side, it should still connect to the comm server even if the host fails to do so.

If the connection fails, you need to identify whether or not it was HTTP Connections or Streaming Communications that failed and troubleshoot accordingly

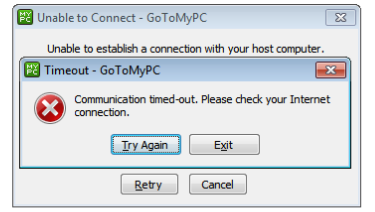
**5.3 Troubleshooting Client Connection Issues**

**Client Software blocked by firewall/security software**

**Bad Connection Settings**

**Network Security**

**Client Software blocked by firewall/security software**



Explanation: Most people use GoToMyPC to connect from a computer at their home to a computer in the office. These users are more likely to use security software. If GoToMyPC is blocked by security software, all connection requests will fail. The log file lines will look like this:

2013-07-02 13:09:34 PST i: [g2viewer] <cda2> Detect: TCP connect to 54.249.44.207:80 (proxy: None=0.0.0.0:0<ip>) [\_]

2013-07-02 13:09:35 PST i: [g2viewer] <cda1> Detect: TCP connect to 54.249.44.207:8200 (proxy: None=0.0.0.0:0<ip>) [\_] stage 1/4 >ERR<

2013-07-02 13:09:35 PST i: [g2viewer] <cda1> Detect: TCP connect to 54.249.44.207:443 (proxy: None=0.0.0.0:0<ip>) [\_] stage 1/5 >ERR<

2013-07-02 13:09:40 PST i: [g2viewer] <cda1> CDA to 54.249.44.207, failed

2013-07-02 13:10:03 PST i: [g2viewer] <Agent-poll> Agent::doHelloRequest() - HTTP request failed - line[] count[1]

(2024) "ECError::eTimeout"

HTTP exchange timeout waiting for response

Error returned at , httpreq.cpp:69

2013-07-02 13:10:03 PST W: [g2viewer] <Agent-poll> HELLO failed to connect

In other words, what the log files are saying is that the program tried all possible connection settings and none of them worked. This would suggest that GoToMyPC is getting blocked.

*Resolution*: The client needs to configure their security software so that the file g2viewer.exe to have full outbound access to the internet.

Client software blocked by firewall/security software.

G2viewer location:

**%temp%\G2\_<Build Number>\g2viewer.exe**

Note: This file is only in that location when the GoToMyPC Viewer software is running. If will disappear if you cancel your connection request

**Bad Connection Settings**

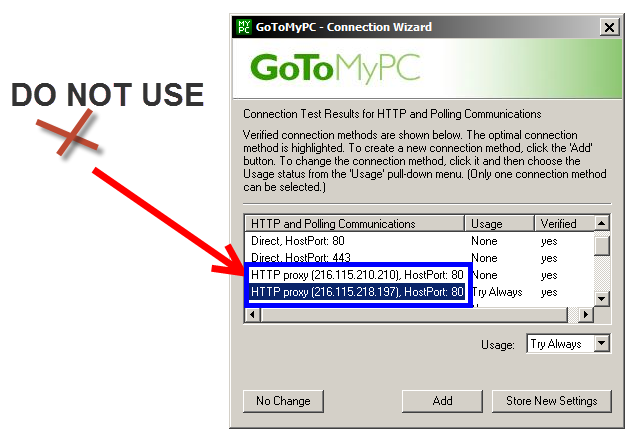
There are multiple connection methods to get from a user’s computer to Citrix servers. A bad connection setting is a method that just won’t work in a user’s environment. If a user is at a home location where there is no proxy, GoToMyPC should not be using a proxy. If GoToMyPC is trying to communicate through a proxy that is not accessible, the connection will fail. The connection method that GoToMyPC is using is not visible to the user. You can only identify what method the connection is using through the log files. Search for cda in the log files. In the lines below, GoToMyPC was trying to talk to a Citrix server as if it was a proxy. This caused connection problems.

2013-05-25 09:07:57 PST i: [g2viewer] <cda2> Started CDA Startup, address count = 5, allotted time = 180000ms

2013-05-25 09:07:57 PST i: [g2viewer] <cda2> Verifying: HTTP 68.64.5.85:80 [216.115.214.236:443(HTTP<ip>)], method=Proxy stage=CDA source=[R;rd]

2013-05-25 09:07:58 PST i: [g2viewer] <cda2> Verified: HTTP 68.64.5.85:80 [216.115.214.236:443(HTTP<ip>)], method=Proxy stage=CDA source=[R;rd]

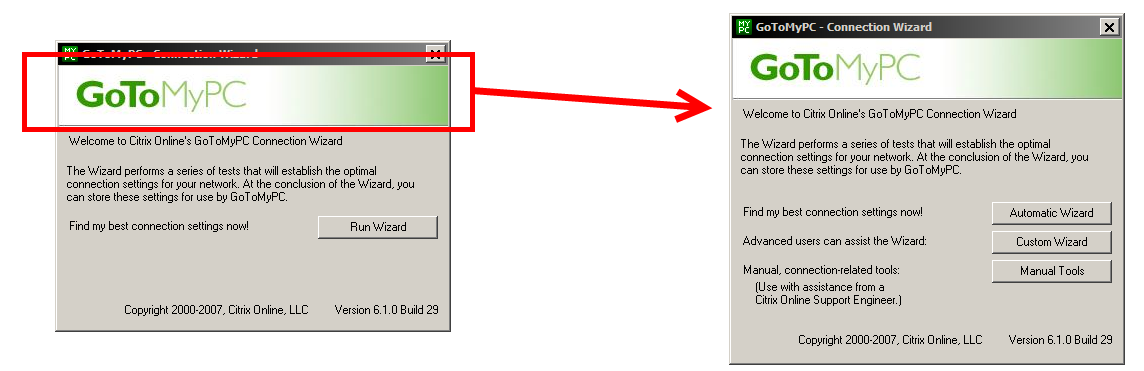
The IP addresses listed there are actually Citrix servers. These servers should not be used for proxy servers. If you want to know whether a proxy server that appears in the wizard result is a Citrix Data server or not, check [www.citrixonline.com/iprange](http://www.citrixonline.com/iprange) This happens because of a bug in GoToMyPC’s connection logic. If you use GoToMyPC and GoToMeeting at the same time, the GoToMyPC program might confuse GoToMeeting’s connection for a proxy connection.



*Resolution*: Reset the connection settings with the wizard manual tools

Open the connection wizard.

Right click on the GoToMyPC Logo Banner



Click Reset

Click Reset

This will clear out the bad settings. The user can try and connect again.

**Network Security**

Explanation: Sometimes the client is attempting to connect to a host computer and the client is in a corporate environment with network restrictions. There may be multiple proxy servers in that customer’s environment. We want to talk to the right proxy. The right proxy is the one that will take a request from GoToMyPC and promptly pass it along through a customer’s firewall to our data servers. The “wrong” proxy is one that will take the request and not pass it along to our data servers. We might also call a proxy the “wrong” proxy if it sends along our requests, but it may take much longer to do this than another proxy or another connection method would.

The way that GoToMyPC tells whether or not it is using the “right” proxy is by trial and error. The GoToMyPC software runs a connection detection algorithm (cda) to figure out what connections methods to try. Each proxy that is found will be tested and the results will be stored in the registry. The software first simply tries to open up a TCP connection to the proxy to see if it can be reached. If this is successful it will try HTTP communications. If this is successful it will begin to send JEDI requests. Usually if a connection to a proxy is unsuccessful, it will happen at either the TCP or HTTP stages. Typically GoToMyPC is able to recognize the failure and try to talk to a different proxy or try another communication method. If the connection moved past the HTTP stages and into the JEDI stages and failed, GoToMyPC might not try another communication method even if the connection fails. This is why for corporate users it can be good to work with the user’s administrator to determine which proxy is the correct proxy that GoToMyPC should communicate with to get out to the internet. It’s also important to note that just because GoToMyPC uses

**HTTP Communication Error to Comm server**

Search for **cda** in the log files to find these types of reusts

2013-07-02 13:09:34 PST i: [g2viewer] <cda2> Detect: TCP connect to 54.249.44.207:80 (proxy: None=0.0.0.0:0<ip>) [\_]

2013-07-02 13:09:35 PST i: [g2viewer] <cda1> Detect: TCP connect to 54.249.44.207:8200 (proxy: None=0.0.0.0:0<ip>) [\_] stage 1/4 >ERR<

2013-07-02 13:09:35 PST i: [g2viewer] <cda1> Detect: TCP connect to 54.249.44.207:80 (proxy: None=0.0.0.0:0<ip>) [\_] stage 1/3 >ERR<

2013-07-02 13:09:35 PST i: [g2viewer] <cda1> Detect: TCP connect to 54.249.44.207:443 (proxy: None=0.0.0.0:0<ip>) [\_] stage 1/5 >ERR<

*Resolution:* Run the wizard multiple times and try and identify the user’s proxy. If this is unsuccessful, contact the network administrator.

**Streaming Communication Error to Comm Server:**

Search for **JEDI** to find these types of requests in the log files

2013-07-13 17:06:09 PST W: [g2viewer] <MG\_hiWR13610172> CChannelConnector::connectToCommServer() failed

(4002) "ECNetworkError::eConnectError"

2013-07-13 17:06:09 PST E: [g2viewer] <MG\_hiWR13610172> connect to commServer error

2013-07-13 17:06:11 PST i: [g2viewer] <FFOConnector> Connection failed for FFO server address: 216.115.214.23 [216.115.214.23]

2013-07-13 17:06:13 PST i: [g2viewer] <cda2> Failed: Jedi 67.217.77.22:8200 [0.0.0.0:0(None<ip>)], method=SSL stage=Manual source=[U]

(4002) "ECNetworkError::eConnectError"

**Resolution:** Run the connection wizard and pick another port for streaming communications. If only port 443 is available and you still see the same errors in the log files, there may be an SSL inspecting proxy in the customer’s network. The customer may have to talk to IT to disable SSL inspection.

**5.3 Troubleshooting Host Connection Issues**

Host unable to respond to connection requests

HTTP Communications to comm server fail

Streaming Communication to comm server failure

**Host Cannot Respond to Connection Requests**

The host computer establishes a connection to the poll server. The poll server will forward any incoming connection requests to the host computer. Usually the host is able to respond and prepare for the connection.

2012-05-30 01:04:13 PST i: [g2comm] <RPCThread:IPersistentPollConnection.waitForNotification> CPersistentPollConnection::waitForNotification - report notification: 696346630

2012-05-30 01:04:13 PST i: [g2tray] <OnPersistent(ECMethodTask)(0)> COnPersistent::poll() poller received connect request.

2012-05-30 01:04:13 PST i: [g2tray] <Launcher-main> \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2012-05-30 01:04:13 PST i: [g2tray] <Launcher-main> --- SESSION START ---

2012-05-30 01:04:13 PST i: [g2tray] <Launcher-main> \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

In some cases the host computer may not have shut down the last session successfully and it can’t accept a new session until after the old session shuts down properly

2012-05-30 07:12:05 PST i: [g2comm] <RPCThread:IPersistentPollConnection.waitForNotification> CPersistentPollConnection::waitForNotification - report notification: 696433283

2012-05-30 07:12:05 PST i: [g2mainh] <1103> ECActivityMonitor::handleTimerEvent(): inactivity timeout

2012-05-30 07:12:05 PST i: [g2mainh] <1103> G2HostAgent::quit( cause=13, reason=inactivityTimeout )

2012-05-30 07:12:06 PST i: [g2tray] <OnPersistent(ECMethodTask)(0)> COnPersistent::poll() poller received connect request.

2012-05-30 07:12:06 PST i: [g2mainh] <RPCThread:IG2HostAgent.stopSession> G2HostAgentPI::stopSession(anotherUser)

2012-05-30 07:12:06 PST i: [g2mainh] <RPCThread:IG2HostAgent.stopSession> G2HostAgent::quit( cause=0, reason=anotherUser )

2012-05-30 07:12:10 PST i: [g2mainh] <1103> ECActivityMonitor::handleTimerEvent(): inactivity timeout

**Resolution:** The only resolution is to restart the GoToMyPC software. This can be done by restarting GoToMyPC directly or rebooting the computer

**HTTP Communications failure:**

**Explanation:** In a corporate environment, sometimes HTTP communications may fail if GoToMyPC is not communicating using the corporate network’s proxy or is using the wrong proxy. In the log lines below, the user was in a corporate network and explained there were issues with GoToMyPC not connecting consistently. The user’s network had two proxies. One proxy was for the United States and one proxy was for users in Europe. The problem was that GoToMyPC would at times go through the proxy in Europe and this would cause connections to fail.

2013-02-21 06:34:40 PST i: [g2comm] <cda1> Verifying: HTTP 66.151.158.177:80 [10.138.239.2:80(HTTP<ip>)], method=Proxy stage=CDA source=[R;irw]

2013-02-21 06:34:40 PST i: [g2comm] <cda1> Failed: HTTP 66.151.158.177:80 [10.138.239.2:80(HTTP<ip>)], method=Proxy stage=CDA source=[R;irw]

(4002) "ECNetworkError::eConnectError"

Connect failed

When connections were routing through the United States proxy server they would be successful

2013-01-14 04:57:40 PST i: [g2comm] <cda1> Verifying: HTTP 66.151.158.177:80 [10.138.239.2:80(HTTP<ip>)], method=Proxy stage=Active source=[R;rw]

2013-01-14 04:57:42 PST i: [g2comm] <cda1> JAuthResolveConnector: proxy 10.138.239.2:80 supports auth: NEGOTIATE, NTLM, BASIC

2013-01-14 04:57:42 PST i: [g2comm] <cda1> Verified: HTTP 66.151.158.177:80 [10.138.239.2:80(HTTP<ip>)], method=Proxy stage=Active source=[R;rw]

Resolution: Confirm the correct proxy IP address with the network administrator. If the proxy does not change, you may consider running the connection wizard and forcing GoToMyPC to use that proxy for HTTP communications. If the proxy is dynamically allocated with an autoconfiguration file, make sure that this file does not list any proxy servers that may not be available.

**Streaming Communication Error to Comm Server:**

Search for **JEDI** to find these types of requests in the log files

2013-07-13 17:06:09 PST W: [g2comm] <MG\_hiWR13610172> CChannelConnector::connectToCommServer() failed

(4002) "ECNetworkError::eConnectError"

2013-07-13 17:06:09 PST E: [g2comm] <MG\_hiWR13610172> connect to commServer error

2013-07-13 17:06:11 PST i: [g2comm] <FFOConnector> Connection failed for FFO server address: 216.115.214.23 [216.115.214.23]

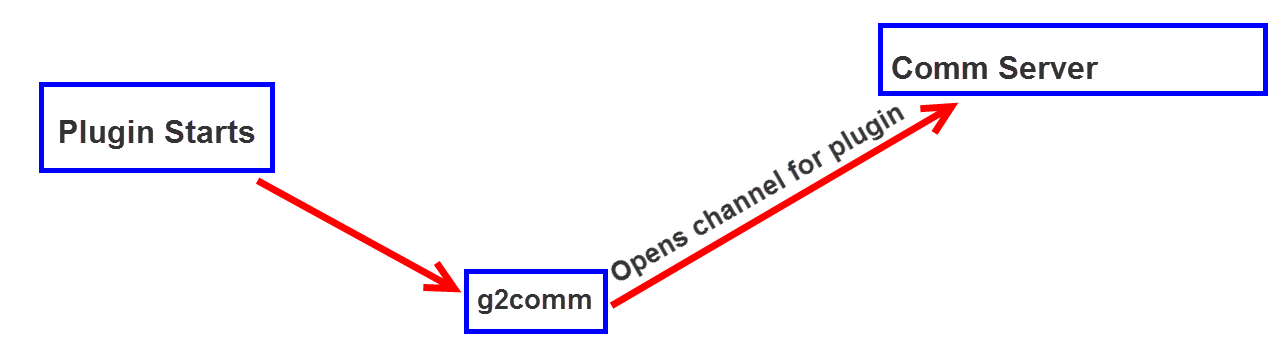
2013-07-13 17:06:13 PST i: [g2comm] <cda2> Failed: Jedi 67.217.77.22:8200 [0.0.0.0:0(None<ip>)], method=SSL stage=Manual source=[U]

(4002) "ECNetworkError::eConnectError"

**Resolution:** Run the connection wizard and pick another port for streaming communications. If only port 443 is available and you still see the same errors in the log files, there may be an SSL inspecting proxy in the customer’s network. The customer may have to talk to IT to disable SSL inspection.

# 6 In session communications

**6.1 In session communications**



At the beginning of the session, GoToMyPC starts up processes or plugins that are responsible for handling different features for the session. After the process starts up, g2comm opens up a channel to the comm server through which the data for the plugin will be sent.

Troubleshooting

**6.2 Troubleshooting In session communications**

If you’re having a problem with a particular plugin, make sure that it’s running when the GoToMyPC session starts.

Processes that control plugins

G2audioh.exe – Handles remote sound, must be running for sound to be available

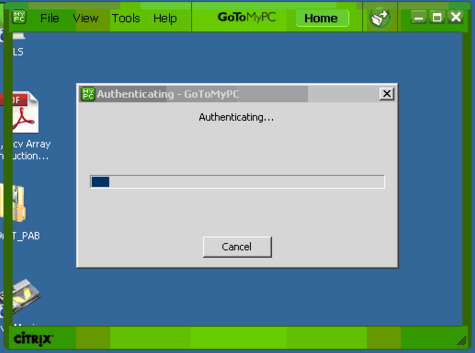
G2printh.exe – Handles remote printing

G2simpleft.exe – Handles file transfer between a Windows host and a Mac client

G2fileh.exe – Handles file transfer. Note: This plugin is only called up after the user attempts to do file transfer

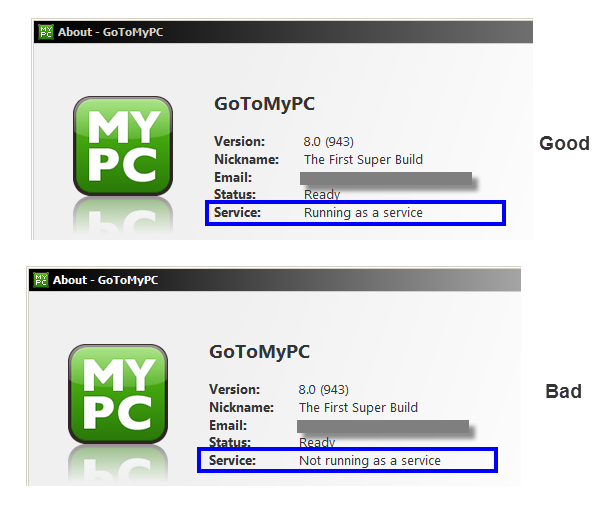
**Screen sharing does not start**

User symptom: The GoToMyPC software will prompt the client computer to enter the access code. After the client enters the access code, the GoToMyPC viewer will show authenticating. The viewer window will not expand into its full size and the GoToMyPC software may stop responding.



*Explanation:* GoToMyPC needs to be able to be running with the highest level of privileges (system level privileges) on the user’s system. If GoToMyPC does not have the highest level of privileges it may not be able to capture a user’s screen if the computer is locked or the user is not logged in.

*Resolution:* Restart the GoToMyPC software and make sure that it is running as a service



*Resolution if GoToMyPC is already running as a service:* Engineering is continuing to investigate this issue. The only workaround is to downgrade the user to an older build (830 or below).

The log file lines can be used to identify whether or not a user is affected by this issue. In the log file lines below, the screen sharing plugin g2host does not start up correctly. This happens because it was not launched with the appropriate permissions in order to be able to start screen sharing.

2013-05-17 06:26:47 PST i: [g2comm] <CPluginManagerIpcAdapter::getPlugin()> Creating process: "C:\Program Files\Citrix\GoToMyPC\g2host.exe" "StartID={CC79D7F8-2A04-4B81-AE23-8AB2993EC440}&Debug=Off&Stat=On&StatDb=Off&Index=0"

2013-05-17 06:26:47 PST E: [g2mainh] <G2HostAgentThread> CPluginProxy::CPluginProxy() - getPlugin\_() failed

                (2014) "ECError::eEnd"

                Plugin process quit

                Error returned at , starthere.cpp:123

                "g2host.exe" process exited with code 0 before establishing connection

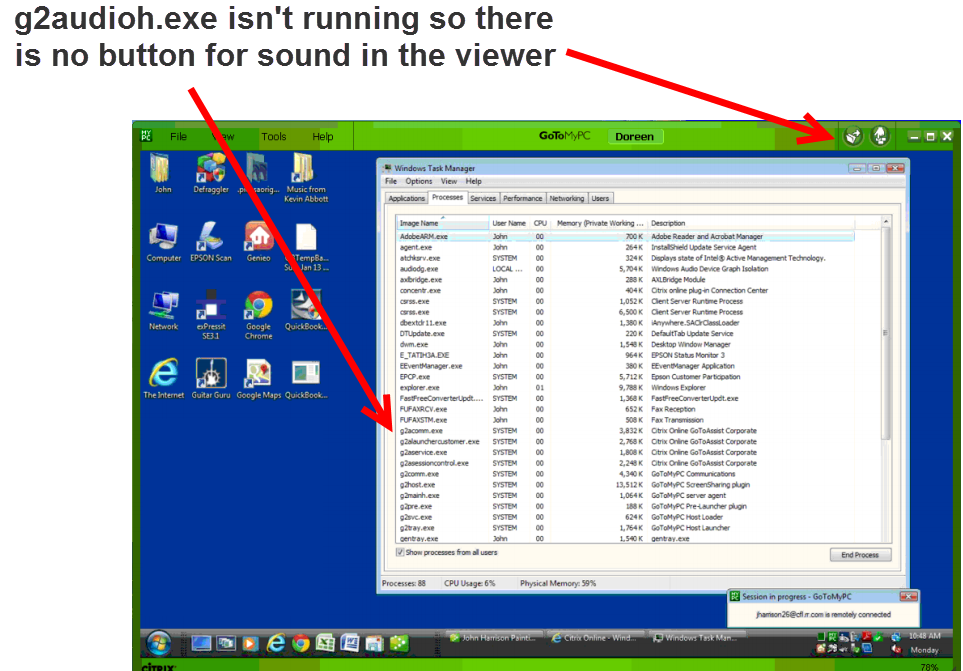
                Error returned at , ipcpluginhost.cpp:252

                Error returned at , pluginmanageripcadapter.cpp:135

                Error returned at , pluginprovider.cpp:59

**Other plugins (not screen sharing) not starting**

User symptom: Sound icon button is missing from viewer window. File transfer button is unresponsive and file transfer does not work



Explanation: The process that is responsible for this feature on the host computer is not running

Log file lines from an affected user’s system:

2013-07-10 07:24:18 PST i: [g2comm] <CPluginManager::getPlugin()\_1275> Creating process: "C:\Program Files\Citrix\GoToMyPC\g2printh.exe" "StartID={CC79D7F8-2A04-4B81-AE23-8AB2993EC440}&Debug=Off&Stat=On&StatDb=On&Index=0"

2013-07-10 07:24:19 PST W: [g2comm] <CPluginManager::getPlugin()\_1275> The G2Printing plug-in process exited while waiting for initial connection [Exit code = 0x800401E4].

2013-07-10 07:24:19 PST E: [g2mainh] <G2HostAgentThread> CPluginProxy::CPluginProxy() - getPlugin() failed:0x80040205

*Resolution*: Upgrade the user to the latest version of GoToMyPC

**Plugins not starting for Windows XP users**

Explanation: For Windows XP users, check to see if there may be any registry entries that are missing. GoToMyPC needs Windows to let it know that a user has logged in so it knows when to start plugins for different features. If some registry keys have bad permissions or are missing this error message may appear.

Log file lines from an affected user’s system:

2013-04-17 05:42:09 PST i: [g2mainh] <G2HostAgentThread> G2FileTransWrapper::startPlugin(1) enabled=true epoch=0

2013-04-17 05:42:09 PST i: [g2mainh] <G2HostAgentThread> G2FileTransWrapper::startPlugin(): cannot start plugin - no user logged in while running as local system

Method "WTSQueryUserToken" returned Win32 error [1245]. The operation being requested was not performed because the user has not logged on to the network.

Error returned at , ecwindowssecurity.cpp:1376

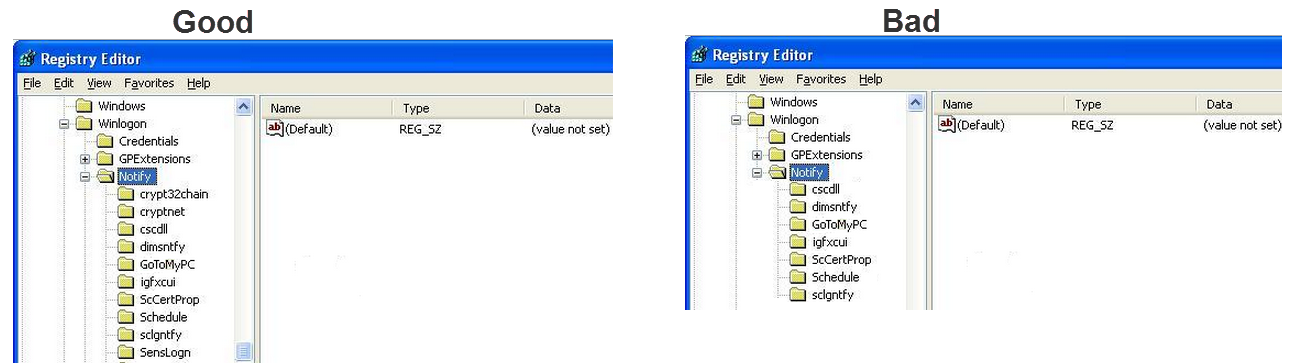
Error returned at , ecwindowssecurity.cpp:911

Error returned at , audiocontrol.cpp:1192

Resolution:

Make sure that all system registry keys are present in the registry folder **HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify**

Make sure that the GoToMyPC subfolder has the correct permissions



**Communication Channel can’t be opened to the comm server**

User symptom:



*Explanation:* GoToMyPC opens up many different communication channels to the comm server. In some networks the size of the packets of information GoToMyPC sends may be too large. By default, GoToMyPC sends out requests that can be as large as 2048 bytes. The largest size packet a network can send is referred to as the network’s Maximum Transmission Unit (MTU). If a packet is sent that is larger than the MTU the packet may become fragmented. If a networks’ MTU threshold is less than 2048, connections to the comm server may not be established properly.

The request to open up communications to the comm server may be too large. The network may fragment the packet.

2013-06-21 15:13:24 PST d: [g2comm] <SS\_hiWR12619324> join response: JEDI response=0&proxy=130600 HTTP/1.0

2013-06-21 15:13:24 PST d: [g2comm] <SS\_hiWR12619324> JConnection::joinCommServerSession() -- leave --

2013-06-21 15:13:24 PST d: [g2comm] <SS\_hiWR12619324> ChannelSplitter is being initiated

2013-06-21 15:13:24 PST d: [g2comm] <SS\_hiWR12619324> SS Reconnector: created a connection

2013-06-21 15:13:24 PST d: [g2comm] <SS\_hiWR12619324> SS Retransmission: send SYN packet

2013-06-21 15:13:24 PST i: [g2comm] <SS\_hiRD12608620> SS Retransmission: drain sees SYN packet

2013-06-21 15:13:24 PST d: [g2comm] <SS\_hiRD12608620> SS Retransmission: drain sees illegal sequence 18 (expected 23)

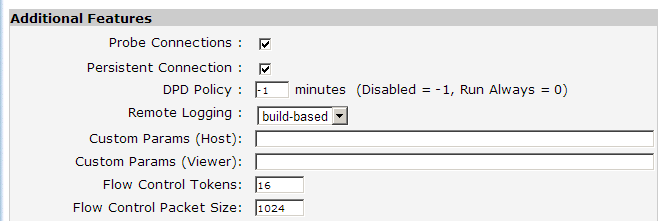
2013-06-21 15:13:24 PST d: [g2comm] <SS\_hiRD12608620> SS Retransmission: drain sees illegal sequence 19 (expected 23)

2013-06-21 15:13:24 PST d: [g2comm] <SS\_hiRD12608620> SS Retransmission: drain sees illegal sequence 20 (expected 23)

2013-06-21 15:13:24 PST d: [g2comm] <SS\_hiRD12608620> SS Retransmission: drain sees illegal sequence 21 (expected 23)

2013-06-21 15:13:24 PST d: [g2comm] <SS\_hiRD12608620> SS Retransmission: drain sees illegal sequence 22 (expected 23)

*Resolution:* Contact a network administrator to configure the network’s MTU threshold. GoToMyPC can also be configured to send smaller packets in Internal Admin.

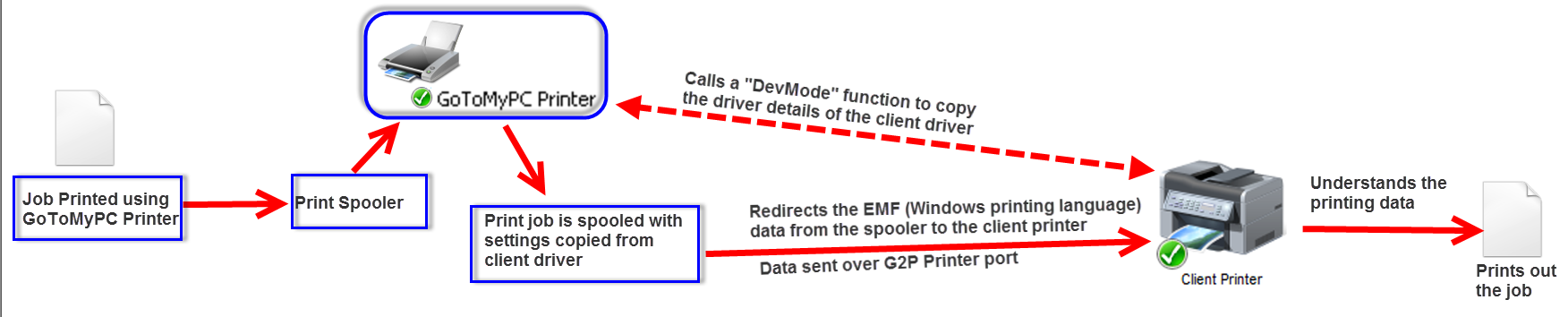


# 7 Remote printing

**7.1 Architecture: GoToMyPC Universal Printing**

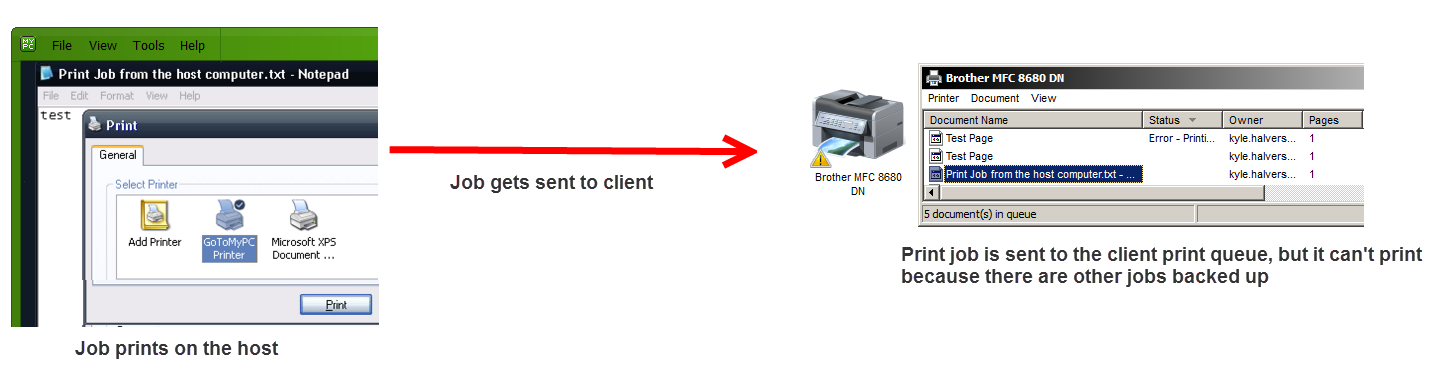
When you connect to a host computer the GoToMyPC software will create a printer called the GoToMyPC Printer on the host computer. This GoToMyPC Printer will become the default printer on the host computer. When you print out a document, the GoToMyPC printer will redirect the print job from your host computer to the printer on your client computer.

How does the GoToMyPC Printer work? The GoToMyPC uses a print driver that’s called the Universal Print Driver. A better name for the driver might be a “copycat” driver. What happens is that GoToMyPC examines the print driver on the client. It checks to see what the printing capabilities of the client printer are. This includes things like its margin size, dots per inch, etc. The name for this check is called the dev mode. Once GoToMyPC finds out what the priner capabilities of the client printer, the GoToMyPC Printer makes a driver that copies the capabilities. So if the client print driver uses .125 margins then the GoToMyPC Print Driver on the host computer will use .125 margins.



**7.2 Troubleshooting: Best Practices**

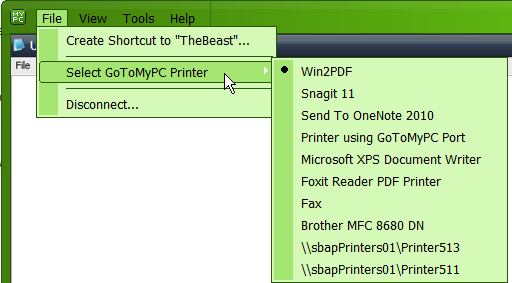
**Make sure that the client printer is able to print locally**



GoToMyPC can send print jobs from the host computer to the client computer. If the client print queue is backed up, new jobs will not print out.

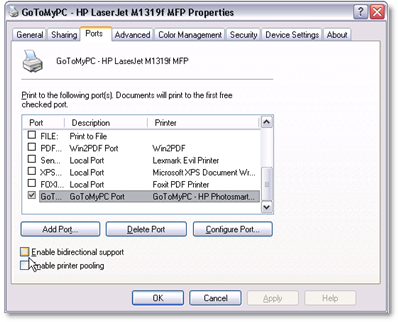
**Make sure that print jobs go to the right printer**

If you have multiple printers on your client computer, make sure that GoToMyPC is sending the print job to the correct printer. If the user takes no action, GoToMyPC will send the print job to the default printer at the client computer.



**Disable bidirectional support**

Bidirectional support is a printing feature where the printer and the computer communicate to each other. The printer can inform the computer that it is out of paper or whether or not ink levels are low. The problem with this feature is that the actual printer is at the client location and not the host. This can prevent the print driver from functioning correctly. It is strongly recommended that you disable bidirectional support as a best practice. If you don’t some print jobs may not spool correctly and get stuck.



**7.3 Troubleshooting: Missing GoToMyPC Printer**

Symptom: The GoToMyPC Printer is not created when the user connects to the host computer

There are three things that need to be in place in order for the GoToMyPC Printer to show up when a session starts. The log files can tell you exactly why the GoToMyPC Printer is missing. If you don’t use the log files then you have to check on all of the Printing components one by one.

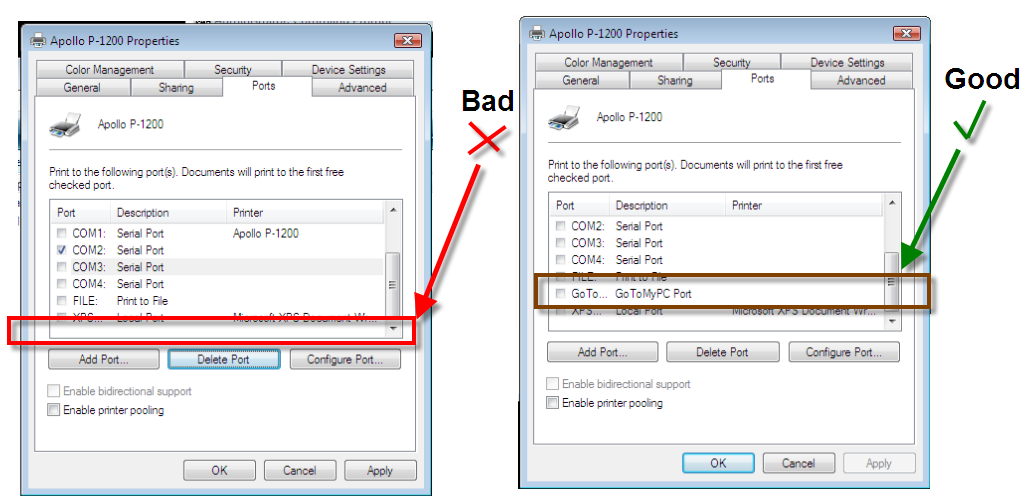
**The GoToMyPC Printer port needs to be installed and available**

**The GoToMyPC Print Driver and processor need to be installed**

**The printercontrol thread needs to be running to add the printer**

**The GoToMyPC Printer port needs to be installed and available**

Symptom of this problem: The GoToMyPC Printer Port is not available in the list of ports for printers



Log file lines:

2012-05-11 10:23:16 PST E: [g2tray] <Launcher-main> Error in , mainthread.cpp:3973

Error returned at , printercontrol.cpp:93

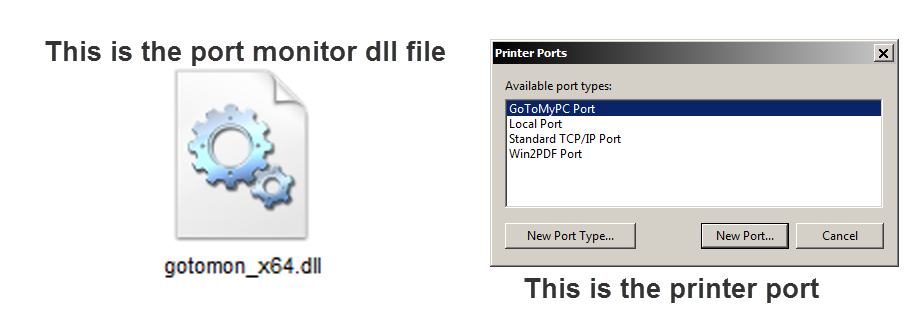
Error returned at , printercontrol.cpp:245

Couldn't add 'GoToMyPC Printer'

Method "AddPrinterW" returned Win32 error [1796]. The specified port is unknown.

(2001) "ECError::eSomeError"

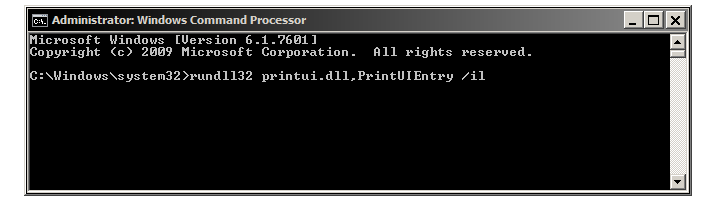
If it didn’t then something went wrong with the installation. There are two components to the GoToMyPC Printer Port. There is the port monitor DLL file and there is the printer port that gets added to the list of ports from the DLL file.



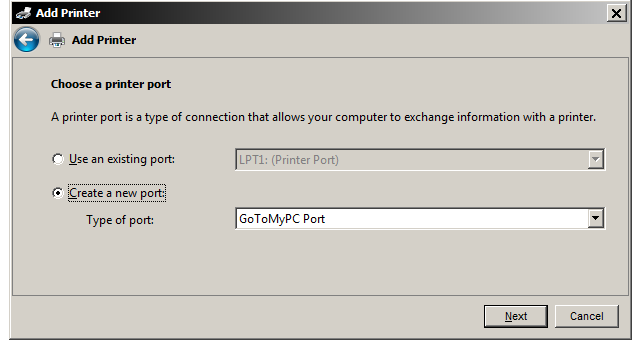
Sometimes after installation on some Windows Vista and Windows 7 systems, the printer port does not get added to the list from the DLL file. If this happens, you can add it to the list manually.

Open up a command prompt with Admin rights and enter:

**rundll32 printui.dll,PrintUIEntry /il**



This will open up the Add Printer Wizard with administrative privileges. From here you can add the GoToMyPC Printer Port Successfully.



**The GoToMyPC Print Driver and processor need to be installed properly**

Symptom of this problem: The GoToMyPC Universal Print Driver does not show up as an installed driver in Windows

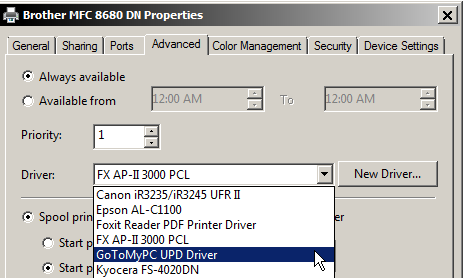
The GoToMyPC Printer should only show up while a GoToMyPC Session is running. The print driver is always installed even if there is not a printer that is using it.

If you open up the printer properties for any printer that is installed on the host computer

Click the advanced tab

Select the driver drop down

The GoToMyPC UPD driver should be in the list of drivers



Resolution: If the driver is not in the list, it means that it did not install correctly. It’s usually recommend that you uninstall GoToMyPC, reboot and reinstall. For some more difficult cases you may want to remove the GoToMyPC Printing components manually before attempting to reinstall GoToMyPC or manually set up a native printer instead of using the GoToMyPC Universal Printer.

**For 32 bit computers:**

The GoToMyPC Printer Port DLL file**:**

**C:\Windows\System32\gotomon.dll**

The GoToMyPC Universal Print Driver DLL File: **C:\Windows\System32\spool\drivers\w32x86\3\G2PrintUPDDriver.dll**

The GoToMyPC Print Processor DLL File:

**C:\Windows\System32\spool\prtprocs\w32x86\GoToPrintProcessor.dll**

Registry Folder for the GoToMyPC Printer Port:

**HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Print\Monitors\GoToMyPC Port**

Registry Folder for the GoToMyPC Universal Print Driver:

**HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Print\Printers\GoToMyPC**

Registry Folder for the GoToMyPC Print Processor:

**HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Print\Environments\Windows x64\Print Processors\GoToMyPC Print Processor**

**For 64 bit systems:**

The GoToMyPC Printer Port DLL file**:**

**C:\Windows\System32\gotomon\_x64.dll**

The GoToMyPC Universal Print Driver DLL File:

**C:\Windows\System32\spool\drivers\x64\3\G2PrintUPDDriver\_x64.dll**

The GoToMyPC Print Processor DLL File:

**C:\Windows\System32\spool\prtprocs\x64\GoToPrintProcessor\_x64.dll**

Registry Folder for the GoToMyPC Printer Port:

**HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Print\Monitors\GoToMyPC Port**

Registry Folder for the GoToMyPC Universal Print Driver:

**HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Print\Printers\GoToMyPC**

Registry Folder for the GoToMyPC Print Processor:

**HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Print\Environments\Windows x64\Print Processors\GoToMyPC Print Processor**

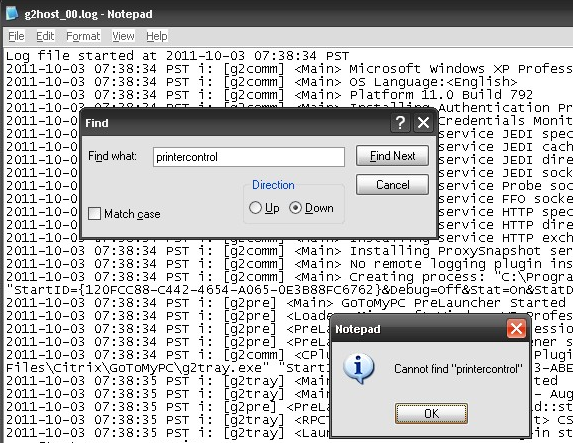
Note: Most print driver installation issues will be resolved in the next build. For extreme cases, a tool is available to manually install printing components. This tool has not been QA tested and is available only by request.

**The printercontrol thread needs to be running to add the printer**

Symptom of this issue: The GoToMyPC Printer will randomly not show up when a session starts. Rebooting the computer or reinstalling GoToMyP will temporarily bring it back, but it may disappear again

Explanation: This happens when part of the GoToMyPC software, the printercontrolthread is not running on the host computer.

In the log files on the host computer search for **printercontrol**

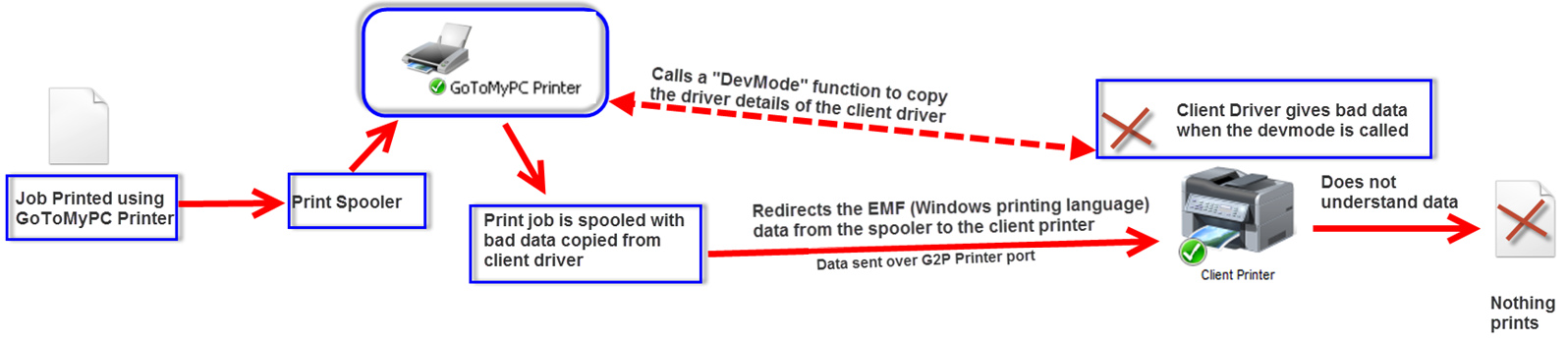


If printercontrol is not found on the logs, then this is the cause of the issue. The printercontrol thread is supposed to start every time that GoToMyPC starts up. If it’s not running then it won’t be able to add the GoToMyPC Printer when the session starts.

Resolution: Upgrade the customer to build 943. Build 943 put in a fix for this problem by manually starting the printercontrolthread when the GoToMyPC session starts if it is not running.

**7.4 Troubleshooting: GoToMyPC Print Job doesn’t print out**

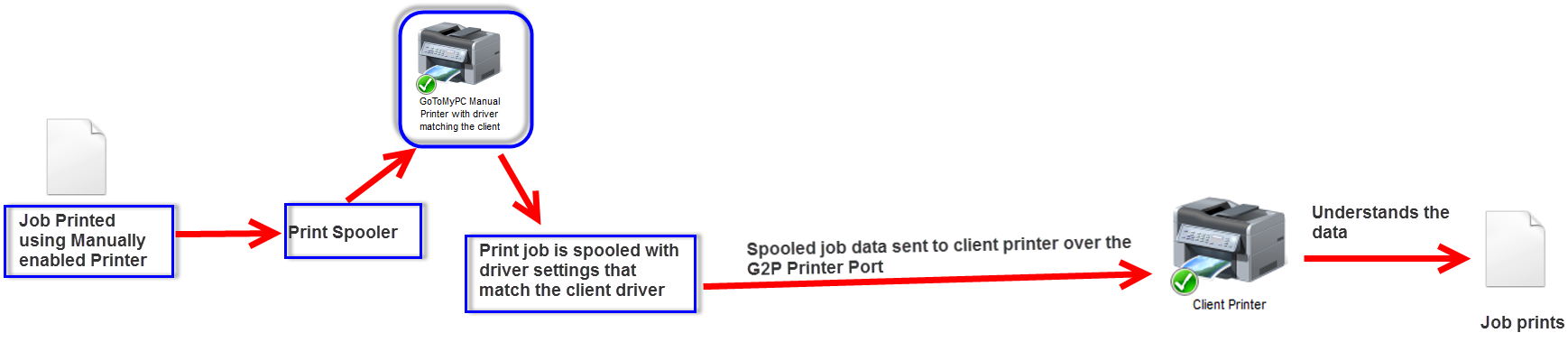
Whether or not the GoToMyPC Printer works with the client’s printer depends on whether or not GoToMyPC was able to collect good information on the client’s printer driver. The call to get information on the client’s printer driver is called the devmode call. There is not a lot of standardization with print drivers and so sometimes when the dev mode checks to see things like color profiles it doesn’t get any helpful information. When the GoToMyPC Printer is created its driver is not a good match up with the client print driver and so the print jobs won’t be understood when they are sent to the client.



If the GoToMyPC Printer is not able to work with a particular model printer on the client, there is no troubleshooting that can be done. The only option is to manually enable a printer on the host.

**7.5 Architecture: Native Printing**

When you manually enable a printer, you’re setting up a print driver on the host that is identical to the print driver on the client. With the GoToMyPC Universal Printer, the viewer tried to copy the client driver settings, but there was no guarantee that it would be able to find and use these settings. With a manually enabled printer, you’re downloading an identical driver from the printer manufacturer. The manual printer can actually spool the job and the result should be in a format that the client printer can understand.



**7.6 Troubleshooting: Native Printing**

Troubleshooting native printing will need to be done using both the host log files and the client log files. It is important to make sure that the client print driver and host print driver match up.

Here is an example where the print driver that was manually setup on the host computer does not match up with the client print driver:

Host

2013-07-09 16:15:49 PST i: [g2printh] <CStreamPrinterData::sendPrintJob()> Drivername: Lexmark C540 Class Driver

2013-07-09 16:16:00 PST i: [g2printh] <CStreamPrinterData::sendPrintJobFinish()> G2PrintServer::SendPrintJobFinish() (Finished Ok)

When a job is sent from a manually enabled printer, the viewer will check to see if the driver that is being used on the manually enabled printer matches the driver of the printer receiving the job on the client. If the drivers don’t match, the user asks the user to select a different printer from the list of other printers.

Client

2013-07-09 16:12:43 PST i: [g2viewer] <DataPipe\_RemotePrinting-main> Drivername: Lexmark C540 Class Driver

2013-07-09 16:12:43 PST d: [g2viewer] <92527> G2SelectUnmatchedPrinter()

2013-07-09 16:12:53 PST i: [g2viewer] <92527> ReceivePrintJobMsgThread::Run() - No printer matches, user chose [\\sbapPrinters01\Printer513]

2013-07-09 16:12:54 PST i: [g2viewer] <92529> ReceivePrintJobThread::PrintRawStream(): Printed stream (13193 bytes)

Here is another example where the print drivers match

Host

2013-07-09 16:52:50 PST d: [g2mainh] <G2HA\_Dispatcher> G2HA\_Dispatcher\_Thread::run() - Signalling print job

2013-07-09 16:52:50 PST i: [g2printh] <CStreamPrinterData::sendPrintJob()> G2PrintServer::DisplayPrintJobInfo\_

2013-07-09 16:52:50 PST i: [g2printh] <CStreamPrinterData::sendPrintJob()> Drivername: Canon iR3235/iR3245 UFR II

2013-07-09 16:52:50 PST i: [g2printh] <CStreamPrinterData::sendPrintJob()> G2PrintCore::sendInMemoryMsg() -- jobID[48059] length[1560] --

2013-07-09 16:52:50 PST i: [g2printh] <CStreamPrinterData::sendPrintJobFinish()> G2PrintServer::SendPrintJobFinish() (Finished Ok)

2013-07-09 16:52

Client

2013-07-09 16:56:17 PST i: [g2viewer] <G2ViewerAgentThread> G2PrintingWrapper::connect(1) enabled=true connected=true epoch=1

2013-07-09 16:56:17 PST i: [g2viewer] <DataPipe\_RemotePrinting-main> Drivername: Canon iR3235/iR3245 UFR II

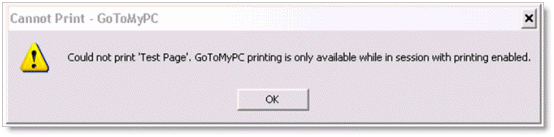
2013-07-09 16:56:17 PST i: [g2viewer] <92822> ReceivePrintJobMsgThread::Run() - One matching printer, using [\\sbapPrinters01\Printer513]

2013-07-09 16:56:18 PST i: [g2viewer] <92823> ReceivePrintJobThread::PrintRawStream(): Printed stream (2658 bytes)

Resolution: Make sure that the client print driver and the manually enabled host print driver match up

**7.7 Printing Error messages**

Symptom: User receives an error message – GoToMyPC Printing is only available while in session with printing enabled



Explanation: This error message happens when the GoToMyPC Printer Port can’t communicate with the remote printing plugin. A diagram of these communications is included below



**Remote Printing plugin (g2printh.exe) is not running**

**GoToMyPC is not running as a service**

**Print Spooler is not running on the host**

**Version number of Printer port DLL file and Remote Printing Plugin don’t match**

The remote printing plugin (g2printh.exe) should always be running in a GoToMyPC session. If the printing plugin is not running then the port monitor file won’t be able to communicate with it and you will see the error message.

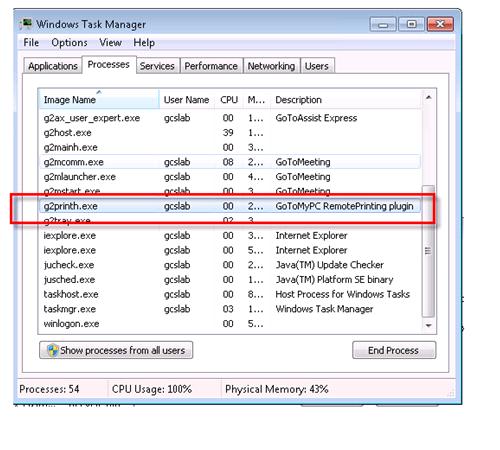
Note: If you try and print to a GoToMyPC Printer and you are not in session you will also see the same error message about printing enabled. This is because the remote printing plugin does not run out of session.

**Remote Printing Plugin (g2printh.exe) is not running**

Explanation: The remote printing plugin (g2printh.exe) should always be running in a GoToMyPC session. If the printing plugin is not running then the port monitor file won’t be able to communicate with it and you will see the error message.

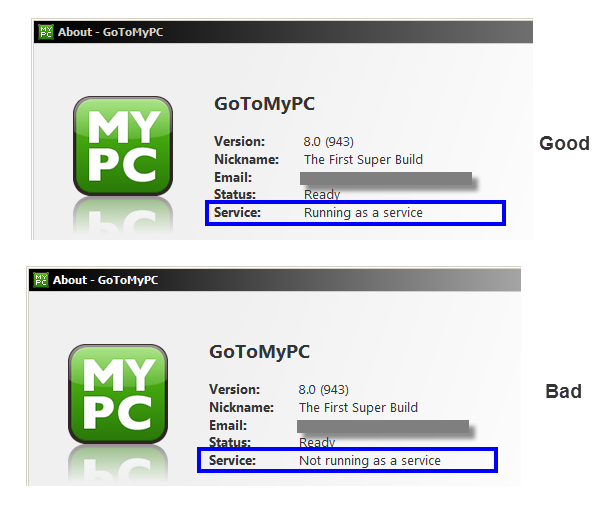
Note: If you try and print to a GoToMyPC Printer and you are not in session you will also see the same error message about printing enabled. This is because the remote printing plugin does not run out of session.

Resolution: Make sure the user is on the latest version of the GoToMyPC software. Also consider restarting the computer.



**GoToMyPC is not running as a service**

GoToMyPC needs to be running as a Windows service in order to get the print spooler to be able to load the Port file when it starts. If GoToMyPC is not running as a service the Port File won’t be loaded correctly and remote printing will fail. You can check to see if GoToMyPC is running as a service by right clicking on the MYPC icon in the system tray and selecting **“About”**



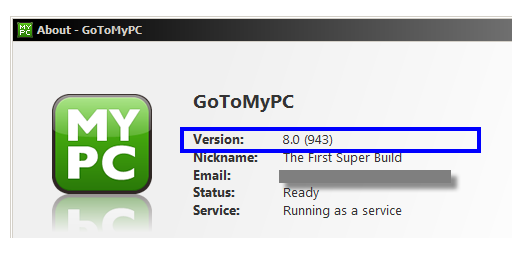
Resolution: Quit the GoToMyPC program and start the software from the windows services

**Print Spooler is not running on the host**

If the Print Spooler stops running or terminates unexpectedly and there is a job pending for a GoToMyPC Printer you will see this message. You need to start/restart the print spooler

Resolution: Restart the print spooler

**Version of Printer Port DLL and Remote Printing Plugin Don’t Match**

Explanation: If you have GoToMyPC build 943 installed, but you still have the port monitor DLL file from build 635 this will prevent communication between the port and printing plugin from working. This situation arises when you’re upgrading from one build of GoToMyPC to another. The mismatch happens when performing a build upgrade and the installer can’t update the DLL from the previous version immediately. GoToMyPC should ask the user to reboot and this should complete the update. If the user doesn’t reboot or if there is another problem this issue may arise.

You can check the version of the Port DLL file by going to the following place:

32 bit version of Windows:

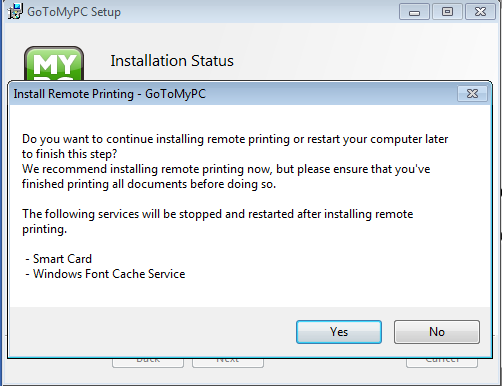
C:\Windows\System32\gotomon.dll

64 bit version of Windows

C:\Windows\System32\gotomon\_x64.dll

When GoToMyPC installs components for printing, it needs to stop the print spooler and add a DLL file for the GoToMyPC Printer port. If everything goes well, the DLL file is copied over and when the Print Spooler service is started back up the GoToMyPC Printer port will be available to handle jobs. GoToMyPC may have issues either stopping the print spooler or

Usually GoToMyPC may not be able to stop the print spooler because other Windows Services may depend on the print spooler. If for example the user has a Fax, there is a Fax Service in Windows that depends on the print spooler. This dependency makes stopping the print spooler a bit problematic. Beginning with builds 635 and up, GoToMyPC is able to detect that there are other services that depend on the print spooler and it will ask the user if she/he wants to continue with the installation anway and force the other services to stop:



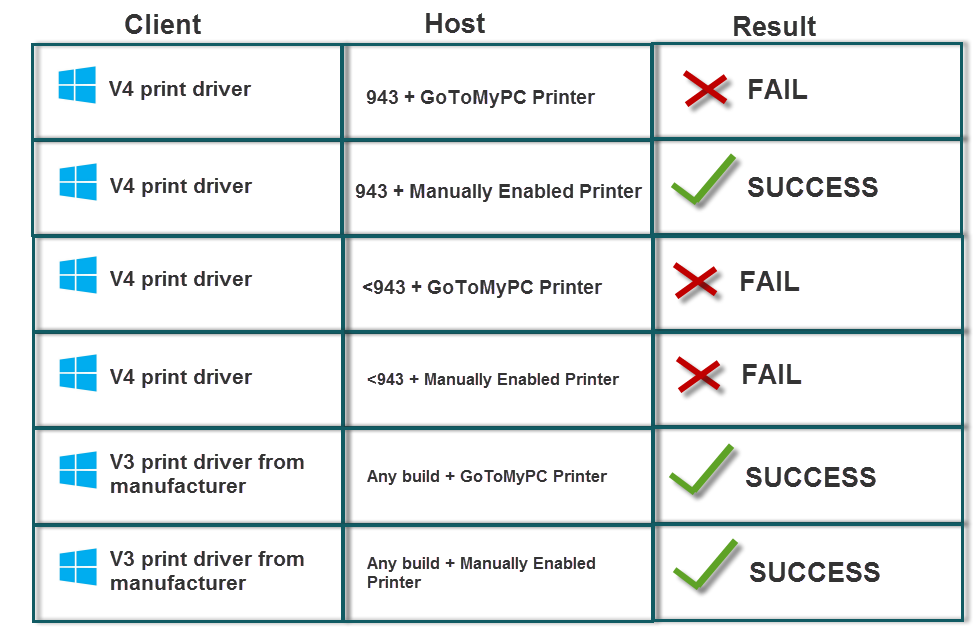
Sometimes a previous version of the GoToMyPC port is in use by the print spooler and it can’t be updated to the latest version immediately. In these situations, the installer will ask the user to restart the computer. After the restart, the latest version of the printer port DLL file will be copied over and remote printing should work successfully. No, it’s not always a good idea to restart later.



Resolution: Make sure to reboot when prompted to by the GoToMyPC installer. If you do not reboot and try to use remote printing immediately, you may encounter this issue. In rare cases you may need to stop GoToMyPC and the Print Spooler service and delete the port monitor DLL file manually before attempting to reinstall.

**7.8 Troubleshooting: Windows 8 Print Driver compatibility**

Windows 2000, XP, Vista, and 7 all used print drivers that had a common architecture. We’ll call this the version three (v3) architecture. Windows 8 introduced a new print driver architecture, a v4 architecture. Windows 8 still supports v3 drivers, however if you connect a new printer to a Windows 8 computer, Windows will automatically attempt to install a v4 driver. If you are using the remote printing feature and your client computer uses v4 print drivers, remote printing might not work. 943 has added compatibility some compatibility with these printers, however the GoToMyPC printer does not yet support these drivers. Older builds may simply not work at all with v4 print drivers. If the client computer is Windows 8 using a v4 print driver and the host computer is running Windows 2000 the only workaround is to install a v3 driver on the Windows 8 client



The next release after 943 will support v4 drivers with the GoToMyPC printer.

# 8 In session features

**8.1 Architecture: Screen Blanking**

With the latest build of GoToMyPC, the software has two possible methods of blanking out the host computer screen. The default method of blanking out the host computer screen is by using a screen blanking driver. A backup method of blanking out the screen using hardware overlays is also available

**Screen Blanking Driver Method**

A screen blanking driver named monblanking.sys is installed on the user’s system. When a connection starts, the screen blanking driver sends a command to the user’s display adapter essentially saying “**go to sleep**” (it activates display sleep as if there was no video signal) . This is how it looks from the logs:

2012-08-02 17:53:09 PST d: [g2tray] <CheckLocked> COnCheckLocked::turnScreenBlankingOnUsingDriver() - Sending Blank ON signal to monitor

2012-08-02 17:53:09 PST d: [g2tray] <CheckLocked> MonBlanking::SetBlanking(): Resetting SetThreadExecutionState to ES\_DISPLAY\_REQUIRED

Once the session is over a signal saying **“wake up**” to the display adapter.

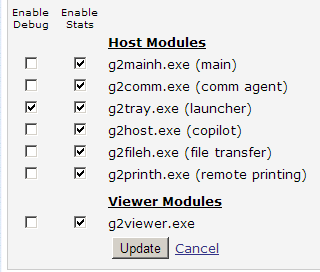
2012-08-02 17:53:45 PST i: [g2tray] <CheckLocked> COnCheckLocked::turnScreenBlankingOFF() - Turning Screen Blanking OFF

Hardware Overlay Method

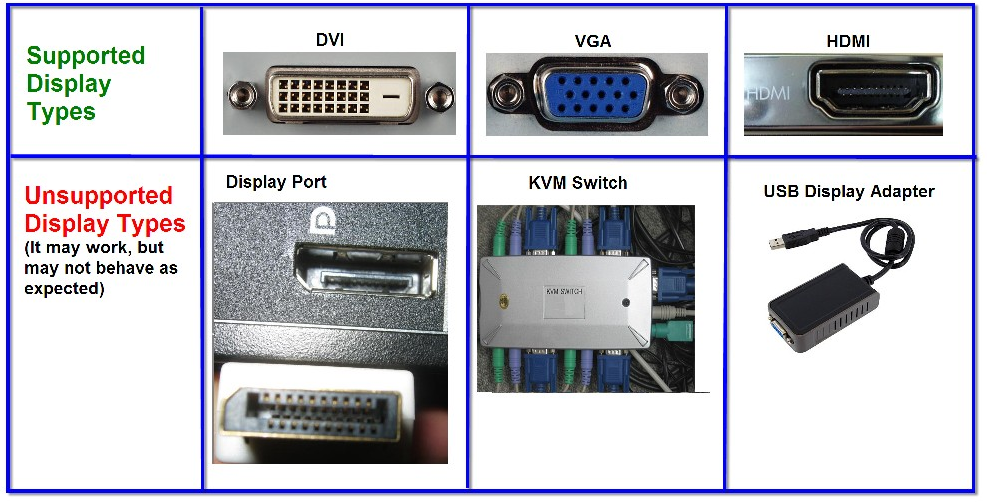
GoToMyPC tells the video card to draw a “splotch” of color over the user’s display in an overlay layer. The idea is that the top layer that the monitor will show will be a green or black window while GoToMyPC is capturing the layer the user’s desktop is on and it shows this to the user below.

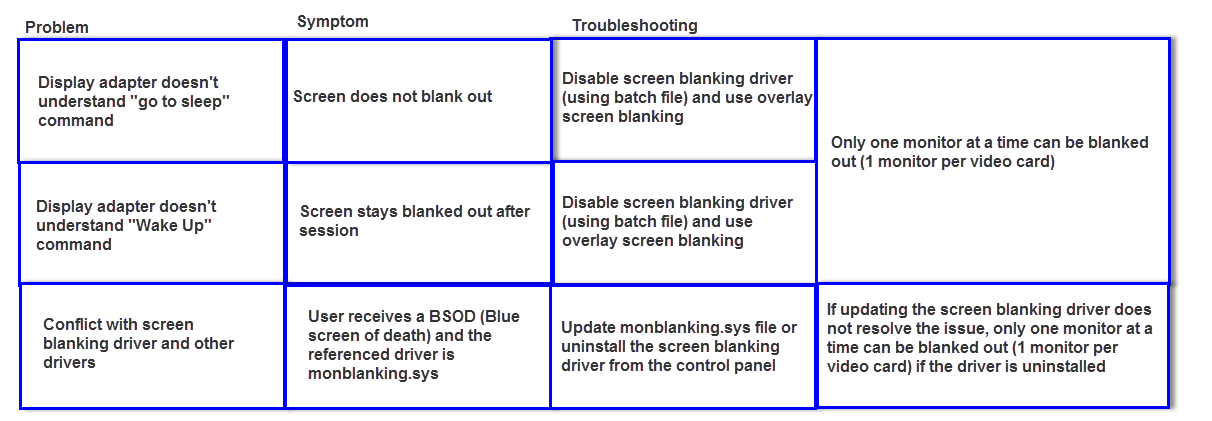
**8.2 Troubleshooting Screen Blanking**

**Note:** Troubleshooting screen blanking requires that debug logging be enabled for a user’s account in the GoToMyPC Internal Admin.



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Supported Operating Systems** | **Windows 2000** | **XP** | **Vista** | **Windows 7** | **2008 Server** | **Windows 8** |
| **Overlay** | Yes | Yes | Yes | Yes | No | No |
| **Screen Blanking Driver** | Not available | Yes | No | Yes | No | Yes |

**



**Troubleshooting the screen blanking Driver**

You can make sure that the screen blanking driver is running properly on a user’s system. If it is not, you may restart the computer and ensure that it is running

Click Start

Click Run or click the search box

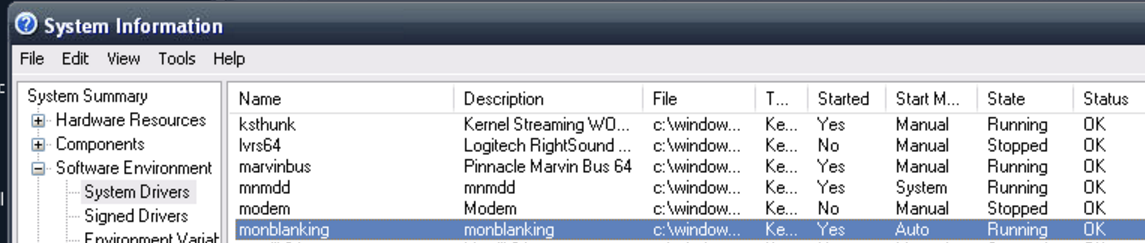
Type msinfo32 and press enter

Click to expand Software Environment

Click System Drivers

Locate monblanking

Make sure that the monitor blanking file is **Started** and **Running**



If the screen blanking driver is running, but is not working as expected you may run a batch file to configure GoToMyPC to use the Overlay screen blanking method instead.

**The Batch file:**

The batch file prevents GoToMyPC from using the screen blanking driver to blank out the screen when a session starts. The screen blanking driver is still installed and runs on the user’s system even after it’s used.

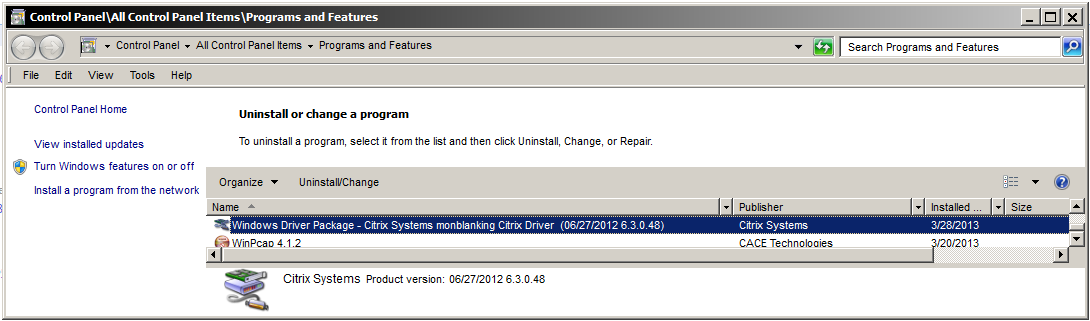
Download the batch file: <https://citrix.sharefile.com/d/sb724129b0174eba8>

Right click on the batch file and select Run as Administrator

Enter 1 to disable the screen blanking driver



To remove the screen blanking driver from a user’s system and prevent it from being loaded, you can uninstall it from the control panel.



**Troubleshooting Hardware Overlay Screen Blanking**

Troubleshooting hardware overlat issues can only be done with log files.

**Known Issues:**

**Only one monitor per video card is supported**

**Some video cards will not blank out at all and not report any error messages**

**Windows 8 does not support this screen blanking method**

**Only one monitor per video card is supported**

Explanation: Due to hardware limitations, almost every video card will only support blanking out one monitor. If a user wants to blank out two monitors using hardware overlays, the user will need to have two video cards. In the log file lines below, the video card was able to blank out the first monitor, but not the second.

2012-09-03 16:15:58 PST d: [g2tray] <CheckLocked> CDXOverlay[1]::enable - NVIDIA GeForce GTX 260

2012-09-03 16:15:58 PST d: [g2tray] <CheckLocked> CDXOverlay[1]::InitDirectDraw - NVIDIA GeForce GTX 260

2012-09-03 16:16:00 PST d: [g2tray] <CheckLocked> SURFACE: flags[4] fourcc[961893977] bitdepth[9] rmask[0]

2012-09-03 16:16:00 PST d: [g2tray] <CheckLocked> CDXOverlay[2]::enable - NVIDIA GeForce GTX 260, \\.\DISPLAY2

2012-09-03 16:16:00 PST d: [g2tray] <CheckLocked> CDXOverlay[2]::InitDirectDraw - NVIDIA GeForce GTX 260, \\.\DISPLAY2

2012-09-03 16:16:00 PST i: [g2tray] <CheckLocked> CDXOverlay[2]::InitDirectDraw - MAX: 1 CUR: 0

2012-09-03 16:16:01 PST d: [g2tray] <CheckLocked> CDXOverlay[2]::InitDirectDraw - No Overlay PixelFormat found

2012-09-03 16:16:01 PST d: [g2tray] <CheckLocked> CDXOverlay[2]::enable - InitDirectDraw failed

Resolution**:** Use the screen blanking driver method if available. If not, an additional video card is required.

**Some video cards will not blank out at all and not report any error messages**

Explanation: In the log file lines below, the video card is not reporting any error messages, but it is still not blanking out

2011-07-20 12:34:49 PST d: [g2tray] <CheckLocked> CDXOverlay[1]::enable - NVIDIA Quadro NVS 290

2011-07-20 12:34:49 PST d: [g2tray] <CheckLocked> CDXOverlay[1]::InitDirectDraw - NVIDIA Quadro NVS 290

2011-07-20 12:34:49 PST i: [g2tray] <CheckLocked> CDXOverlay[1]::InitDirectDraw - MAX: 1 CUR: 0

2011-07-20 12:34:49 PST d: [g2tray] <CheckLocked> SURFACE: flags[4] fourcc[842094158] bitdepth[0] rmask[0] gmask[0] bmask[0] amask[0]

2011-07-20 12:34:50 PST d: [g2tray] <CheckLocked> SURFACE: flags[4] fourcc[875714126] bitdepth[0] rmask[0] gmask[0] bmask[0] amask[0]

2011-07-20 12:34:50 PST d: [g2tray] <CheckLocked> SURFACE: flags[4] fourcc[842094169] bitdepth[0] rmask[0] gmask[0] bmask[0] amask[0]

2011-07-20 12:34:50 PST d: [g2tray] <CheckLocked> SURFACE: flags[4] fourcc[1498831189] bitdepth[0] rmask[0] gmask[0] bmask[0] amask[0]

2011-07-20 12:34:50 PST d: [g2tray] <CheckLocked> UYVY Selected[128]

2011-07-20 12:34:50 PST d: [g2tray] <CheckLocked> SURFACE: flags[4] fourcc[844715353] bitdepth[0] rmask[0] gmask[0] bmask[0] amask[0]

2011-07-20 12:34:50 PST i: [g2tray] <CheckLocked> SURFACE: flags[4] fourcc[1498831189][UYVY] bitdepth[0] rmask[0] gmask[0]

Resolution: Use the screen blanking driver method instead

**Windows 8 does not support this screen blanking method**

Resolution: Switch to the screen blanking driver method instead

**Other Issues**

Symptom: Some video card drivers behave erratically when processing the screen blanking request and can cause the entire GoToMyPC session to behave erratically or crash.

Log file lines:

*2012-03-27 07:14:12 PST i: [g2tray] <CheckLocked> CDXOverlay[1]::InitDirectDraw - MAX: 1 CUR: 0*

*2012-03-27 07:14:14 PST i: [g2tray] <CheckLocked> CDXOverlay[1]::InitDirectDraw - MAX: 1 CUR: 0*

*2012-03-27 07:14:16 PST i: [g2tray] <CheckLocked> CDXOverlay[1]::InitDirectDraw - MAX: 1 CUR: 0*

Resolution: Switch to the screen blanking driver using the batch file

**8.3 Troubleshooting: File Transfer**

**8.4 Troubleshooting: Remote Sound**

**Unable to transmit sound in Windows XP**

Explanation: GoToMyPC in Windows XP is only able to transmit sound from an input device on the host computer. In order for this feature to work the host computer must have a sound card that supports a loopback feature like stereo mix, what u hear, or wave out mix. If the sound card doesn’t support loopback, you can still transmit sound using a loopback cable.

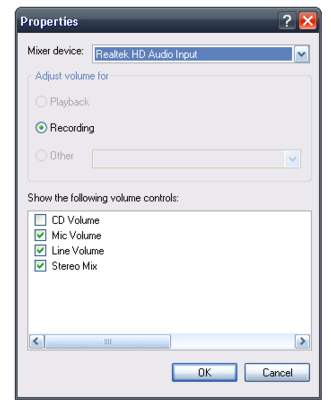
Double click on the sound icon in the system tray

Click Options and select properties

Select the soundcard’s input device in the mixer device drop down menu

See if Stereo Mix, Wave out Mix, or What u hear are available

If these options are not available, the user may need a hardware resolution



Hardware Resolution

A loopback cable is required



Plug one end of the loopback cable into the line out jack and another end into the line in jack

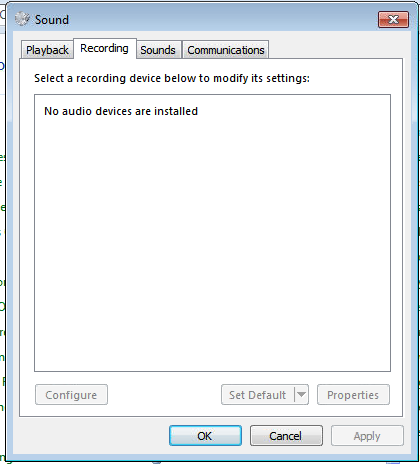


**Customer Symptom: Unable to transmit sound from Windows Vista/7/8**



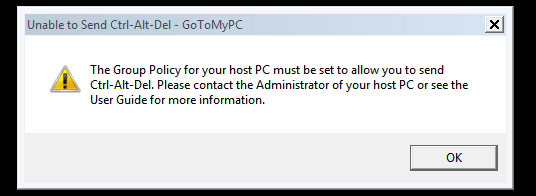
Explanation: GoToMyPC still requires that a sound card in order for the remote sound feature to work with Windows Vista/7/8 operating systems, but it no longer requires the sound card to support audio loopback.

Resolution: Make sure that the computer has a sound card. If a sound card is not installed one will need to be installed in order for the feature to work.

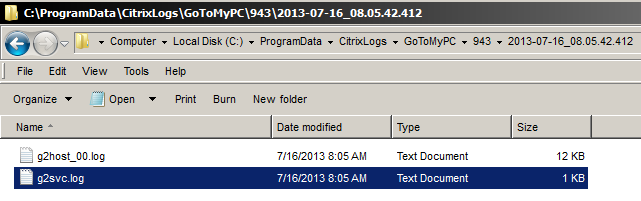


**8.5 Troubleshooting: CTRL ALT DEL**

Symptom: User receives an error message about Group Policy when attempting to send CTRL ALT DEL or sending CTRL ALT DEL does not work at all, or it may work inconsistently



To troubleshoot this we will need the g2svc.log. This log is available in the same folder as the host log file.



The technical name for CTRL ALT DEL is the Secure Attention Sequence (SAS). Normally you initiate this by pressing the CTRL ALT DEL keys on the keyboard. Programs that are running as a service also have the ability to send the SAS. Beginning with Windows Vista or later, there are policies that are available that will determine which programs are able to do this. GoToMyPC is able to send CTRL ALT DEL if the policy is disabled or not configured. If a user enables the policy and sets it to none, no program will be able to send the SAS.

Log file lines:

CTRL ALT DEL sent successfully

013-07-16 07:46:58 PST i: [g2svc] <CStartHereLoader::sendSAS()> CStartHereLoader::i\_sendSAS() - Sending SAS from service (SAS policy = 3, UAC = 1)

SAS Policies

=3  Enabled

=-1 SAS is either disabled or not configured

=0 SAS policy is enabled and no programs can send CTRL ALT DEL

UAC

=0 User Account Controls are off

=1 User Account Controls are on

Can't send CTRL ALT DEL due to group policy

2013-07-16 07:58:30 PST i: [g2svc] <CStartHereLoader::sendSAS()> CStartHereLoader::i\_sendSAS() - Could not send SAS (SAS policy = 0, UAC = 0)

Resolution: Enable User Account Controls and reboot. Configure Windows SAS policy to enabled for services/ease of access applications.

Click Start

Click in the Search Bar

Enter: **gpedit.msc**

Computer Configuration > Administrative Templates > Windows Components > Windows Logon Options > Disable or enable



# 9 Host OfflinE

**9.1 Causes of host computer going offline**

**User quit GoToMyPC software**

**Internet connectivity issues at the host computer**

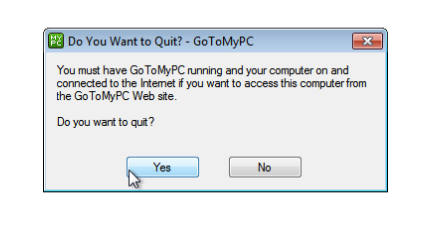
**Computer goes into sleep/standby/power saving mode**

**GoToMyPC software crashes**

User quit GoToMyPC Software

Customer symptoms: GoToMyPC is offline after the GoToMyPC Session Ends

Explanation: Some users accidentally quit the entire GoToMyPC program when they are attempting to end a GoToMyPC session with the host



To confirm this behavior you will need to look at the log files.

Log file lines:

2013-07-10 08:00:57 PST i: [g2pre] <PreLauncher> CPreLauncherThread::\_stopLauncher() - Launcher status is stopped.

2013-07-10 08:00:57 PST i: [g2tray] <167382/IPlugin.dispose> CG2HostLauncher\_PI::~CG2HostLauncher\_PI()

2013-07-10 08:00:57 PST i: [g2tray] <167382/IPlugin.dispose> CMainThread::~CMainThread()

2013-07-10 08:00:57 PST i: [g2comm] <CPluginManagerIpcAdapter::beforeModuleQuit()> beforeModuleQuit, pid: 3076

2013-07-10 08:00:57 PST i: [g2comm] <CPluginManagerIpcAdapter::beforeModuleQuit()> DependencyWatcher, process quitting: g2tray.exe[0]

2013-07-10 08:00:57 PST i: [g2pre] <167381/IPluginControl.stop> CG2HostPreLauncher::stop()

2013-07-10 08:00:57 PST i: [g2pre] <167381/IPluginControl.dispose> CG2HostPreLauncher::~CG2HostPreLauncher()

2013-07-10 08:00:57 PST i: [g2comm] <CPluginManagerIpcAdapter::beforeModuleQuit()> beforeModuleQuit, pid: 240

2013-07-10 08:00:57 PST i: [g2comm] <CPluginManagerIpcAdapter::beforeModuleQuit()> DependencyWatcher, process quitting: g2pre.exe[0]

2013-07-10 08:00:57 PST i: [g2comm] <Main> CCommAgentApp::startInitialPluginAndWait() - Waiting for initial plugin to quit (PID 240)...

Resolution: Educate the user on the proper way to disconnect

**System Standby/Sleep**

Issue: GoToMyPC is offline when away from the host but after returning to the host the computer shows online and ready to connect

Explanation: This may happen if the host computer goes into sleep mode. By default, GoToMyPC will disable the sleep mode on the host computer. This does not prevent the user from putting the computer to sleep manually. If the host computer is a laptop, closing the lid may also cause the laptop to go to sleep. The GoToMyPC logs don’t capture standby/sleep events specifically, but when a sleep/standby event happens things look out of place in the logs. You may notice that during sleep/standby nothing is written in the logs for an extended period of time. GoToMyPC can’t log anything because the system is in standby. The software can’t notice that the connection to the poll server has been broken until the system wakes up from sleep. Once the system wakes up from sleep GoToMyPC can attempt to reconnect it.

2013-07-10 07:30:24 PST d: [g2comm] <1411/IINet.probe> Probe connect: Start connect to 68.64.25.250

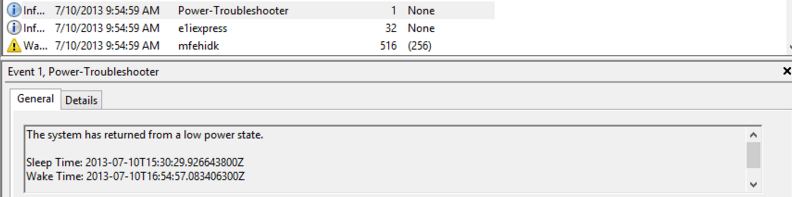
2013-07-10 08:54:56 PST i: [g2comm] <1301/IPersistentPollConnection.waitForNotification> CPersistentPollConnection::waitForNotification - connection broken

(2024) "ECError::eTimeout"

Connection dead

2013-07-10 08:54:56 PST d: [g2tray] <OnPersistent(ECMethodTask)(0)> CPersConn::disconnect()

Windows event viewer may also be used to identify sleep/standby events.



**Internet connectivity issues at the host computer**

Issue: GoToMyPC is offline at random times during the day.

Explanation: A host computer will show as offline if it can’t communicate to the poll server. If the host computer can’t talk to the poll server it will keep trying to re-establish the connection. The GoToMyPC host software does not know why the connection fails. It may be because the user does not have a working internet connection. It may be because either firewall or security software has blocked GoToMyPC. GoToMyPC will try all available ports and also look for proxy settings in hopes of finding a successful connection method. This process is found in the log files below and it will keep repeating until a successful connection is found:

2013-05-18 17:32:58 PST i: [g2comm] <PollThread> JConnSpecProviderCda: CDA requested

2013-05-18 17:32:58 PST W: [g2comm] <3297/IPersistentPollConnection.connect> CPersistentPollConnection::connect - failed to connect to Poll Server

(4002) "ECNetworkError::eConnectError"

Error returned at

, persistentpollconnection.cpp:253

2013-05-18 17:32:58 PST i: [g2tray] <OnPersistent(ECMethodTask)(0)> CPersConn::connect() - failed to connect

(4002) "ECNetworkError::eConnectError"

Error returned at

, persistentpollconnection.cpp:253

Error returned at , persistentpollconnectionipcadapter.cpp:98

Error returned at , persistentpollconnectionipcproxy.cpp:99

2013-05-18 17:32:58 PST i: [g2tray] <OnPersistent(ECMethodTask)(0)> COnPersistent: waitForNotification returned unknown error code

(4002) "ECNetworkError::eConnectError"

Error returned at

, persistentpollconnection.cpp:253

Error returned at , persistentpollconnectionipcadapter.cpp:98

Error returned at , persistentpollconnectionipcproxy.cpp:99

2013-05-18 17:32:58 PST i: [g2comm] <3297/ICommPipe.newPersistentPollConnection> comm::jinet::JSpecProviderBroker::getJediProvider(): Matched the singleton connection spec provider

2013-05-18 17:32:58 PST i: [g2comm] <PollThread> Establishing persistent connection to Poll server

2013-05-18 17:32:59 PST i: [g2comm] <cda2> Trying 66.151.158.177:8200 (proxy: None=0.0.0.0:0<ip>) [\_]

2013-05-18 17:32:59 PST i: [g2comm] <cda2> Detect: TCP connect to 66.151.158.177:8200 (proxy: None=0.0.0.0:0<ip>) [\_]

2013-05-18 17:32:59 PST i: [g2comm] <cda2> Trying 66.151.158.177:80 (proxy: None=0.0.0.0:0<ip>) [\_]

2013-05-18 17:32:59 PST i: [g2comm] <cda2> Detect: TCP connect to 66.151.158.177:80 (proxy: None=0.0.0.0:0<ip>) [\_]

2013-05-18 17:32:59 PST i: [g2comm] <cda2> Trying 66.151.158.177:443 (proxy: None=0.0.0.0:0<ip>) [\_]

2013-05-18 17:32:59 PST i: [g2comm] <cda2> Detect: TCP connect to 66.151.158.177:443 (proxy: None=0.0.0.0:0<ip>) [\_]

2013-05-18 17:33:09 PST i: [g2comm] <cda2> Detected 0 connections to 66.151.158.177, source: IE;Firefox;Netscape;Registry;WPAD script

2013-05-18 17:33:09 PST i: [g2comm] <cda2> CDA to 66.151.158.177, failed

2013-05-18 17:33:19 PST i: [g2comm] <PollThread> JEDI connect: Connect to 66.151.158.177 failed

(4002) "ECNetworkError::eConnectError"

Connect failed

You can use these to determine when a host had internet connectivity and when it didn’t. You can also use the log files to get an understanding of how reliable a user’s internet connection is. Periodically a report of the success rate of the connection probes is included in the log files. These reports can be a good way of measuring how stable a user’s internet connection is. If 0 connection probes failed it suggests that the user has a stable internet connection for the period of time the report covers.

2013-06-13 02:22:36 PST s: [g2comm] <105/IINet.probe> PROBE: 241 probes succeeded and 0 probes failed since the last report

2013-06-13 03:22:36 PST s: [g2comm] <163/IINet.probe> PROBE: 260 probes succeeded and 0 probes failed since the last report

2013-06-13 04:22:36 PST s: [g2comm] <214/IINet.probe> PROBE: 260 probes succeeded and 0 probes failed since the last report

2013-06-13 05:22:36 PST s: [g2comm] <269/IINet.probe> PROBE: 260 probes succeeded and 0 probes failed since the last report

2013-06-13 06:22:36 PST s: [g2comm] <328/IINet.probe> PROBE: 260 probes succeeded and 0 probes failed since the last report

2013-06-13 07:22:36 PST s: [g2comm] <380/IINet.probe> PROBE: 260 probes succeeded and 0 probes failed since the last report

This is from the same log file. Three hours later in the day the user appears to be experiencing some slight packet loss and some of the connection probes have failed:

2013-06-13 10:23:42 PST s: [g2comm] <622/IINet.probe> PROBE: 255 probes succeeded and 5 probes failed since the last report

If a large number of connection probes fail it suggests that the user’s internet connection is intermittent:

2013-06-13 14:55:30 PST s: [g2comm] <950/IINet.probe> PROBE: 109 probes succeeded and 151 probes failed since the last report

Eventually the user’s internet connection has completely dropped off:

2013-06-13 15:55:30 PST s: [g2comm] <1014/IINet.probe> PROBE: 0 probes succeeded and 260 probes failed since the last report

Resolution: Advise the user to adjust her or his internet configuration or contact the internet service provider.

**GoToMyPC Software crashes**

Issue: Host Computer is offline and the MYPC icon is missing from the menu bar when the user goes to the host computer.

Explanation: This may happen if the GoToMyPC host software has experienced a crash. An easy way to see if the GoToMyPC software has experienced a crash is to search for **mdump** in the log file. A crash is when a program tries to do something that is not allowed by the operating system. If the crash or “exception” can either be handled by the program that had the crash or it can be handled by Windows. When I say handled I mean resolved or addressed. If the crashing program can’t handle the crash then Windows will do it. If Windows does it, it will forcefully close the program that’s crashing and a message will be displayed to the user.



If the application is able to handle the crash, it will shut down and try and recover gracefully. The process will not be visible to the user. GoToMyPC has an exception or crash handler. The exception handler’s job is to identify a crash, capture diagnostic information for the crash in the log files, and then try and restart the part of the GoToMyPC software that crashed if it can. The exception handler is able to address nearly every crash that happens with GoToMyPC so a user will almost never see any pop up from Windows about the host software crashing.

Although the exception handler is able to deal with crashes and prevent the user from seeing crash messages, it is not always able to get the GoToMyPC host software to recover from the crash and resume its normal activities. In the case below, the remote logging part of the GoToMyPC software crashed. After this crash, the exception handler was able to restore GoToMyPC to working order and the user was able to connect to the host again.

010-05-06 17:24:46 PST C: [g2mainh] <G2HostAgentShutdownWatchdog> \*\*\* Crash \*\*\*

2010-05-06 17:24:46 PST C: [g2mainh] <G2HostAgentShutdownWatchdog> Exception: C0000005 at 00000000 in

2010-05-06 17:24:46 PST C: [g2mainh] <G2HostAgentShutdownWatchdog> ACCESS\_VIOLATION while reading 00000000

2010-05-06 17:24:46 PST C: [g2mainh] <G2HostAgentShutdownWatchdog> Loaded dbghelp.dll ver. 6.00.6001.18000

2010-05-06 17:24:46 PST C: [g2mainh] <G2HostAgentShutdownWatchdog> Crash dump data successfully generated.

2010-05-06 17:24:46 PST C: [g2mainh] <G2HostAgentShutdownWatchdog> MDUMP BEGIN 13304

2010-05-06 17:24:46 PST C: [g2mainh] <G2HostAgentShutdownWatchdog> MDUMP=eNrtfQtcVMX++Jyzq6KiEqKiJqEijzLDN3nptgoi+AgUFDRKQFDQBVbYwyIRYlIXE42KCh8hFeWLlEoL04i6aKtSoX9SKisMM3xcQ8U3j9/3O3PO7tkVVt3u73O7vz/DZ5iZ73wfM9/5znfmnDPn7AzfGUGvbl4aYUMIcSYs/LykbpqYJQqIpzhCciAqIW9rR0g6ADtBfhdEJx9COkO6BfEgdoHoBTETYi+IwyGWEkuhM+nci+c6i6WkfoTwkGb3hn8gc0pMghCXEBOQoI1RXwWk1lvnPs+

But if you look further down in the same log, the exception handler was able to prevent the crash from causing GoToMyPC to go offline.

2010-05-07 13:51:52 PST i: [g2tray] <OnPersistent(CTask)(0)> COnPersistent::poll() poller received connect request.

2010-05-07 13:52:00 PST i: [g2mainh] <G2HostAgentThread> Session AUTHENTICATED -- starting screen sharing --

2011-12-17 17:08:03 PST i: [g2comm] <cda2> Detected 0 connections to 66.151.158.177, source: IE;Firefox;Netscape;Registry

2011-12-17 17:08:03 PST i: [g2comm] <3506> Inside globalExceptionHandler

2011-12-17 17:08:03 PST C: [g2comm] <3506> \*\*\* Crash \*\*\*

2011-12-17 17:08:03 PST C: [g2comm] <3506> Exception: C0000005 at 758B7911 in

2011-12-17 17:08:03 PST C: [g2comm] <3506> ACCESS\_VIOLATION while reading 758B7911

2011-12-17 17:08:03 PST C: [g2comm] <3506> Loaded dbghelp.dll ver. 6.01.7600.16385

2011-12-17 17:08:04 PST C: [g2comm] <3506> Crash dump data successfully generated.

2011-12-17 17:08:04 PST i: [g2comm] <cda2> Trying 66.151.158.177:8200 (proxy: None=0.0.0.0:0) [\_]

2011-12-17 17:08:04 PST i: [g2comm] <cda2> Detect: TCP connect to 66.151.158.177:8200 (proxy: None=0.0.0.0:0) [\_]

2011-12-17 17:08:04 PST i: [g2comm] <cda2> Trying 66.151.158.177:443 (proxy: None=0.0.0.0:0) [\_]

2011-12-17 17:08:04 PST i: [g2comm] <cda2> Detect: TCP connect to 66.151.158.177:443 (proxy: None=0.0.0.0:0) [\_]

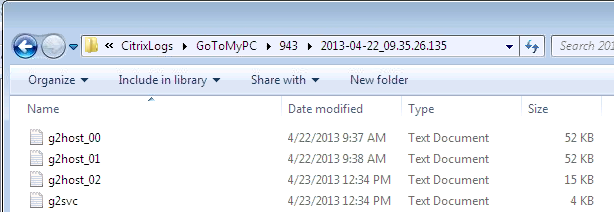
2011-12-17 17:08:04 PST i: [g2comm] <cda2> Trying 66.151.158.177:80 (proxy: None=0.0.0.0:0) [\_]

2011-12-17 17:08:04 PST i: [g2comm] <cda2> Detect: TCP connect to 66.151.158.177:80 (proxy: None=0.0.0.0:0) [\_]

2011-12-17 17:08:04 PST C: [g2comm] <3506> MDUMP BEGIN 28708

2011-12-17 17:08:04 PST C: [g2comm] <3506> MDUMP=eNrtfQ1clMXW+NldUBRSTEs0TDRTVDQUMzJNRBQktc2PRMEEAQVF4MKuiwFKQUWJQqlhiV5KTUorKjP8umFRUZlaUYL4QWpeKitUEkQ+/ufMzLP77MOyCb7/e31/L6vDzJw5c86ZmTNnPp/nmeYzTbtuR36IHQC4AP9Vj/19ugiCBl2JLYCfGsAGw0lOAK6IjCCYhbC71gF0wHAeulwVQEf0PdGloOuKrhJdAVj7dYDOzjYqooHZIb8PAJKFbS4c4BserY+MDp8SrQuP+qqDCpqu//bxuTP3G4huDbosZDJwysXpt2P4TnQrO5icStAEL4A4wY3iFfhHCyEQCpGwCF0olnwm6BASDWH4Nw7/usAsTFkG4S3Kbc+oqWUQJU0fjK+AKIwthgik//c0qbbVRppFTU1NbvY8jP

If you see multiple log files within the same host log folder, this usually happens because some part of the GoToMyPC program crashed and GoToMyPC had to relaunch itself.



Resolution: Relaunch GoToMyPC. Depending on how severely a user is affected, you may need to escalate the case so that engineering can investigate the source of the crash.

Note: A batch file is available and can be run as a scheduled task to automatically restart the GoToMyPC software at certain times. This should only be used in circumstances where no other option is available.

**9.2 Using the g2svc log for troubleshooting**

The g2svc log contains some useful tools for tracking Windows related events that may affect GoToMyPC

User quit GoToMyPC

2013-07-16 08:18:19 PST i: [g2svc] <Watcher(ECMethodTask)(1)> CG2Service::watch() - The Comm Agent exited without error (exit code = 0)

2013-07-16 08:18:20 PST d: [g2svc] <Service> CStartHereLoader::stop()

2013-07-16 08:18:20 PST d: [g2svc] <Loader> <<< ECService::mainService()

2013-07-16 08:18:20 PST d: [g2svc] <Loader> Revoke 14 Interface Marshalers:

GoToMyPC closed due to Windows shutdown

013-07-16 07:56:20 PST d: [g2svc] <Loader> ECService::ctrlHandler(SERVICE\_CONTROL\_SHUTDOWN)

2013-07-16 07:56:20 PST d: [g2svc] <Loader> CStartHereLoader::notifyWindowsShutdown()

2013-07-16 07:56:21 PST i: [g2svc] <Watcher(ECMethodTask)(1)> CG2Service::watch() - The Comm Agent exited without error (exit code = 0)

2013-07-16 07:56:21 PST d: [g2svc] <Service> CStartHereLoader::stop()

2013-07-16 07:56:21 PST d: [g2svc] <Loader> <<< ECService::mainService()

2013-07-16 07:56:21 PST d: [g2svc] <Loader> Revoke 14 Interface Marshalers:

User locked the computer

2013-07-16 08:39:35 PST d: [g2svc] <Loader> WTS\_SESSION\_LOCK message received. Session id = 1

2013-07-16 08:39:35 PST d: [g2svc] <Loader> CStartHereLoader::notifySessionLock(1)

User unlocked the computer

2013-07-16 08:39:41 PST d: [g2svc] <Loader> WTS\_SESSION\_UNLOCK message received. Session id = 1

2013-07-16 08:39:41 PST d: [g2svc] <Loader> CStartHereLoader::notifySessionUnlock(1)

User logged off

2013-07-16 08:40:43 PST d: [g2svc] <Loader> WTS\_SESSION\_LOGOFF message received. Session id = 1

2013-07-16 08:40:43 PST d: [g2svc] <Loader> CStartHereLoader::notifySessionLogoff(1)

User logged on

2013-07-16 08:41:45 PST d: [g2svc] <Loader> WTS\_SESSION\_LOGON message received. Session id = 2

2013-07-16 08:41:45 PST d: [g2svc] <Loader> CStartHereLoader::notifySessionLogon(2)

# 10 GoToMyPC Installation

**10.1 Architecture: GoToMyPC Installer**

With build 943, GoToMyPC uses windows installer to handle installs and uninstalls. The installer file, gosetup.exe doesn’t directly install GoToMyPC. It unpacks the file GoToMyPCSetup.msi and it launches it with the appropriate settings. The settings it uses are sometimes called switches or transformations. The default setting for the installer is English, but if the user’s language is Japanese a language transformation is applied to the installer. GoToMyPC is a 32 bit application, but if the user is running on a 64 bit system, the software will transform the installer to install the 64 bit version of the screen blanking driver. Once all of these settings are checked the GoToMyPC installer will be called up to perform the installation.

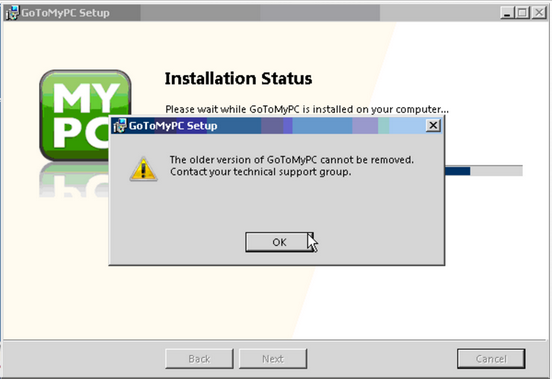
# 

**10.2 Troubleshooting: GoToMyPC Installer**

Customer Symptom: The installation failed

Explanation: This happens if GoToMyPC is unable to apply

Customer Symptom: The older version of GoToMyPC could not be removed when uninstalling or updating GoToMyPC



Explanation: This may happen if the Windows installer archive or registry keys for the GoToMyPC software become corrupt.

Resolution: Download Microsoft’s fix it tool to remove the GoToMyPC installation.

Navigate to: http://support.microsoft.com/mats/program\_install\_and\_uninstall/

Click Run Now

Download and run the MicrosoftFixitProgramInstallUninstall exe file

Click Accept

Click Detect problems and let me select the fixes to apply

Click Uninstalling

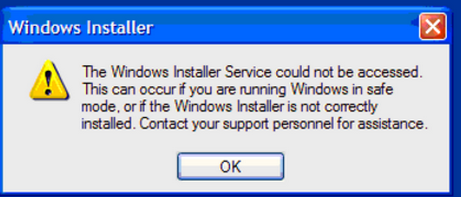
Click GoToMyPC

Click Next

The FixIt utility will remove GoToMyPC. You may close the Window when it reports that it is finished. After GoToMyPC is removed you may install the latest version of the GoToMyPC software if desired.

Installation Failed

Customer Symptom: Windows Installer could not be accessed



Explanation: This may happen if there is a problem with the Windows installer service or if you attempt to install or remove GoToMyPC in safe mode.

Resolution: Make sure that you are attempting to install GoToMyPC in normal mode instead of safe mode. If you are running Windows XP or Windows Vista, make sure that Windows Installer

Windows Installer upgrade for Windows XP and Vista users: <http://www.microsoft.com/en-us/download/details.aspx?id=8483>

**Note:** If a user is running Windows Vista SP1, Windows 7, or Windows 8 and receives this message the user must find a way to repair Windows installer before GoToMyPC can be installed

**Note:** Windows Vista Service Pack 1 users already have this version of Windows Installer and the download will not overwrite an existing installation

# 11 Direct Connections