

WideFS User Guide

A network application interface for Microsoft Flight Simulator

by Pete Dowson, 11^h April 2011

Support Forum: [Pete Dowson's Support Forum](#)

Versions 6.78 and 7.xx of WideServer. (and WideClient 6.86)

(Version 7.xx is for FSX and beyond, version 6.xx for FS2004 and earlier. This only affects the Server. The Client program, WideClient, *is the same in all cases* and is currently at version 6.86)

WIDEFS IS NOT FREE!

For FS2004 and before, WideFS 6 will *not* run on Flight Simulator without FSUIPC also being installed, and at least Version **3.53** of FSUIPC is required. WideFS **must** be registered with FSUIPC, using an Access Key, before it will work.

For FSX and beyond, WideFS 7 consists of the WideServer component built into FSUIPC4, *and* the same version 6 WideClient. No changes at the Client end are needed. However, as with version 6, WideFS **must** be registered with FSUIPC4, using a *different* Access Key, before it will work.

You can purchase access keys from <http://secure.simmarket.com/pete-dowson-widefs6.phtml> (WideFS6) or <http://secure.simmarket.com/pete-dowson-widefs7.phtml> (WideFS7). Full details about purchasing access keys and registering both FSUIPC and WideFS are also to be found in the early pages of the appropriate FSUIPC User Guide. You can save money if you purchase keys for FSUIPC and WideFS at the same time.

The full WideFS package contains the following parts:

WideServer.dll	The WideFS6 server module, for the FS Modules folder (<i>Version 6.78</i>) (This is not used with FSX or later)
WideServer.ini	Initial configuration for the server (This is not used with FSX or later)
WideClient.exe	The client program, running on client PCs (<i>Version 6.86</i>)
WideClient.ini	Initial configuration for the clients
WideFS User Guide.pdf	This document, in Acrobat format
WideFS Technical.pdf	A technical reference and trouble-shooting guide
WideFS History.pdf	Details of changes in Versions 6.xx–7.xx

Note: All my Windows based FS software is always available in the latest versions from <http://www.schiratti.com/dowson>. There may be updates, additional programs and other useful notes posted from time to time on the Support forum, see link above.

IMPORTANT NOTES

This is NOT a program for linking several copies of Flight Simulator, nor will it allow scenery or cockpit views on multiple PCs. It is purely for running EXTERNAL applications on a separate PC whilst still communicating with FS as if on the same PC. For such applications to work through WideFS they need to access Flight Simulator through the interface provided and defined by FSUIPC (or originally, for FS98 programs, FS6IPC) version.

Introduction: What is WideFS?

There are now many add-on programs written to interface to Microsoft's Flight Simulator. Nearly all of these exchange information with FS by using an interface devised many years ago (originally for FS95, the first Windows version of FS) by Adam Szofran. This interface provided methods for programs running outside the Simulator, but in the same PC, to read various values from FS and write different values back. This enabled a wealth of programs such as moving maps, weather control, external instrumentation, extended autopilots and even flight management systems to be developed.

The interface, using a technique known as "Inter-Process Communication" (IPC), was extended by Adam to operate on FS98 (using a module called FS6IPC—the Microsoft internal name for FS98 was FS6). But this was the last in that line as Adam was understandably recruited by Microsoft to work from within instead of without. <G>

It was during this time that I put together my first PC network. An excellent package called 'WidevieW' became available, written by Luciano Napolitano, which allowed several installations of FS on different PCs across a Network to show different outside views from the same user aircraft in flight. I used this as a basis for making Chris Brett's EFISv2 run on a separate PC from the Flight Simulator to which it was interfacing. This was done by fooling the remote copy of WidevieW into thinking it was running with FS, whereas it was actually a program of my devising (the 'client'), and similarly another program intercepted the data at the Simulator end (the 'server'). This package was released as freeware under the name WideEFIS, for obvious reasons.

Not much later I learned a little more about network programming—but not much. Mostly I just re-used the simpler code examples in the Microsoft development kits. I used the IPX/SPX protocol because it looked simpler to understand, was fast in operation, and also because WidevieW used it too. Applying all this resulted in WideFS, a package consisting of two parts:

1. **WideClient.exe**, a program which on the one hand provides what looks like an exact replica of the FS6IPC interface, fooling FS application programs into interfacing with it thinking that it was FS98, and on the other transmitted these exchanges and received data back across the Network, using the SPX (and later TCP) protocol.
2. **WideServer.dll**, a Flight Simulator module, installed into and actually running as part of FS98, which serviced the messages and requests from WideClient and interfaced directly into the same data inside FS as did FS6IPC.

When FS2000 came out I wrote a new IPC module for that simulator, but kept not only the same interface as that devised by Adam, but also the same variable access. This was deliberate, to allow programs written for FS98 to run with FS2000 unchanged. I also made it run on FS98 itself, and extended it to CFS1, CFS2 and FS2002, keeping as far as possible complete compatibility with that FS98 standard. Since this was now a 'universal' IPC interface for Flight Simulator I called it FSUIPC, breaking away from Adam's specific FS version implications.

Nowadays FSUIPC has taken over the interfacing role inside FS, and WideServer interfaces to it, so both parts are needed even if all the application programs are running on client PCs. This is because with FS2000 and then more so with FS2002 and FS2004, access to many of the internal parts of FS, which was so easy in FS98, has become very difficult. Internally it is no longer a matter of simple memory offset and size management and mapping, but often also involves calling procedures elsewhere in FS, and following pointer chains into data private to other modules. Rather than duplicate such complex code, it is now all concentrated into FSUIPC, freeing the WideFS parts to deal predominantly with the efficient running of the Server to Client (or Server to many Clients) exchanges.

With FSX and FSUIPC4, this integration and inter-dependency reached an extreme. With the completely new way Microsoft devised for programs to work with FSX (via SimConnect), a complete re-write was needed both for FSUIPC and WideServer, so the opportunity was taken to merge them, for efficiency in both space and time. This is also more appropriate to the new, more difficult, arrangements for getting DLL's into FSX in the first place. Other DLLs were also merged into FSUIPC4 for the same reasons (AutoSave and GPSout).

So, does this answer the question, 'what is WideFS?' Maybe, if you've followed its history well enough. Otherwise, for now, just consider WideFS as a package that enables you to run Microsoft Flight Simulator applications on computers that aren't even running Flight Simulator.

And **please** don't confuse it with WidevieW. The similarity in the name is because of its history, as described. Both packages deal with spreading the simulation experience across several PCs in a network—hence the 'Wide' part. But you use WidevieW for multiple views, to see around you, and you use WideFS for additional instrumentation, moving maps, and weather control programs, running on separate PCs and lessening the burden of having too many processes on the all-important Simulation PC itself.

Good flying!

Installation

Follow these steps:

On the Flight Simulator PC [FS2004 or earlier]

1. Place WideServer.dll and WideServer.Ini in the Flight Simulator Modules folder, which is inside your main FS folder.

That's it. FS will load it automatically for you.

Note that you must also have FSUIPC.dll (version 3.53 or later) in the Modules folder. WideFS is totally dependent upon that module, which is available as a separate download. For full use of all current WideFS features you should be using FSUIPC version 3.99 or later.

You need to **Register** your WideFS6 by running the FSUIPC Installer (yes, again, even if you already installed FSUIPC), and entering the WideFS6 access key as well as the FSUIPC key if you purchased that too. (If you have not yet bought your WideFS6 key, use the link given above to go to SimMarket).

For full details of this process, please see the FSUIPC Installation and Registration Guide which was packaged inside the FSUIPC.ZIP file.

Keep a safe copy of the **FSUIPC.KEY** file (from the Modules folder). You may have to re-register if you re-install Windows, or want to use FS on another PC. But you are not restricted to using WideFS in only one FS installation.

On the Flight Simulator PC [FSX or later]

You must have installed FSUIPC4. There's no other program to be installed on the FSX computer.

You need to **Register** your WideFS7 by running the FSUIPC4 Installer (yes, again, even if you already installed FSUIPC4), and entering the WideFS7 access key as well as the FSUIPC4 key if you purchased that too. (If you have not yet bought your WideFS7 key, use the link given above to go to SimMarket).

For full details of this process, please see the FSUIPC4 Installation and Registration Guide which was packaged inside the FSUIPC4.ZIP file.

Keep a safe copy of the **FSUIPC4.KEY** file (from the Modules folder). You may have to re-register if you re-install Windows, or want to use FS on another PC. But you are not restricted to using WideFS in only one FS installation.

On each client PC

This means *all* the other PCs in your Network on which you are NOT going to be running FS itself, but on which you want to run some application that interfaces to FS.

1. Place WideClient.Exe and WideClient.ini in any desired folder (I recommend either in the same folder as your FS utility, or a dedicated folder called WideClient).
2. If you are using GoFlight equipment, and wish to connect this to the Client PC and program the buttons and dials in FSUIPC's "buttons" page, you either need to do the full install of GoFlight's recent driver release (the GFConfig program included), *or* extract the GFDev.dll from that and place this in the same folder as WideClient.exe. Alternatively, if you have a full GoFlight installation on your Server PC, find the GFconfig program folder, and copy over the GFDev.dll from there into the WideClient folder. Note that earlier versions of GoFlight's installation did not install GFDev.dll, so you may need to get an update first—a suitable version can be found on my Support Forum too.
3. Drag a shortcut for WideClient onto your desktop.

Now, when you have your Network correctly configured, you can either run WideClient (from the shortcut), THEN your utility program, or you can, for more convenience, edit the WideClient.ini file and enter the full pathname for your utility in the Run1= parameter line. WideClient will then start your application for you. Up to 18 programs can be started in this way, 9 initially and 9 when FS is ready to fly—see the section on WideClient parameters in the WideFS Technical Guide.

NOTE that simply installing WideClient on a PC and running it does nothing useful at all (except possibly getting weather from a copy of Active Sky Evolution, ASE, running on the same Client and linking to FSX via SimConnect). To make use of it you will probably be running one or more FSUIPC client programs on that PC, or simply using it to convey Button information from local devices to FSUIPC's Buttons programming options page. Another use for a WideClient installation is to act as recipient of GPSout data from the FS PC. In this case, WideClient would need configuring to send the data out on a local Serial port (to the receiving device or PC), or on a local "virtual port" link to the receiving program in the same PC. The latter can use a proprietary Virtual Port program, or (on Windows XP or earlier) the freeware MixW virtual port program, more of which later.

Configure your Network

IT IS IMPORTANT FOR ALL USERS TO READ AT LEAST PART OF THIS!

WideFS uses TCP protocol (part of TCP/IP) by default, but also supports UDP (also part of TCP/IP) and SPX (part of IPX/SPX). Just ensure the TCP/IP protocol is installed on all PCs (as it will normally be by default) and you will be set. IPX/SPX may be more efficient but it is difficult to recommend these days, as it isn't supported by Microsoft, it isn't installed by default, and it is more complicated to configure.

The Server automatically operates with any clients that connect, whether they be using TCP, UDP or SPX. You can mix them, to suit the specific individual client PCs. Better, leave the protocol choice out of the Client INI file altogether and just use "ProtocolPreferred= ..." in the Server INI to select your choice. Details of these parameters are given below.

VERY IMPORTANT #1: The Firewall

Firewalls can be a problem. I have the default Windows firewall turned off, relying on the excellent one in my Router to protect the whole Network from the outside world. If you want to use firewalls between your PCs you will need to allow the FS exe (e.g. FS.EXE or FSX.EXE) and Wideclient.exe as exceptions, or at least sockets 8002 and 9002. I cannot advise you on how to make these exceptions—I would need to look up Windows firewall help the same as you will.

VERY IMPORTANT #2: The WorkGroup

The "WorkGroup" is a Windows concept used to divide up large networks into smaller groups and stop those groups all interacting with each other more than explicitly needed. In other words it is part of Microsoft's "efficiency" efforts. Unfortunately, though, PCs in different Workgroups do not receive Broadcasts from each other, and Broadcasts are used by WideFS in order to link Client to Server with no user hassle or worry.

This never mattered till Vista and Windows 7 came on the scene, as all Windows XP PCs by default had the same workgroup name and were therefore already in the same workgroup unless you changed it. But Microsoft changed the default Workgroup name with the introduction of Vista. So everyone with a mix of WinXP and Vista or Win7 PCs has, unless they fix things, at least two differently named Workgroups.

Grrr.

So, to summarise: in order to use WideFS with its default protocol (recommended) you should really not need to do anything other than Install as described above. BUT if you have a mix of XP, Vista and Win7 PCs, check and fix the workgroup names first. To do this:

On Windows XP: right-click on "My Computer" and select Properties, then the "Computer Name" tab. You will see your Computer Name and, below that, your Workgroup name. You can change both by pressing the "Change" button. I recommend you set both to something suiting you. My computer Names relate to their function, and of course you want a different name for each PC. My Workgroup is always "PETES". You want the same Workgroup name for all of your PCs.

On Windows Vista and Windows 7: click the 'Start' button (left-hand Windows button in the start bar) then right-click on "Computer" in the list of things on the right side of the pop-up. There select Properties. You'll see one of the sections deals with computer name and domain and workgroup settings. You can change the computer name and workgroup name by clicking the "Change settings" link. I recommend you set both to something suiting you. My computer Names relate to their function, and of course you want a different name for each PC. My Workgroup is always "PETES". You want the same Workgroup name for all of your PCs.

Okay. Once you have the same Workgroup names throughout, and either no firewall or exceptions properly made, everything should work ...

... However, if you find the connection is still not happening, or you have more than one Server and it is connecting to the wrong one, then you will need to add these lines to the [Config] section of the WideClient.ini file (add the [Config] section too if your ini file hasn't even got one!):

```
ServerName=NameOfServer  
Protocol=TCP
```

The protocol can be UDP instead, but that may be more problematic, so I'd recommend sticking to TCP, especially if you are finding you need to add these lines in the first place—this does seem to indicate that there are Network problems.

The ServerName is the computer name you gave to the PC on which you will be running Flight Simulator with WideServer. If you prefer you can use the Server's IP address instead of the name, thus:

```
ServerIPAddr=192.168.0.3
```

But I'd recommend only ever using names. The only time an explicit IP address works better is when the Broadcasts from the Server provide a bad IP address. This only seems to happen when the Network hub is a router which does odd things via "proxy" addresses. If you do specify an IP address you MUST set that specific IP address as the fixed one used in the Server PC --- you cannot then allow the Router to assign you one automatically via DHCP, as then it could change at each switch-on. More about that next (**Possible Gotchas**)

By the way, if both Name and Address are provided in the INI file, only the latter is used.

Possible Gotchas

If the IP Address for the server is not one of those treated by Windows as 'local', you may find WideClient attempting to connect via your Internet connection. I'm not really sure how all that works, but you should be safe if all your networked PCs have assigned IP addresses in the form "192.168.0.N" where N runs from 1 to 255. To assign IP addresses (at least in Windows 98—NT and so on may be different)—open up the Network Properties dialogue, find and highlight the line mentioning "TCP/IP" bound to your local Network adapter (e.g. "TCP/IP->3COM Ethernet Adapter"), then click the Properties button. In the IP Address tab, select "Specify an IP Address", then set the IP Address to 192.168.0.N (N numbering your PCs) and the subnet Mask to 255.255.255.0.

Incidentally, assigning specific IP addresses to your PCs in this way also helps reduce periodic but regular stutters in Flight Simulator, caused by the Network drivers querying the Network to get an address assigned (or something like that).

Please be aware of possible problems, especially when using TCP/IP, which may be introduced by Routers and Firewalls, whether hardware or software. Some firewalls (ZoneAlarm is one) will need you to give WideClient permission to connect to WideServer, and maybe vice versa. This may be needed each time you update the WideFS program, as it will be seen as a change and therefore a new potential danger. Do not think that closing the front-end of such programs solves this—it may make it worse. Best to officially tell the firewall that you are happy with the WideFS components. You may also find the web site www.helmig.com useful in solving Network problems, whether with IPX/SPX or TCP/IP.

Hint for Wireless Connections

I have received reports that the automatic wireless network settings for Windows XP can cause problems with timing. The service searches automatically for new wireless networks, and whilst doing this the current connection can be actually paused for as long as 1.5 seconds. These interruptions not only reduce performance but can also cause errors in WideFS, resulting in re-connections, affecting busy programs such as ActiveSky's weather updates quite severely.

There are some tools (for example **WirelessZeroShutdown**** from www.elligs.net) that successfully cut-off this service after the system boot. This particular one has been tried successfully. After using it, the Wideclient.log did not show any disconnects and the weather updates ran smoothly.

****WARNING:** This was contributed from a user:

This software (**WirelessZeroShutdown**) is able to cause major troubles with your Computer by removing access to the WLAN connection services "Windows Wireless Zero Configuration (wzcsvc)". The result is that you're unable to connect to any WLAN network, because your computer won't scan for any.

The culprit of the latest version of this application 1.23 is that it's unable to rollback the changes it made to the Windows system, so once you hit "Disable Windows Wireless Configuration (recommended)" [sic!] and want to revert to the original "Enable Windows Wireless Configuration" setting it is possible, *although the software tells you to make the changes with the next reboot* that wzcsvc will not load, which effects in you being unable to connect to any wireless lan whatsoever.

After some research I was able to find the solution to get my laptop connecting with the Internet again: Use regedit and go to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WZCSVC, add a DWORD entry named "Start" and value it "2" ("2" is automatic.) This can easy save you one hour or two of finding the error.

[Pete Dowson addition: you can also enable and disable services, and set their loading modes, in the Services section of your computer's "Manage" facility—right click on "My Computer" and select "Manage"].

What's this about UDP?

UDP is simpler and faster than TCP, and is installed on your PC as part of the TCP/IP package. It differs from TCP in that it is "connectionless"—data is sent to specific addresses (by IP address), but there are no checks on whether they get there and no error recovery to ensure they do. This is why it is faster, and with the block sequence and checking now in WideFS you should at least be able to detect if it is going wrong. With a WinXP or Vista server and a WinXP or Vista client, you can add "ProtocolPreferred=XXX" (XXX being TCP, SPX or UDP) to the [Config] section of the WideServer.INI file, and those clients not otherwise committed will use that protocol if possible. This gives you a quick way of changing the protocol on all computers in the Network, for performance comparisons. If you wish to select a specific protocol on any client, just add "Protocol=XXX" (XXX being TCP, SPX or UDP) to the [Config] section of the WideClient.INI file

Running WideFS

WideClient can be up and running even when there's no WideServer program elsewhere on your network to serve it. It will simply keep trying to connect until one is ready. Similarly it will re-connect if a server is closed and re-opened.

Likewise, WideServer is very tolerant of clients starting and stopping. Each server can serve many clients, but each client only talks to one server.

Simply run Flight Simulator on the server machine, and run WideClient (plus your utility program(s) on your other machines. That is all there is to it! Apart from some possibly displays in Flight Simulator's title bar there will be no other sign in Flight Simulator that WideFS is doing its job, and you should not notice any affect on frame rates at all.

In FS's Display settings you can set a limit on the Frame Rate which FS will keep to. I have found that WideFS operates much more smoothly and efficiently if you use this to prevent FS running away with the processor. With a reasonably fast machine try frame rate limits of 20–40, but reduce this on slower machines. For really fast machines and with no slow or overloaded client PCs, you could try running with the frame rate limiter set to unlimited. This may give rather uneven performance, but this may not be too noticeable, depending on the client PC capabilities.

Note that the number of current connections is shown in the title bar of Flight Simulator. This may fluctuate if responses to clients aren't currently being sent. This is because the client assumes that a lack of response for many tries means a disconnected link, so opens a new one. The server detects disused links after a while, and closes them, but there's an overlapping period whilst more connections may actually be open than there are clients.

Some programs—but very few now, Chris Brett's EFIS and John Hnidec's Moving Map are old examples—place their windows *within* the Simulator window, or now, in this case the WideClient window. But almost all modern utilities don't actually need WideClient visible at all—you can minimise it (or simply set an INI file parameter to do this as described in the Technical documentation).

Error reporting

Unless there is something really bad preventing WideClient from running, it will normally keep trying. Any errors it gets are described in text form in a Log file ("WideClient.log") which you will find in the same folder as the EXE file you are running. Please check there if you see WideClient still waiting for a connection when you believe it should be connecting.

Except for an incorrect or missing Registration, a serious error which prevents WideServer from offering a service can result in a message box appearing whilst Flight Simulator is getting ready. Pressing OK should allow you to continue using FS, but the WideFS link will not be operating. If you configure a "hot key" to restart WideServer (see **RestartHotKey** below, in the section on WideServer.ini options) then you can make WideServer try the whole initialisation sequence again. In FSX you can actually disable WideServer and re-enable it in the FSUIPC options. This forces a restart.

However, if the error is that serious you will probably have to close FS and sort out the network error being reported in the WideServer.log file (find this in the FS Modules folder).

If you have not yet registered WideFS with FSUIPC, the server will not start but will not have any adverse affect on FS operations. You will see a message in the WideServer.Log file reporting the registration problem.

Just about all of the errors that you are likely to see there will be due to network problems. I cannot give a precise list of messages with their meanings here—WideFS is merely reproducing the error number it gets from Windows software ("WinSock"), and the

text is also direct from Microsoft's own documentation. If you get stuck with any real errors, check through ALL the points listed in the trouble-shooting section in the Technical document, supplied separately.

If you write to me with any problems to look at (and I sincerely hope that this won't be needed, as everything I know is written down here somewhere), please attach a Zip file containing both the WideServer and WideClient LOG and INI files (or in the case of FSX and later the FSUIPC4.INI file instead of the WideServer.INI file).

FSUIPC Button Programming for Client Buttons

With a user-registered FSUIPC3 or 4 all Windows, EPIC and GoFlight buttons will be recognised on Client PCs and transmitted to FSUIPC for programming *unless* this action is explicitly turned off in the Client.

This action is automatic. You need do nothing to enable it—except, for GoFlight only, installing the Gfdev.dll as mentioned in the WideClient installation instructions earlier. For WideClient there is only one controlling parameter:

ButtonScanInterval=20

This parameter, in the [Config] section of the WideClient.ini file, controls the rate at which WideClient scans EPIC and Windows joystick buttons. GoFlight buttons are not scanned by WideClient—those are dealt with by the Gfdev.dll. However the parameter is still relevant since if you set this to 0 (zero), it switches the whole facility off.

The units here are in milliseconds. You can scan faster by reducing the value, but bear in mind that the frequency of button changes seen by the Server will be limited by the Network frame rate in any case, and this is pretty well tied to the FS frame rate by WideServer. The 20 millisecond default is generously allowing up to 50 changes per second in any case.

Buttons from different clients are differentiated by WideServer according to client names. When WideServer sees a new Client connecting, it makes a record of its name in the WideServer.ini file, assigning it a number. For example:

```
[ClientNames]
1=LEFT
2=CENTRE
```

Each time it reconnects it can then check and assign it the same number. This provides consistency in button numbering—the “joystick” number displayed by FSUIPC in its Buttons page will have 1000 added for Client 1, 2000 added for Client 2 and so on.

There is a limit of 32 client names. You aren't likely to reach that (I hope!), but if you change PCs or rename them, the list could build up with redundant names of clients no longer connected. WideServer has no way of knowing that the Client which used to be called “OLDPC” is effectively the same one as that called “NEWPC”. So when you change or rename PCs, please check the list here and edit the names so that they still reflect where things are correctly. You need to do that when FS is not running.

GPSout relay facility

WideClient will now receive GPS data from my GPSout module installed in the Server's FS (or, for FSX, from the GPSout option in a Registered FSUIPC4 installation). This data will simply be relayed to a local COM port. To do this you need set the serial port in GPSout to “WideFS”. Then, in the appropriate client's WideClient INI file add this section:

```
[GPSout]
Port=COMn
Speed=n
```

... with, of course, the correct Port and Speed for the local connection. For a USB-connection you might find you need the Port name to be something more like \\.\WCEUSBH001 (for example).

Note that, unless you are using Windows Vista on the Client PC, you can use the freeware Virtual Serial Port program “**MixW ComEmul**” (provided in the GPSout package for FS2004 or before, or downloadable from the Support Forum). This way you do not actually need any real COM ports to support a moving map program on the Client PC. In fact you can now use them on a Network with no actual serial cables involved. For Vista or to get over any other problems you find with MixW, you could try one or other of the proprietary solutions currently available. The ones I know about are Eltima Virtual Serial Port Driver (<http://www.virtualserialport.com>), which seems quite good but is not cheap, and the Eterlogic Virtual Serial Port Emulator, which is apparently free and which can be found at <http://www.eterlogic.com/Products.VSPE.html>

Note that you can have the same GPSout data being used on any of your client PCs. It is not restricted to one recipient.

For more details about GPSout please see the GPSout package or FSUIPC4 documentation.

Lua plug-in facilities

Since version 6.81, WideClient supports Lua plug-in programs in the same way as FSUIPC. The supported facilities are a subset of those in FSUIPC, but include most of the more useful features including:

- The complete **ipc** library with the exception of: **ask**, **buttons**, **display**, **getdisplay**, **getLvarId**, **getLvarName**, **keypressplus**, **linedisplay**, **readLvar**, **setdisplay**, **testbutton**, **testflag**, **writeLvar**.
- The complete **logic** library.
- The complete **com** library.
- The **event.com**, **event.timer**, **event.offset** and **event.cancel** functions from the **event** library.
- The complete **sound** library.

The **socket** support and all the other normal Lua facilities are supported identically to those in FSUIPC. The Lua documentation package is used for both FSUIPC and WideClient plug-ins, and indicates those facilities not included in WideClient quite clearly.

Unlike the FSUIPC facilities, WideClient automatically runs *all* Lua programs (files with filetype ".lua") which it finds in the same folder as its own WideClient.EXE program. There are no facilities to start and stop specific Lua programs. With one exception they are all started as soon as WideClient connects successfully to FSUIPC.

The exception is a Lua file named **initial.lua**. If such a file is found it is run almost immediately after WideClient starts up, irrespective of any connections. Naturally it cannot obtain any details from FSUIPC at this time, but it can do other local setting up operations in anticipation. It can be used to run other Lua programs—not by the **ipc.macro** function, which will want to run them on the FS PC, but using the newer **ipc.runlua** function. If this is used at all it should refer to Lua programs in other folders, such as a subfolder, not in the main **Wideclient.exe** folder because otherwise those will be restarted automatically when FSUIPC connection is successful.

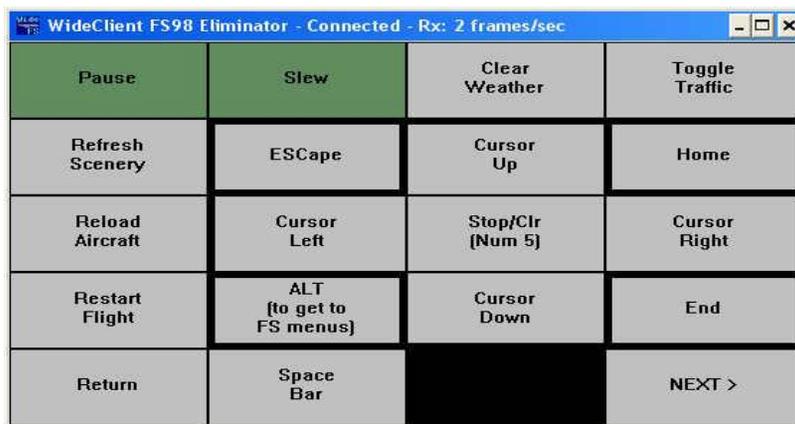
The facility is designed to run one-off set-up programs, or continuous loop or event-driven programs which run for the duration of the session. Of particular usefulness in this regard are Lua programs to provide additional **event.offset**-driven Sounds (via the **sound** library), and those to setup com-port connected devices, using the **com** library, possibly with a port sharing or pairing program like **VSPE**. Another interesting application would be to connect to the Internet via the **socket** facilities.

I'd be interested in feedback on uses of this facility. Contact me (Pete Dowson) on the Support Forum, please.

Button Screen Facilities

You can have the WideClient window itself turn into an array of “buttons”. The window becomes a regular matrix of rectangular buttons, by default as many as will fit in the particular size of window set. These are actually “virtual buttons” and operate the 288 virtual buttons provided in FSUIPC, numbered as Joystick 64, Button 0 up to Joystick 72, Button 31 (with 32 buttons on each of 9 imaginary “joysticks”). These buttons can, in turn, be programmed to do just about anything in FSUIPC’s Buttons page.

The main use of this will be for touch-sensitive screens in cockpits, for all those FS-related (and perhaps Instructor Station-related) things you need access to but which certainly aren’t part of a real cockpit. It beats having a mouse and keyboard around by miles! As a fairly simple example, look at the picture below, which is what appears on my Lilliput 7" touch sensitive screen (which, incidentally, is just right for mounting in an enclosed cockpit):



Pause	Slew	Clear Weather	Toggle Traffic
Refresh Scenery	ESCAPE	Cursor Up	Home
Reload Aircraft	Cursor Left	Stop/Clr (Num 5)	Cursor Right
Restart Flight	ALT (to get to FS menus)	Cursor Down	End
Return	Space Bar		NEXT >

A more extensive example, using all of the facilities about to be described, is provided in the Appendix at the end of this document.

To use this facility you need to place the parameter

ButtonScreen=Yes

into the [User] section of the WideClient.INI file, then add a completely new **[ButtonScreen]** section to the WideClient.ini file. Here's an example (this is for the screen above). Explanations are given below.

```
[ButtonScreen]
Size=4,5

0=T"Pause"
1=T"Slew"
2=B"Clear\rWeather"
3=B"Toggle\rTraffic"

4=B"Refresh\rScenery"
5=Ba"ESCAPE"
6=Bt"Cursor\rUp"
7=Ba"Home"

8=B"Reload\rAircraft"
9=Bl"Cursor\rLeft"
10=B"Stop/Clr\r(Num 5)"
11=Br"Cursor\rRight"

12=B"Restart\rFlight"
13=Ba"ALT\r(to get to\rFS menus)"
14=Bb"Cursor\rDown"
15=Ba"End"

16=B"Return"
17=B"Space\rBar"
```

Explanation:

The Size parameter is optional. It gives the desired button matrix size in terms of buttons wide by lines of buttons high. Here I've fixed it to 4 x 5 as you can see, as this suits my 7" display and fixing it like this allows me to develop it and test it, with that correct size, on another PC with a larger screen.

If you omit the size parameter then WideClient will fit in as many buttons of a finger-usable size as it can inside its Window. Just maximise the Window for the most on your screen.

The numbers 0= etc can run from 0 to 1023 max. Note that these are NOT the button numbers (they used to be, in earlier versions of WideClient). They are screen position reference numbers. If the screen is 4 x 5, then references 0-19 are on the first page, 20-39 are on the second, and so on. The numbering runs left to right from the top left corner to the bottom right corner, just like rows of text.

Button numbers are assigned in reference number order, from 0 (joy# 64, btn# 0) to 287 (joy# 72, btn# 31), but some parameter types will reserve a space on the display for management functions such as selecting the page to go to, or even declaring an unused position. These do not get button numbers allocated. All this will now hopefully be made clear.

In the part after the = the first character should be one of

First, an *optional* background colour selector digit, 0-9. Note that if you override Toggle button background colours this way, they will not switch to show the toggled state. Colour settings are described below.

Then/or one of these:

- B is for Button: this declares an ordinary button, i.e. push on, release off. These go Blue when you touch them.
- T is for Toggle: push once, it's on, push again, it's off. Dull green when off, bright Red when on.

- TN is the same as T, for Toggle, except that WideClient assumes the toggle condition is on initially, rather than off.
- P is for "Previous", meaning go to the previous page, if any
- N is for "Next", meaning go to the Next page if any
- G is for "Go to", and is followed by a line reference number. The page displaying that slot is the target.
- U is for "Unused". It will just be black and unresponsive. The place where the default "< PREV" button would have gone in the first page shown above is effectively an automatic "U" entry. Note that no virtual button number is reserved for an Unused position. If you merely want to reserve a place for use later, give it a blank label and colour it as you wish, even black so it is indistinguishable from "Unused" (except that it will still send its button number to FSUIPC). Blank labels can be obtained by using spaces.

Then there are optional letters for border highlighting (thickening), so you can divide up the matrix on screen or highlight special areas. See the picture above where you will see its usefulness. The letters are:

- a all 4 borders
- t top border
- l left border
- r right border
- b bottom border

You can combine them, for instance ltr (in any order).

Then there's an optional Label for the button, in quotes (""). You can have up to 3 lines (with \r between them). The program doesn't automatically wrap them, you have to fit the text in yourself.

If there's no label the default label for a "T" or "B" button definition is "Button N\rJ,B" where N is the virtual button/line number (0-287) and J,B are the Joy,Btn numbers as seen by FSUIPC. For "G" it will be "Go To" and the reference, and for "P" and "N" it will be the default "< PREV" and "NEXT >". Naturally "U" (Unused) has no label.

If, on any particular page, you do not use the P, N or G facilities to define your own method of moving to different pages, the program plants the PREV and NEXT buttons as the last two on that page, and moves whatever you had there to the next. This means that the default presentation, with no use of the newer P, N and G facilities, will still look as it does in the example provided.

If you resize the window the buttons are resized and rearranged accordingly. The definitions in the INI file are re-read as well when it is resized, so you can take advantage of that when making changes to avoid having to close and reload WideClient. For actual operation, though, it is best either to use the Size parameter, or to keep it maximised (set Visible=Max in the [Config] section, for example).

When WideClient is not connected, the buttons will all be blank and inoperative. They may actually flash if you are losing and regaining connections to WideServer.

Page titles:

You can have an extra line, across the whole button screen, giving a page title. This uses the same font, but a different background colour (colour 8, a blue unless changed as mentioned in the next section).

Titles are of up to 64 characters, specified in lines like this:

```
Title1="This is the title for Page 1"
```

Lines can be provided for pages up to 99. Even if only one page title is provided, the title space is reserved on every page so that the button rectangles don't vary in size.

Colour settings:

There are only 5 default colours. These are numbered 0 to 3 (by default 4-7 act like 0-3), and 7:

- 0 Light gray, for ordinary buttons which have been labelled
- 1 A light blue, not currently used automatically
- 2 The light green used for Toggle buttons when not set
- 3 The pink used for Toggle buttons when set
- 7 Light blue used for page titles

These can be changed, and you can add 5 more colours numbered 4 to 6 and 8 and 9 if you wish. Colours can be forced for individual button spaces (including replacing the darker gray of Prev, Next and GoTo buttons) by prefixing the parameter with the colour number, 0–9 as already described.

Specify colours as follows:

ColourN= r, g, b

Where N is the number, 0–9, and r, g and b are the Red, Green and Blue components as a number from 0 (none) to 255 (full).

In deference to my American users, the spelling “Color” is also accepted. ;-)

For example: Colour3=255,0,0

Will produce pure red at its maximum, and this will replace the Toggle set indication pink.

Note that if WideClient judges the background to be on the dark side, it will automatically make the text colour white instead of black.

Font size:

The font size can be set, too, within reason. You'd need to take care with this though:

FontSize=X,Y

where X is the width and Y is the height, as relative values based on 10x point sizes. The default font is approximately 240,240, so judge changes relative to this.

The rest of the Example:

To complete the above example, here's the FSUIPC [Buttons] section entries corresponding to the 18 virtual buttons programmed:

```
609=P64,0,C65794,0
610=U64,0,C65795,0
611=P64,1,C65731,0
612=U64,1,C65730,0
613=P64,2,K87,11
614=P64,3,C1009,0
615=P64,4,C65562,0
616=P64,5,K27,8
617=P64,6,K38,8
618=P64,7,K36,8
619=P64,8,C66512,0
620=P64,9,K37,8
621=P64,10,K12,8
622=P64,11,K39,8
623=P64,12,C65591,0
624=P64,13,K0,24
625=P64,14,K40,8
626=P64,15,K35,8
627=P64,16,K13,8
628=P64,17,K32,8
```

Note that, in order to get all the desired results some editing in the FSUIPC.INI file was necessary. Some of the keys enabled actually allow FS's Menus to be used. Pressing the ALT key on its own starts the process. To get this programmed in FSUIPC you need to tell it to produce a normal combination which it will accept, eg. Shift+Alt+A. This will give you an entry K65,25 in the file. Change that to K0,24 and you will get just the ALT. The “0” is a “null” character not featured on the keyboard. You will see this result in line 624 above.

The other one you may need to edit is the ESCape key. This is K27 as seen in line 616.

Incidentally, the Clear Weather key encoding is Shift+Ctrl+W, which is how it is programmed in FSUIPC's “Hot Keys” page. Buttons programmed to produce keystrokes will be detected as hot keys by FSUIPC or any application using FSUIPC's programmable hot keys facilities.

APPENDIX: Extensive ButtonScreen Example

This is a working example, kindly provided by Dennis Hardy. Thanks, Dennis!

Only the ButtonScreen definition is shown here, with pictures of a sample two of the 7 pages it creates. To see the other pages just insert the complete [ButtonScreen] section into your WideClient.INI file, remembering also to put "ButtonScreen=Yes" into the [User] section, then run your WideFS system. You will have to then drag the WideClient window to a large enough size to accommodate the 13 x 5 matrix of buttons used, otherwise the default sizes will prevail and it will look completely wrong.

The way the buttons are assigned is not shown, and in some ways it is specific to the particular installation, but most can be seen to be normal FS controls.

Here's how two of the pages (1 and 3) look, when sized appropriately (actually, they really look a lot better—the PDF format of this document doesn't show the fonts so well!):

SYSTEMS MENU												
SYSTEMS	Battery Master	Alternator Master	Avionics Master	Pitot Heat	Fuel Pump	Engine 1 Auto start	Engine 2 Auto start	Engine 3 Auto start	Engine 4 Auto start	Engine Auto Shutdown	Parking Brake	GPS
AUTO PILOT	Magneto OFF	Magneto Left	Magneto Right	Magneto Both	GA Engine Starter	APU OFF	APU ON	APU Fire Extinguisher		Cowl Flaps Open	Gear Up	Misc
NAV COM	Panel Lights	Beacon	Landing Light	Nav Lights	Strobe Light	Anti-ice OFF	Anti-ice ON	No Smoking		Cowl Flaps Close	Gear Down	
G1000 PFD	Standard Barometer Setting	Barometer +	Barometer -	Prop Sync	Water Rudder	Prop De-ice		Seat Belts		Release Tow Line	Request Tow Plane	
G1000 MFD	Aileron trim Left	Aileron trim Right	Rudder trim Left	Rudder trim Right		Door	Jetway	Pushback	P/B Left	P/B Right	Call Fuel Truck	

NAVCOM MENU												
SYSTEMS	Com 1 Int +	Com 1 Dec +	Com 2 Int +	Com 2 Dec +	Com 1 Swap	ADF 100s +	ADF 10s +	ADF 1s +	ADF 0.1s +	DME 1	DME 2	GPS
AUTO PILOT	Com 1 Int -	Com 1 Dec -	Com 2 Int -	Com 2 Dec -	Com 2 Swap	ADF 100s -	ADF 10s -	ADF 1s -	ADF 0.1s -			Misc
NAV COM	Nav 1 Int +	Nav 1 Dec +	Nav 2 Int +	Nav 2 Dec +	Nav 1 Swap	XPDR 1000s +	XPDR 100s +	XPDR 10s +	XPDR 1s +			
G1000 PFD	Nav 1 Int -	Nav 1 Dec -	Nav 2 Int -	Nav 2 Dec -	Nav 2 Swap	XPDR 1000s -	XPDR 100s -	XPDR 10s -	XPDR 1s -			
G1000 MFD	Com 1	Com 2	Both	Nav 1 Ident	Nav 2 Ident	MKR	DME	ADF	NAV/GPS			

Here's the complete section (please read columns left then right on each page to retain correct sequence):

```
[ButtonScreen]
Size=13,5

Colour4=221,250,182
Colour5=235,250,160
Colour6=238,183,172
Colour8=180,44,2
Colour7=26,122,144
Colour9=231,155,17

Title1="SYSTEMS MENU"
Title2="AUTOPILOT MENU"
Title3="NAVCOM MENU"
Title4="G1000 PRIMARY FLIGHT DISPLAY"
Title5="G1000 MULTI-FUNCTION DISPLAY"
Title6="GPS"
Title7="MISC"

=====PAGE 1 SYSTEMS=====
0=7G0r"SYSTEMS"
1=T1"Battery\rMaster"
2=T"Alternator\rMaster"
3=T"Avionics\rMaster"
4=T"Pitot\rHeat"
5=T"Fuel\rPump"
6=B1"Engine 1\rAuto start"
7=B"Engine 2\rAuto start"
8=B"Engine 3\rAuto start"
9=B"Engine 4\rAuto start"
10=B"Engine\rAuto\rShutdown"
11=Tr1"Parking\rBrake"
12=4G3251"GPS"

13=4G65r"AUTO\rPILOT"
14=6B1t"Magneto\rOFF"
15=6Bt"Magneto\rLeft"
16=6Bt"Magneto\rRight"
17=6Bt"Magneto\rBoth"
18=6Bt"GA Engine\rStarter"
19=5Bt1"APU\rOFF"
20=5Bt"APU\rON"
21=5Btr"APU Fire\rExtinguisher"
22=Bt " "
23=Bt1"Cowl Flaps\rOpen"
24=Bt1r"Gear\rUp"
25=4G3901"Misc"

26=4G130r"NAV\rCOM"
27=Ttbl"Panel\rLights"
28=Ttb"Beacon"
29=Ttb"Landing\rLight"
30=Ttb"Nav\rLights"
31=Ttb"Strobe\rLight"
32=Bt1"Anti-ice\rOFF"
33=Btr"Anti-ice\rON" "
34=Ttr"No\rSmoking"
35=B " "
36=Blb"Cowl Flaps\rClose"
37=Blbr"Gear\rDown"
38=U

39=4G195r"G1000\rPFD"
40=7B1"Standard\r Barometer\rSetting"
41=6B"Barometer\r+"
42=6Br"Barometer\r-"
43=T"Prop\rSync"
44=T"Water\rRudder"
45=Bb1"Prop\rDe-ice"
46=Brb " "
47=Trb"Seat\rBelts"
48=B " "
49=B"Release\rTow\rLine"
50=B"Request\rTow\rPlane"
51=U

52=4G260r"G1000\rMFD"
53=6B1t"Aileron trim\rLeft"
54=6Bt"Aileron trim\rRight"
55=6Bt"Rudder trim\rLeft"
56=6Btr"Rudder trim\rRight"
57=B " "
58=T1"Door"
59=T"Jetway"
60=T"Pushback"
61=2B"P/B\rLeft"

62=2B"P/B\rRight"
63=B"Call\rFuel\rTruck"
64=U

=====PAGE 2 AUTOPILOT=====
65=4G0r"SYSTEMS"
66=T1"A/P\rMaster"
67=T"Heading\rHold"
68=T"Nav\rHold"
69=T"Apr"
70=T"Back\rCourse"
71=T"Alt\rHold"
72=T"A/T"
73=T"Speed\rHold"
74=T"Yaw\rDamper"
75=T"Flight\rDirector"
76=Tr"Loc\rHold"
77=4G3251"GPS"

78=7G65r"AUTO\rPILOT"
79=B1"Nav/GPS"
80=5B"Heading\rBug +"
81=5B"Obs1\r +"
82=5B"Obs2\r +"
83=5B"ADF\r+"
84=5B"Alt\r+"
85=5B"TOGA "
86=5B"Speed\r+"
87=5B"Vert Speed\r+"
88=5B"Spoilers\rArm "
89=5Br"Autobrake\rIncrease "
90=4G3901"Misc"

91=4G130r"NAV\rCOM"
92=Tb1 " "
93=5Bb"Heading\rBug -"
94=5Bb"Obs1\r-"
95=5Bb"Obs2\r-"
96=5Bb"ADF\r-"
97=5Bb"Alt\r-"
98=5Bb " "
99=5Bb"Speed\r-"
100=5Bb"Vert Speed\r-"
101=5Bb"Spoilers\rOFF"
102=5Bb"Autobrake\rDecrease"
103=U

104=4G195r"G1000\rPFD"
105=B1 " "
106=B " "
107=B " "
108=B " "
109=B " "
110=B " "
111=B " "
112=B " "
113=B " "
114=B " "
115=B " "
116=U

117=4G260r"G1000\rMFD"
118=B1 " "
119=B " "
120=B " "
121=B " "
122=B " "
123=B " "
124=B " "
125=B " "
126=B " "
127=B " "
128=B " "
129=U

=====PAGE 3 NAVCOM=====
130=4G0r"SYSTEMS"
131=5B1"Com 1\r Int\r+"
132=5Br"Com 1\rDec\r+"
133=5B"Com 2\r Int\r+"
134=5Br"Com 2\rDec\r+"
135=7Br"Com 1\rSwap"
136=6B1"ADF\r100s\r+"
137=6B"ADF\r10s\r+"
138=6B"ADF\r1s\r+"
139=6B"ADF\r0.1s\r+"

```

140=7B1"DME 1"
141=7Br"DME 2"
142=4G3251"GPS"

143=4G65r"AUTO\rPILOT"
144=5Bb1"Com 1\rInt\r-"
145=5Brb"Com 1\rDec\r-"
146=5Bb"Com 2\rInt\r-"
147=5Bbr"Com 2\rDec\r-"
148=7Bbr"Com 2\rSwap"
149=6B1"ADF\r100s\r-"
150=6B"ADF\r10s\r-"
151=6B"ADF\r1s\r-"
152=6B"ADF\r0.1s\r-"
153=B1" "
154=Br" "
155=4G3901"Misc"

156=7G130r"NAV\rCOM"
157=1B1"Nav 1\r Int\r+ "
158=1Br"Nav 1\rDec\r+ "
159=1B"Nav 2\r Int\r+ "
160=1Br"Nav 2\rDec\r+ "
161=7Br"Nav 1\rSwap"
162=6B1"XPDR\r1000s\r+ "
163=6B"XPDR\r100s\r+ "
164=6B"XPDR\r10s\r+ "
165=6B"XPDR\r1s\r+ "
166=B1" "
167=B" "
168=U

169=4G195r"G1000\rPFD"
170=1Bb1"Nav 1\r Int\r- "
171=1Brb"Nav 1\rDec\r- "
172=1Bb"Nav 2\r Int\r- "
173=1Brb"Nav 2\rDec\r- "
174=7Bbr"Nav 2\rSwap"
175=6Bb1"XPDR\r1000s\r- "
176=6Bb"XPDR\r100s\r- "
177=6Bb"XPDR\r10s\r- "
178=6Bb"XPDR\r1s\r- "
179=B1" "
180=B" "
181=U

182=4G260r"G1000\rMFD"
183=2B1"Com 1"
184=2B"Com 2"
185=T"Both"
186=T"Nav 1\rIdent"
187=T"Nav 2\rIdent"
188=T"MKR"
189=T"DME"
190=T"ADF"
191=2B"NAV/GPS"
192=B1" "
193=B" "
194=U

=====PAGE 4 G1000 PFD=====
195=4G0r"SYSTEMS"
196=B1"Group Knob\rIncrease "
197=B"Group Knob\rDecrease "
198=U
199=U
200=U
201=U
202=U
203=U
204=U
205=B"Range\rZoom Out"
206=Br"Range\rZoom In"
207=4G3251"GPS"

208=4G65r"AUTO\rPILOT"
209=B1"Page Knob\rIncrease "
210=B"Page Knob\rDecrease "
211=U
212=U
213=U
214=U
215=U
216=U
217=U
218=B"Direct"
219=Br"Menu"
220=4G3901"Misc"

221=4G130r"NAV\rCOM"
222=B1"Cursor"
223=B" "

224=U
225=U
226=U
227=U
228=U
229=U
230=U
231=B"FPL"
232=B"PROC"
233=U

234=7G195r"G1000\rPFD"
235=6Btrl"Soft Key\r1"
236=Bb" "
237=U
238=U
239=U
240=U
241=U
242=U
243=U
244=Bb"CLR"
245=Bb"Enter"
246=U

247=4G260r"G1000\rMFD"
248=6B1"Soft Key\r2"
249=6B"Soft Key\r3"
250=6B"Soft Key\r4"
251=6B"Soft Key\r5"
252=6B"Soft Key\r6"
253=6B"Soft Key\r7"
254=6B"Soft Key\r8"
255=6B"Soft Key\r9"
256=6B"Soft Key\r10"
257=6B"Soft Key\r11"
258=6B"Soft Key\r12"
259=U

=====PAGE 5 G1000 MFD=====
260=4G0"SYSTEMS"
261=B1"Group Knob\rIncrease "
262=B"Group Knob\rDecrease "
263=U
264=U
265=U
266=U
267=U
268=U
269=U
270=B"Range\rZoom Out"
271=Br"Range\rZoom In"
272=4G3281"GPS"

273=4G65"AUTO\rPILOT"
274=B1"Page Knob\rIncrease "
275=B"Page Knob\rDecrease "
276=U
277=U
278=U
279=U
280=U
281=U
282=U
283=B"Direct"
284=Br"Menu"
285=4G3941"Misc"

286=4G130"NAV\rCOM"
287=B1"Cursor"
288=B" "
289=U
290=U
291=U
292=U
293=U
294=U
295=U
296=B"FPL"
297=B"PROC"
298=U

299=4G195"G1000\rPFD"
300=1Btl"Soft Key\r1"
301=U
302=U
303=U
304=U
305=U
306=U
307=U
308=U

309=Bb"CLR"
310=Bb"Enter"
311=U

312=7G260"G1000\rMFD"
313=1B1"Soft Key\r2"
314=1B"Soft Key\r3"
315=1B"Soft Key\r4"
316=1B"Soft Key\r5"
317=1B"Soft Key\r6"
318=1B"Soft Key\r7"
319=1B"Soft Key\r8"
320=1B"Soft Key\r9"
321=1B"Soft Key\r10"
322=1B"Soft Key\r11"
323=1B"Soft Key\r12"
324=U

=====PAGE 6 GPS =====

325=4G0r"SYSTEMS"
326=B1"Group\rKnob\r-"
327=B"Group\rKnob\r+ "
328=U
329=U
330=U
331=U
332=U
333=U
334=U
335=B"Zoom\rIn"
336=Br"Zoom\rOut"
337=7G3251"GPS"

338=4G65r"AUTO\rPILOT"
339=B1"Page\rKnob\r- "
340=B"Page\rKnob\r+ "
341=U
342=U
343=U
344=U
345=U
346=U
347=U
348=B" "
349=Br"Direct\rTo "
350=4G3901"Misc"

351=4G130r"NAV\rCOM"
352=B1" "
353=B" "
354=U
355=U
356=U
357=U
358=U
359=U
360=U
361=B" "
362=Br"Menu"
363=U

364=4G195r"G1000\rPFD"
365=B1" "
366=B" "
367=U
368=U
369=U
370=U

371=U
372=U
373=U
374=B"Cursor"
375=Br"Clear"
376=U

377=4G260r"G1000\rMFD"
378=B1"PWR"
379=B" "
380=B" "
381=B" "
382=B"NRST"
383=B"OBS"
384=B"MSG"
385=B"FPL"
386=B"TERR"
387=B"PROC"
388=Br"Enter"
389=U

=====PAGE 7 MISC =====

390=4G0r"SYSTEMS"
391=B1" "
392=B" "
393=B" "
394=B" "
395=B" "
396=B" "
397=B" "
398=B" "
399=B" "
400=B" "
401=Br" "
402=4G3251"GPS"

403=4G65r"AUTO\rPILOT"
404=B1" "
405=B" "
406=B" "
407=B" "
408=B" "
409=B" "
410=B" "
411=B" "
412=B" "
413=B" "
414=Br" "
415=7G3901"Misc"

416=4G130r"NAV\rCOM"
417=B14G260"G1000\rMFD" "
418=B" "
419=B" "
420=B" "
421=B" "
422=B" "
423=B" "
424=B" "
425=B" "
426=B" "
427=Br" "
428=U

429=4G195b"G1000\rPFD"
430=4G260trb"G1000\rMFD"
431=B" "
432=B" "