



INTRODUCTION TO FILEPROWEB

SUMMARY

The basic guide to configuring and setting up fileProGI and fileProWeb to utilize its interfaces and themes.

Christopher Henschen

Henschen and Associates, Inc

Copyrighted by Henschen and Associates, Inc

(All rights reserved 2019)

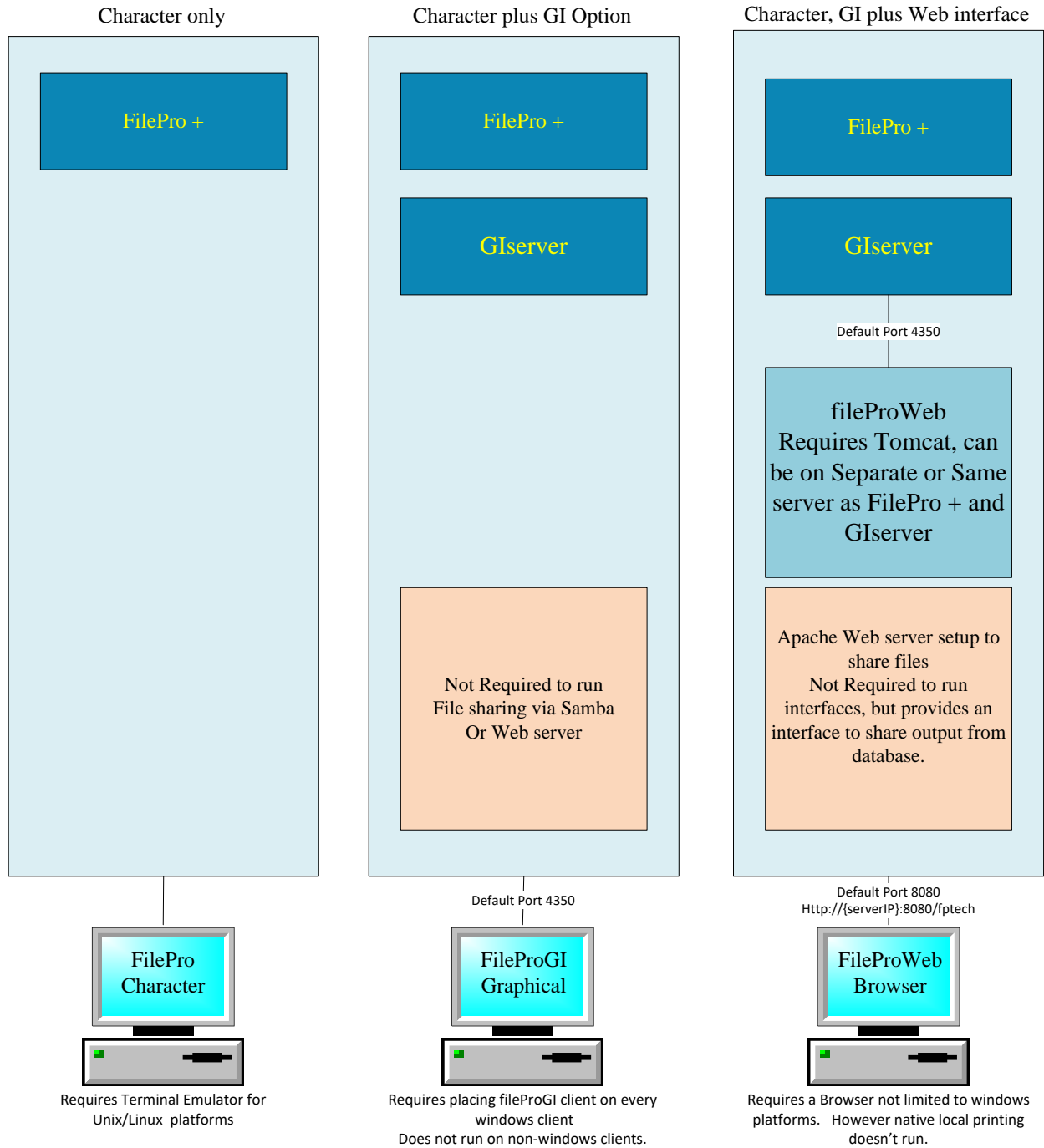
Contents

How's it all work?	5
Install fileProWeb.....	6
Separate WebServer Changes or port GIservr isn't default 4350:.....	8
Setup Webserver	10
~/fp/giserver/lib/password.cfg.....	11
~/fp/giserver/lib/giserver.cfg.....	11
~/fp/giserver/user/default.cfg {login account}.cfg	14
GIservr V6 – Account management.....	14
popPDF – popup PDF in GIclient.....	16
After Installation, Customize some of the features:.....	18
Dropdowns	20
fileProWeb themes.....	22
How to edit/create themes:	23
Modification creating specific themes:.....	24
Notification windows:.....	27
Show and Edit Buttons in fileProGI/fileProWeb.....	29
Buttons using Virtual Keys:	33
Top Button Icon Association in Web.....	33
Processing Changes.....	36
Message Send Command:.....	36
@GUI.SEND – allows the passing of the message send format flags without the need of setting up a handle, etc. and receive the results.....	37
@P- Sets Global pathname for client dropdown locations (Web and GIclient)	37
@LDD- Sets alternative drop down list definition file. Where alt_droplist is the name of the alternative dropdown list definitions. (GIclient with toolkit be 6.x).....	38
@TBPATH - Specify path to get theme buttons graphic representation (Web only)	38
@D Display - BUILT-IN VIEWING (GIclient only)	38
@X - Takes down Display (GIclient only used with D)	40
@E Embed - EMBEDDED FILES/GRAPHICS (Web up to 10 [0-10] embedded, GIclients single embed)	40
@EZ – Embedded files – Place in an iframe (Web Only)	41
@EF – Embedded files – Place in and iframe (Web Only)	42
@EFZ – Embedded files – Place in an iframe above filePro's z-order (Web Only)	42

@EX - Takes down the Embedded displays (Web used with @EF)	43
@EXZ - Takes down the Embedded displays (Web used with @EFZ).....	43
@R – Run program based on default extension (GIClient only)	43
@RH – Run program based on default extension in background (GIClient only)	44
@NW – New Window popup (Web only)	44
@SETNORTHHEIGHT--- Sets north region height in pixels, 0 means no region shown (Web only) ...	45
@SETNORTHURL -- Sets the URL content for the north region (Web only)	45
@SETEASTWIDTH-- Sets east region width in pixels, 0 means no region shown (Web only)	45
@SETEASTURL-- Sets the URL content for the east region (Web only)	45
@CollapseWest - Collapse of the west region in a data entry screen. (Web only)	46
@ExpandWest - Expand the west region in a data entry screen. (Web only).....	46
@EMAILFILES - Dumps error log and attaches to email for debugging purposes (GI Client only)	46
@LOGF - Dumps a snapshot log of the GIClient to a specified location (GI Client only)	46
@PBS/@PBH – Progress Bar Status/Hide (GI Client only)	46
@GUI.DLG- Display a Dialog Box (GIClient).....	47
@GUI.IP – Returns the IP address of Client. (web and GI)	48
@GUI.PCName – Returns the PCName of Client. (GI only, Web will report unknown).....	48
@GUI.URL - Web only returns the full URL for the session. Can be used to parse out the theme ...	48
@GUI.PAUSE()/@GUI.RESUME - Pause/Resume output to client. (Glients and web)	49
@GUI.APPDIR - reports back the directory location the Glients is running from. (Glients only).	49
@GUI.WINDOWSDIR - reports back the windows directory location. (Glients only)	49
@GUI.FW - Find Window reports existence of window with the specified caption. (Glients only)	49
@GUI.SFN - Select File Name on the Client PC returns the full path of the selection (Glients only)	49
@GUI.DELETEFILE - Deletes full pathed files on the Client PC (Glients only)	50
@GUI.FE - Check for existence of file on the Client PC (Glients only).....	50
@GUI.DE - Check for existence of a directory on the Client PC (Glients only)	50
@GUI.CREATEDIR - Create Directly full pathed on the Client PC (Glients only)	50
@GUI.ZIP - zips file(s) on GIClient to full path zip location	50
@GUI.UNZIP - unzips file(s) on GIClient to full path zip location	51
@SETCLIPBOARD- Places text in Web client’s clipboard for local paste. (Web only).....	51
Building HTML Tools	52
atKey:{key} (Use in your own html files with fileProWeb only).....	52

@wvf with sendVF:{content}, '*' - Pass data string to @VF (fileProWeb only format).....	54
Sendfieldleavemsg= (GIclient only format).....	55
Data Tunneling:	56
Processing changes and additions from GI version	59
Comparison Chart	59
Setting up a GIclient executable (not Web related)	62
Application Configuration Table – DEVELOPER (Toolkit).....	63
Dropdown Lists	66
Execution with title format:	67
MISCELLANEOUS CHANGES	68
Changing colors in GIclient (Not Web Related)	70
From Experience	75

Adding to the FilePro Product Line



For the past several years **FilePro** has been a rapid, cross-Operating System, programmer's development tool. Growing from its roots of a character-based interface, the **fileProGI** product line provides a graphical interface with mouse clicks, and the ability to launch other applications using their default extensions. Building on fileProGI comes the **fileProWeb** application. With little-to-no changes to existing filePro code, your programs can be shown in a browser. Making adjustments to your existing code can open up a whole new world of possibilities.

How's it all work?

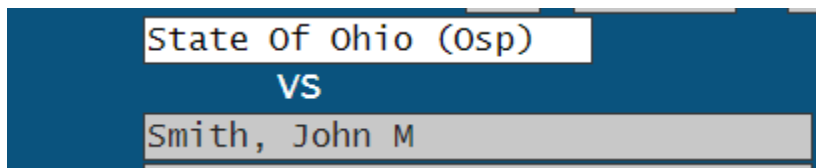
Using the diagram above let's take a look at how the fPTechnologies of Ohio, Inc. product line works with the new fileProWeb product. First, we start off with the database server. This server could be any of the supported FilePro Plus OS platforms, including Linux, XinuOS (SCO), BSD, or Windows. On its own, only a character version of FilePro can be accessed through a terminal emulator on PCs (Unix/Linux type platforms), or using a .bat script for Windows platforms. Adding to the base product, the graphical fileProGI product line was created; it requires the GIserv module be installed on the server with the database, and an associated GIclient installed on each Windows workstation. (Windows-only client at present.) The GIserv handles the messaging between the server's database and the GIclients, and keeps track of sessions status. Building onto this structure is the current fileProWeb. (Note: GIserv and filePro must be the current 6.0.1 version or above for fileProWeb to run.) The fileProWeb infrastructure can run on the same server as the database server, or on its own separate server communicating request through a port. It requires the use of tomcat and Java 1.8 or above on the web server, allowing it to be more flexible, supporting different server platforms. (At this time OpenServer 6/UW7 do not support java 1.8 or higher; for these platforms a separate OS is required to complete the solution.) With fileProWeb, Clients are no longer limited to Windows PC, but support web/network browsers, presented with a login prompt, to establish connections and make requests. (For best results, we recommend Chrome; other browsers should work, but were not tested as much. If you find yourself getting dropped while moving between sessions this may be due to a lack of memory available on the client device. A possible solution, short of getting more memory, if you are using Chrome, is to Open a new tab. In the search box type "chrome://flags/#automatic-tab-discarding"; change it from default to disabled.) Though not required it might make sense to also setup an Apache server to use the vast array of html 5 code available in some environments. This may help to launch output, such as PDF documents, charts, etc. if fileProWeb isn't installed on the same server as database.

Install fileProWeb

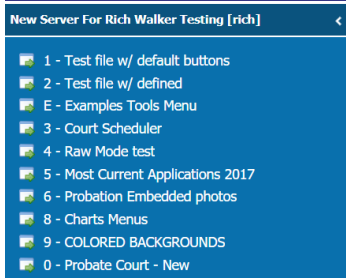
The fileProWeb part of the installation has a couple of options. It can be installed on the same server as the filePro database (preferred method) and GIservr, or on a separate server. (Note: As of this writing, if you are running on one of the XinuOS platforms, OpenServer or UnixWare, you have to use a separate platform for fileProWeb. This is because fileProWeb requires Java 1.8 or above with use of Tomcat and these platforms do not have current support.) First, be sure the filePro and GIservr are running correctly as mentioned above. Download the fileProWeb installer on your server that supports java 1.8 or above. There is no license to download as the GIservr license contains the information about the number of sessions, features, and other information needed. Once installed the webserver will run as a service starting up at boot. For Windows platforms, there's a few simple questions regarding installation locations and it's off and running.

For BSD/*nix platforms, there are additional setup options for you; these require manually changes if installing on Windows. For most, the default options will work for a server setup. Since the BSD/Linux platforms have more options, I'll use this for the walk-through. Both BSD/Linux platforms use the same installer; simply make sure you login as root and run the file ./fileProWeb-6.x.x...installer.run or variance of the command.

- 1) The installer first checks for the required Java 1.8 is installed and will ask if you wish to install if no java's installed. If an older version it will show a notice before it terminates.
- 2) If there's not been a filePro app loaded, it will prompt for install locations defaulting to /appl
- 3) If a filePro app is already installed, it'll ask if this is an upgrade or fresh copy. "Fresh" moves existing folder as a backup and lays down a clean copy. Any previous changes will have to be copied back. Updates will update core parts, leaving the root location alone as this is the workspace for developer's changes.
- 4) Some assumptions are made. There's a new /appl/fpoffice folder created which holds the text dropdowns, PDF, images, and font temp location on the BSD/*nix platforms; this is linked to the document home folder for fileProWeb.
- 5) "Protected 3D-look fields" (Y/N) grays out the protected data fields, but makes them easier to read, separated from the normal page text. The default is (Y) for this look. Ex: the Smith, John M in 3D. Otherwise, select 'N' for the field text to show up like the "VS"; not grayed out.



- 6) Next, see the associated action keys. This references the keys associated to menu items. (Default N)



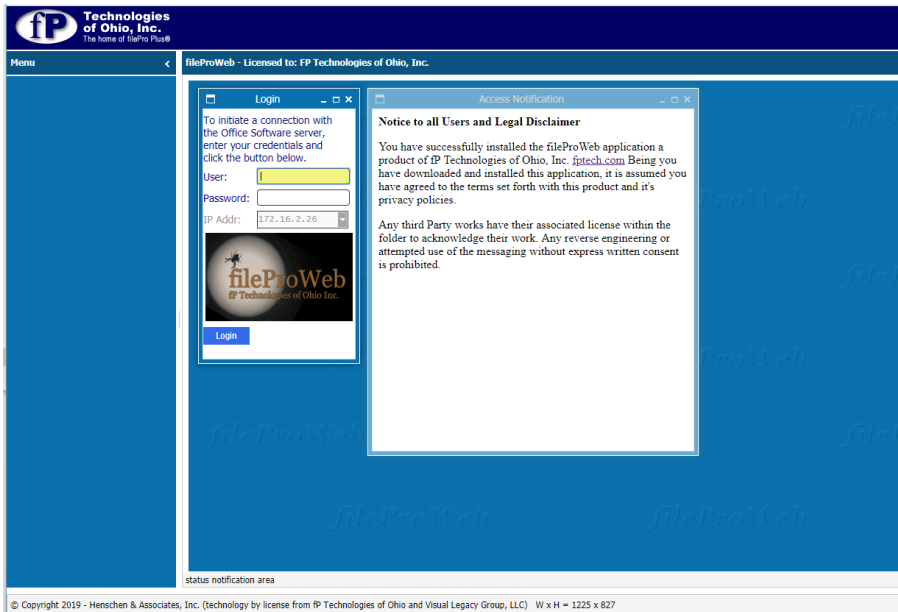
- 7) Lastly, if you wish to setup a "PDF printer" to pop things up in the new tab, answer "Yes" and a "popPDF" printer is setup in filePro's printer maintenance table.

Provided the fileProWeb service was installed on the same server as the filePro and fileProGI, it should be running and accessible via the URL : (default portno is 8080) (If GIservr is on a separate server or port then default skip to section **Separate WebServer Changes first and return here**)

<http://{this servers IP address}:{portno}/fptech>

ex default (**where localhost is the domain/IP of the server fileProWeb is running on**):

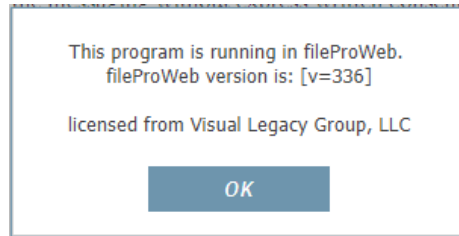
<http://localhost:8080/fptech>



<http://localhost:8080/fptech/>

Note: in the event you make changes to configuration files after installation, or change associated images, it may require emptying the cache before the changes are seen.

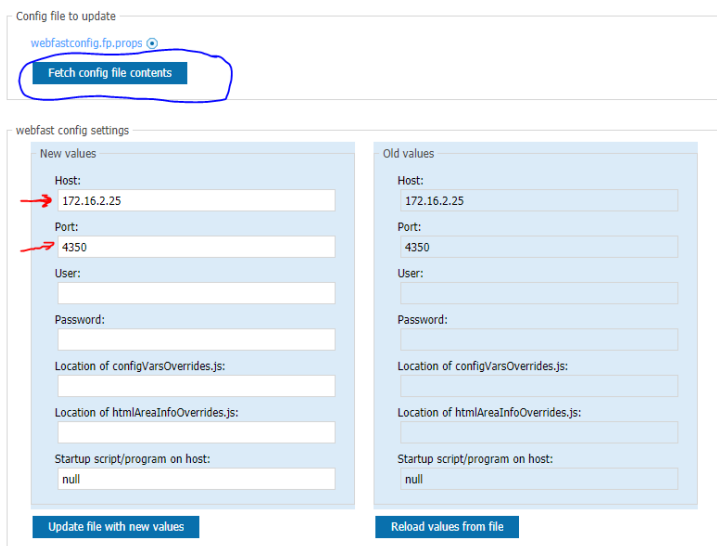
Also note, once logged in, if you ever need the version you are running, a simple keyboard combination of '<CTRL>=' will display the fileProWeb version from any screen. This version is separate from the GIservr version used to interface with fileProGI and fileProWeb.



If your browser asks to save your password at fileProWeb prompts it is recommended not to do so. Doing so might cause data to appear in other fields by the nature of its design. In Chrome you can disable saving passwords: Open chrome->Click the 3 dot in top right-hand corner -> Click settings -> Under Autofill section, click Passwords -> Turn OFF "Offer to save passwords". If by accident a password was saved, this area can be used to remove the saved password.

Separate WebServer Changes or port GIservr isn't default 4350:

By default, there is also a configuration link <http://{this servers IP address}:{portno}/webconfig> (default, <http://{server IP}:8080/webconfig>) For those on a separate server from fileProGI, you need to go here to define GIservr's location. By default, giservr is set to localhost run on port 4350 for GIservr. **This is separate from the port used to access the URL mentioned above.** To change these values, first click on the "Fetch config file contents" circled on the image below. Next change the values to point to the GIservr host machine's IP address and the port number as indicated below. Once these are changed be sure to click on the "Update file with new values" button at the bottom.



Restart your fileProWeb services; in Linux/BSD this is "service fileproWeb restart"; in Window, restart fileProWeb from Services (services.msc).

WARNING: Once the webconfig values are changed, reference to it should be removed from the fp_tomcat.cfg file avoid someone going to this link and making changes (see 8d below).

8) Tomcat configuration: (~/.fileproWeb/conf/fp_tomcat.cfg)

This file controls tomcat's relationship to fileProWeb. The default values are shown below. You may find other values in your file depending on the way you responded to the installation questions. You can change these values and restart the fileProWeb and the new values will go into effect. Let's cover the values here.

```
1 port=8080
2 baseDirectory=/appl/fileproWeb
3 /fptech=fptech.war
4 /=root
5 webconfig=wf_config.war
6 /fpPDF=root/fpPDF
```

- a) Port – this is the port number through which the web clients (browsers) will access the server.
- b) baseDirectory – the location where fileProWeb is installed. You could change this value and rename the fileproWeb folder to the new name. From a support standpoint, it is best to leave it at this location.
- c) / - this defines root location. Don't change this value as the interface requires this definition.
- d) webconfig – this defines the configuration page to define giserver address and port number. **PLEASE REMOVE THIS ENTRY AFTER CONFIGURED.**
- e) /fpPDF – Defines the location for the temp drop location for PDF documents that appear in new tabs. Usually linked to /appl/fpoffice/fpPDF during installation.

Opening your browser and using the URL <http://{server IP}:{Port}/fptech/> brings you to the login screen as seen here: (<http://{server IP}:8080/fptech/> if not modified in webconfig)

Setup Webserver

Before proceeding to install fileProWeb, let's cover a few changes within GIservr from previous versions. For those not aware of how GIservr works compared to the character versions, let's do a quick review. GIservr is a go-between for the database and the GIclient - and now the fileProWeb interface with version 6.0.1. The GIclient is a Windows PC program to login and access the database, with a mouse environment and features to interact with and launch other applications and color palettes. GIservr runs on a single server through which the clients access the database, as such it has a client/server relationship. This client/server relationship means that, for DOS uses, the programs execute on the server and not on the PC client. Please keep this in mind when setting up, as printing and file locations are from the server's point of view not the PC's. One can use the optional PFPT=on for local printing through the GIclients applications interface or the newer PDF:[pop] option to display PDF outputs for printing. (Note PFPT isn't available through the web interface, so one must use the PDF output or directly address the printer in filePro's Printer Maintenance.) The PDF option requires a mapped drive or the setup of a URL for the output to display on the PC client. Before version 6, one had to purchase a toolkit to enable messaging and dropdown features. After version 6, the toolkit is no longer needed and every 6.x version includes the messaging between the clients and the server required to launch external applications.

For the web interface to work we must install or update filePro Plus, and GIservr to version 6.0.1 or above. I'm not going to go through all those steps here as it's documented on fptech.com site. I would recommend downloading all your license files prior to installation. There's been a few changes with version 6 between GIservr and fileProGI.; most noticeable is the speed increase, and several security and stability enhancements. Starting with version 6, GIservr was rewritten from the ground up, switching it from java-based to C for each platform. These enhancements removed plain-text passwords options, moving passwords out of the login's cfg files. For existing GIservr installs, when installing version 6 a conversion takes place on the accounts to handle password and variable changes. If you install on an existing GIservr system, the installation will ask if you wish to remove unneeded variables; I'd suggest to do so. Many of these have been replaced with newer ones and are just ignored, so let it do the housekeeping. Also, during an upgrade it converts any existing login account to the new format of removing the passwords from the {login}.cfg to a single password file. If desired, one can even setup accounts to use active directory to authenticate them. (See fptech.com on-line manual for more information on how to set this up.)

After installing GIservr, I wish to call your attention to `~/fp/giserver/lib/password.cfg`, `~/fp/giserver/lib/giserver.cfg`, `~/fp/giserver/users/default.cfg`, and the `~/fp/giserver/users-/{login}.cfg`. These 4 files provide the feature options for GIservr. Starting with the `password.cfg` file, it controls password-requirements options. (Note, if you are using the newer Active Directory authentication, these variables do not apply. Instead, the authentication is passed to the Active Directory server to confirm authentication before continuing on to the application.) If this file does not exist, the default values are assumed when launched. (The `giserver` path has been added to line 5 of the `/etc/default/fppath` file.)

The following is a list of options:

~/fp/giserver/lib/password.cfg

Variable	Values	Description
minimumLength	Default 5	Set the minimum length of a password
maximumLength	Default 0=infinite, default 48	Set maximum length of a password
expireDays	Default 0=not expire	Set number of days until expired; overrides giserver.cfg value
reuseDays	Default 0=Can't reuse	Set how many days till password can be reused
reuseInstances	Default=3	Set how many password changes until a password can be reused.
requireUpper	Default false	If true, requires at least 1 upper case letter
requirelower	Default false	If true, requires at least 1 lower case letter
requireNumeric	Default false	If true, requires at least 1 number
requireSymbol	Default false	If true, requires at least 1 symbol
requireInitialChange	Default false	If true, requires an initial password change after upgrading/adding user account
complexityErrorMsg	String -default "Enter Password is not sufficiently complex"	Defines Complexity-Check error message.
reuseErrorMessage	String - default "Too soon to reuse this password"	Defines error message on a reuse-of-password failure.

The giserver.cfg file, the general configuration of Giserver, including logging locations and timeouts, etc. The list of variables included in version 6 are as follows. As mentioned before, there were other variables in previous versions that have been removed from the new design. Example: "port" is no longer used since it's within the license file download.

~/fp/giserver/lib/giserver.cfg

Variables	Values	Description
Logging:		
loglevel	Default=info (off server warning info config fine finer finest)	logging level from least to most verbose
clientLogToServer	Default=false	Client-Related fields Get own log
logAdmin	Default=false	Admin interface Events Logging
log	Default=false	Any Event Logging
logSize	Default=0 (infinite)	Max bytes before rotating Logs.
logSizeKB	Default=0 (infinite)	Max kilobytes before rotating Logs
logSizeMB	Default=5 (0=infinite)	Max megabytes before rotating logs.
maxLogRotation	Default=5 (0=infinite)	Max number of log files rotations

General:		
userActiveDirectory	Default=false	Server should authenticate using Active Directory
activeDirectoryDomain	Default=undefined	Active Directory Domain for authentication ex my_domain.com
activeDirectoryServer	String undefined	Active Directory server for authentication (IP address or domain)
serverName	Default=server name	Set Name of Server
serverAddress	Default=from server	Address to server to listen to.
adminName	Default=admin	User name for administrator
idleout	Default=0 (never)	Length of time before session is dropped when idle. 0-never
deadlockDetectionSeconds	Default=10	Time session is considered deadlocked when reading and writing becomes blocked.
adminPort	Default 4450	Admin port to connect on
admin	Default=true	Admin Account Active
remoteShutdown	Default=false	Remote shutdowns allowed from admin interface.
doesNotExpire	Default=false	Check if password expires on account
exec	Default=fpdaemon location	Usually ~/fp/fpdaemon_{os}
maxLoginAttempts	Default=0 (infinite)	Sets number of failed logins (GIclient) before automatic disconnect
restrictClientVersion	Default=false	Prevents outdated clients from logging in.

The last two files are ~/fp/giserver/users/default.cfg and the ~/fp/giserver/users/{login}.cfg. contain the variables relative to all login accounts. The variables can either be in the global account table, default.cfg, or the individual's {login}.cfg. Place your global variables for all user accounts in file default.cfg. Each login requires its own {login}.cfg file. If the same variable is in both .cfg files, the {login}.cfg values will be used. In this manner, if you have a site setup for everyone to use the same menu, you can setup default.cfg with the startcmd to cover all logins. But you can specify a different startcmd for an individual login. Example: the default.cfg file might contain:

```
startcmd=/appl/fp/runmenu dpromenu
PATH=/usr/bin:/usr/local/bin:/sbin
doesNotExpire=false
```

Note: the PATH is set in default.cfg because the PATH variable within the OS login doesn't carry through to the GIServer accounts environmental variables. doesNotExpire=false forces accounts to abide by the values set in passwd.cfg. The {login}.cfg would overrule the default.cfg file values in the following example:

```
startcmd=/appl/fp/runmenu mymenu
doesNotExpire=true
```

This account/login would start with a different menu and be required to abide by the values set in passwd.cfg, for example, "public access" accounts or some admin accounts. (Note: PATH was not changed from the default.cfg PATH.)

With version 6, comments are allowed in the ~/giserver/users/*.cfg files. To specify a comment, prepend the comment with either rem or #.

Ex:

```
# Menu name
startcmd=/appl/fp/runmenu menuname
```

Also added to these configuration files are group tags, defined as follows:

[GLOBAL] - This is the default tag. It suggests to Giserver to include the following variable(s) for all client types. A blank line will also reset the tag to GLOBAL

[GI_ONLY] - This tag suggests to Giserver that it only sets this variable for a fileProGI client.

[WEB_ONLY] - This tag suggests to Giserver that it only sets this variable for a fileProWeb client.

NOTE: These tags are ignored in the 'giserver.cfg' and 'password.cfg' files.

Example user.cfg file if I want to use local printer for GI and PDF for Web.

```
[GLOBAL]
startcmd=/appl/fp/runmenu menu
```

```
[GI_ONLY]
PFPT=on
PFPRTC=hplaser
```

```
[WEB_ONLY]
PFPRINTER=popPDF
```

Note: We recommend there be an associated Operating System account for every GI/Web account. This allows the accounts to execute with the permissions of the end-user, not the account giserver was brought up with.

~/fp/giserver/user/default.cfg |{login account}.cfg

Variable	Value	Description
startcmd	/appl/fp/p {menu_name}	Defines the starting command to launch when signing in.
doesNotExpire	Default=true	Though this is also in the giserver.cfg file mentioned above. Having it in a specific sign-on can allow one to chose account(s) exceptions from expiring.
log	Default=false	Sets if any events should be logged on an account level.
Suggested in default.cfg below: TERM	Unix values change for DOS ansi	Set Terminal type for use with other applications
PFDSK PFDIR PFPROG PFTERM PFPRINTER PFAUTOGOTOWARN	/ /appl /appl ansic popPDF off	Variables setup by FP need to be set in GI since values don't pass through from OS: PDF popup for web Turn off warning couldn't get to field before returning to the field one left when clicking mouse.
PPFORCECURSORPATH	off	Turns off the cursor Path logic on the GIclient to quickly skip to the entry field. If this is set to off any @wlf/@wef logic in between the location the cursor leaves and goes to isn't honored.
PFEDFAILBOX PFLOCKBOX	On On	Edit fail message turn to popup Record Lock message turn to popup

GIserver V6 – Account management.

As mentioned above, existing accounts from previous GIserver versions will attempt to automatically make the transition to the new format. Password locations will change from the user's {login}.cfg file to a separate password file during the next login attempt. GIserver V6 introduces a new set of commands to manage your accounts. These are: giadmin (Remote administration module, currently for Windows installs), and giaccounts (Local admin module for all other OSs).

GIserver now uses an on-line licensing mechanism which is required for fileProWeb. On first run, GIserver must contact the fpTech remote server (remote-licensing option only). After first contact, if it cannot contact the server periodically, GIserver will drop into a "grace period".

On Windows Servers, an additional -sa option is supported to run in 'Stand Alone' mode, not requiring the use of a system service. (Note, this still requires its own license file for each incident run). When installed, "giserver" is setup as a **service** to start on bootup. For Windows, just restart in your services interface. The giserver service has been configured with the following options:

OS:

BSD/Linux run as service giserver {option}
UnixWare 7/SCO /etc/rc2.d/S99GIservr {option}

Options:

start - startup GIservr
stop - stop GIservr using a kill -15 {pid}
kill - kills stubborn processes with kill -9 {pid}
status - report status and PID for giserver
restart - restart GIservr
version - report back version and compile date information

To manage GIservr accounts, V.6 now uses the command `${PFPROG}/fp/giserver/giaccount`. This is where you create the accounts and passwords for GIservr. Most options are only available as a superuser / system administrator. The options available are as follows:

Usage: giaccount option [arguments]

-m | --mode giaccount -m username (true|false)
 Set administrator flag for account. Can be tied multiple accounts, but only one connected at a time. (Requires current user to be root.)

-l | --list giaccount -l
 List all accounts saved in the password file.

-p | --passwd giaccount -p username [current_password] new_password
 Change a user's password. (Requires current password if user is not root.)

-i | --invalidate giaccount -i username
 Invalidate a user's password. (Requires current user to be root.)

-a | --useradd giaccount -a username password [template_user]
 Add a new user. (Requires current user to be root.)
 Can take an optional "template" user for initial config values.

-x | -d | --userdel giaccount -x username
 Delete a user. (Requires current user to be root.)

-c | --convert giaccount -c [username]
 Convert pre-6.0 GIservr passwords to new format. (Requires current user to be root.)

-v | --version Display version information.

-h | -? | --help Display this help message

Before proceeding to install fileProWeb components, it is recommended to test the login connection using the GIclient. Just download on a PC, set the proper IP address and connect with one of the accounts you created. If you need to diagnose something, logs are kept in the `${PFPROG}/fp/giserver/logs` folder for GI issues and `${PFPROG}/fp/logs` for character ones.

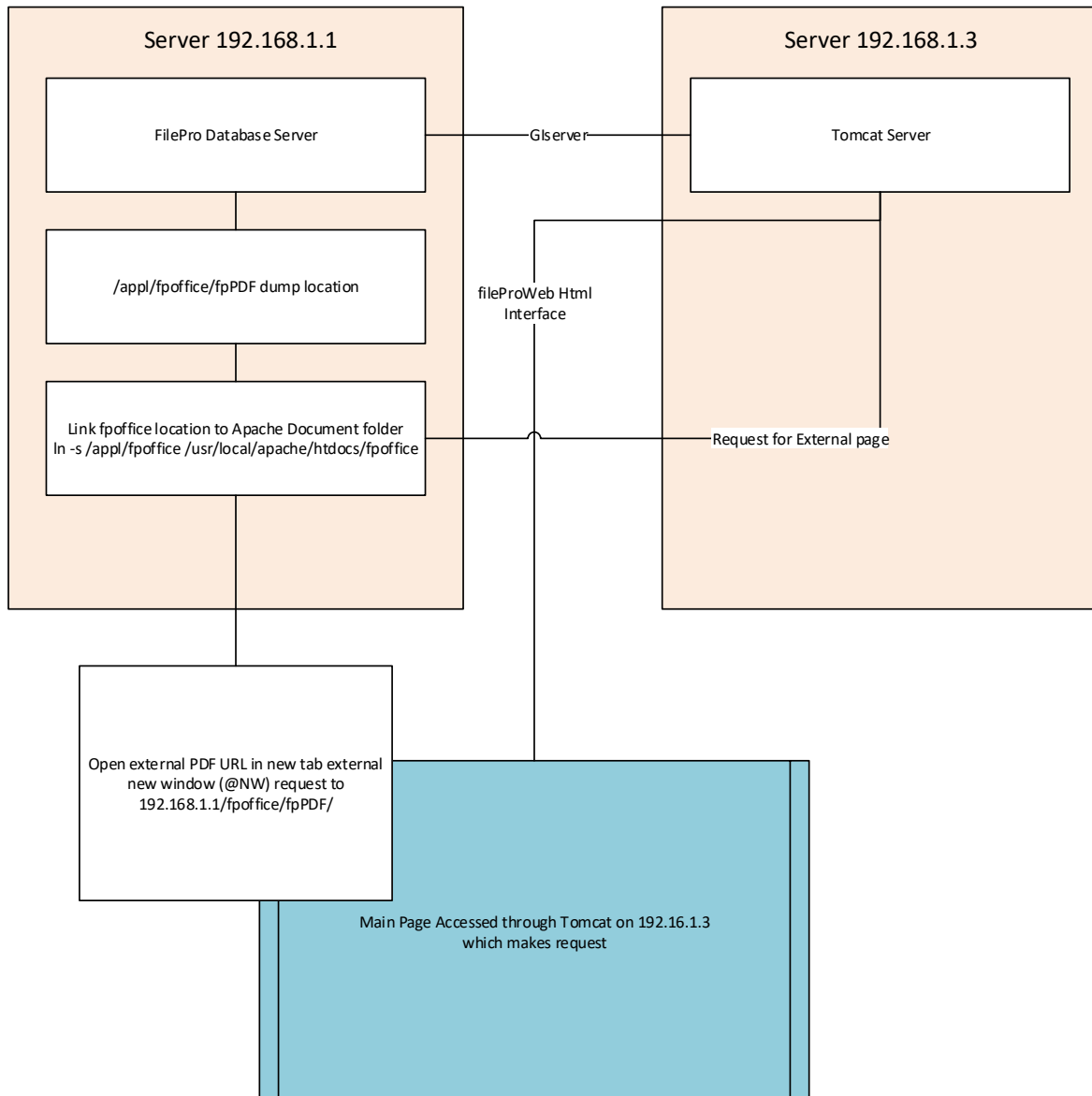
popPDF – popup PDF in Glclient

The "popPDF printer" mentioned in the variables above is similar to the PDF[Open] option for DOS versions, but works with the fileProGI and fileProWeb interfaces. When launched, it will open the printer's output in a new browser tab. **Be sure to allow popups for the specific server address within each of the client's browser.** For this to work, place a printer entry in your configuration table using the following format, substituting the server IP with the address the clients' browsers see the webserver. (Note: a printer is created for those on Linux/BSD if you answered "Yes" to the popPDF question during installation and is installed on same server as Gserver.)

Separate Server to Host Tomcat

Printer Defined as

```
popPDF,fpml,PDF:[Pop]/appl/fpoffice/fpPDF;http://192.168.1.1/fpoffice/fpPDF,PDF printer
```



Steps to setup to use popPDF:

1) Create printer using format:
{printer-name},{type},PDF:[Pop]{drop location on server};{external-location},{comment}
ex: popPDF,fpml,PDF:[Pop]/appl/fpoffice/fpPDF;http://192.168.1.1:8080/fpPDF, PDF printer

2) Create dump location that's associated to the Server's Document Root directory in your webserver. In my example, I dump to /appl/fpoffice/fpPDF

*nix|BSD

```
mkdir -p /appl/fpoffice/fpPDF
chmod -R 2744 /appl/fpoffice
chown -R filepro:filepro /appl/fpoffice
```

DOS|windows

```
mkdir \appl\fpoffice\fpPDF
```

3) Link to webserver Document Folder (/appl/fileproWeb/webapps/root/fpoffice). For a systems where GIserv and fileProWeb are on the same machine the access is a bit easier then on separate platforms required by UnixWare/OpenServer. The link is just a matter of using:

For Linux/BSD:

```
# link -s /appl/fpoffice /appl/fileproWeb/webapps/root/fpoffice
```

For Windows: (Admin command line mklink /J "Link" "Target")

```
C:> mklink /J "c:\appl\fileproWeb\webapps\webfastfp\fpoffice" "c:\appl\fpoffice"
```

If fileProWeb is on a separate server link (as UnixWare/OpenServer requires), then you'll need to setup a webserver on the database server to allow access to these documents. In this case you'd link your /appl/fpoffice folder to within the DocumentRoot folder. (Ex. In -s /appl/fpoffice /usr/lib/apache/htdocs/fpoffice , or alias the reference to the file in the configuration files). The link location can vary by OS and installation location and if you use tomcat, Apache, IIS, etc. For example, SCO has apache root Document as /usr/lib/apache/htdocs, while Red Hat might be /var/www/html. To use the Apache, check your httpd.conf file variable for DocumentRoot. **Please check OS/webserver documentation for your platform for more information, as we aren't going to cover this in details and how to handle security lockdowns.**

4) Everything within the fpoffice folder would be accessible via the Internet with this configuration as it's considered a temporary dump location; if this isn't desired, consider adding security to deny access similar to the following in your .htaccess, vhosts.conf, or httpd.conf file: (Please check your OS/apache server for up-to-date guidelines.

This example denies access to the folder unless coming from a 172. or 192. address.

```
#
# Prevent access to various directories from external clients. These directories
# may contain sensitive information that cannot be available on the Internet but
# need to be available to internal clients.
```

```
#  
<Directory "/usr/lib/apache/htdocs/fpoffice">  
  Order Deny,Allow  
  Deny from all  
  Allow from 172.  
  Allow from 192.  
</Directory>
```

- 5) Restart your webserver ("service fileproWeb restart" if local *nix/BSD; restart from services in Windows; if Apache, console OS documentation).
- 6) Create a script to empty this documents directory as they are not removed by any filePro process. An example script for linux to remove day-old documents, run as a cron request:

```
#!/bin/sh  
find /appl/fpoffice/fpPDF -mtime +1 -exec rm {} \;
```

After finishing the setup above, let's follow the flow of the graphic above. First, with PFPRINTER or default printing set to the popPDF printer, a print request is issued through the filePro database. The "print command" starts with "PDF:" so the database knows to format it in a PDF structure. The [Pop] tells this to "popup" after the PDF file is dumped. The designated temporary location is where the PDF document goes (/appl/fpoffice/fpPDF in our example). This, being linked to the webserver's documentRoot folder, is accessible by the web interface. Once the print requests are spooled, the "external location" request is sent through tomcat to open in a new tab at the requested URL of the document.

If successful, the PDF will popup the image when launched. If it doesn't launch, check permissions so that filePro can write to the folder, check that a file is written to the dump folder, and check the external location is pointing to the correct webserver address.

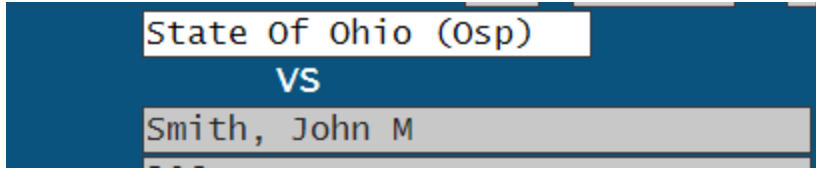
Note: this principal works with all message send commands discussed later on.

After Installation, Customize some of the features:

There are several files in common between fileProGI and fileProWeb, though the locations would be different. Within GIclient's folder was the default.ini file; this is equivalent to the web version's ~/fileproWeb/conf/webfast/webfastsite.properties. Some options to point out are:

fpUseBevelsForProtectedFields=true

If set to **true** protected fields are shown as a ghost out gray. If **false**, protected fields appear as field labels, having no color distinction or rectangle around the field. The image below reflects what the 3 different types of data fields would look like. First is the data entry field, second fpUseBevelsForProtectedFields value set to **false**, and third fpUseBevelsForProtectedFields set to **true**:

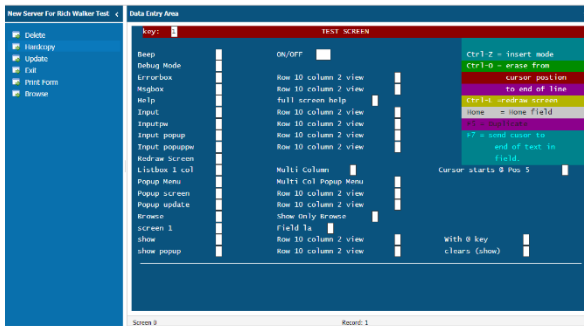


DEBtnsInWestPanel=true

If **true**, your "buttons" are placed in a menu panel on the left (for buttons defined to show on rows 22-25).

If **false**, all of your "buttons" display below the data entry area and "west panel" is collapsed for data entry screens. As in character, screen buttons/labels "show" at all rows.

True

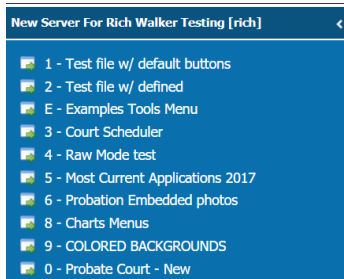


False



fpShowActionKeyInMenu=false

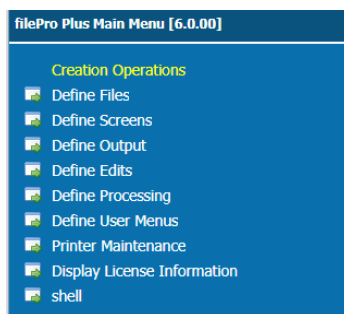
If set to **true**, the character associated to the menu-item appears as shown below; default (**false**) is to have no "action-key".



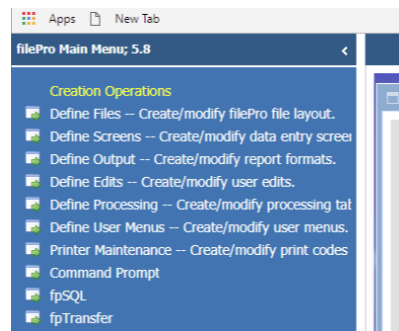
fpAppendHintsToMenuItems=false

If set **true** the "Long Description" line on menu is displayed with the main description.

False:



True:

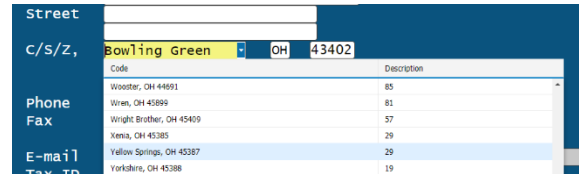


DDfileName0=/appl/fpoffice/dd0.txt (default location)

Specifies the location of the table file used to define dropdown.

Dropdowns

Dropdowns provide a quick way to fill in fields by displaying a list of preconfigured options to a user. When a sequence of characters before a field match a sequence found in a definition file, the field will then allow selections using the matching dropdown.



These definitions can be created or modified by using the fileProGI toolkit, or by using a standard text editor (see GIclients Section below on dropdowns). The example above shows a dropdown in use. When the associated C/S/Z sequence found in the definition file is on the screen, the following field becomes dropdown enabled.

With the fileProGI client, the dropdown-associated definition files have to be accessible by the local machine, either by local path or via a UNC address. This required all PCs to have access to the definition file(s). With the introduction of fileProWeb, all dropdown files are now stored on the server, allowing for changes to easily propagate to each client.

To create a dropdown definition, the file dd0.txt must be defined. It also must be referenced by the associated client's configuration file. In fileProWeb, this is the variable **DDfileName0** in **/appl/fileproWeb/conf/webfast/webfastsite.properties**.

The referenced dd0.txt file has an identical structure to that of fileProGI. It is a tab-delimited file with the following layout:

- 1) Sequential number for the dropdown definition.
- 2) Enabled (True/False).
- 3) Label to associate dropdown with.
 Note: definitions are matched in a first-defined order.
- 4) Position of dropdown (**left** is fileProWeb only).
- 5) User defined description for the dropdown (doesn't show in dropdown).
- 6) Full path to file containing the content for the dropdown.
- 7) List of column titles for the defined dropdown, each separated by a delimiter.
- 8) Delimiter to use to separate values for columns.
- 9) Row number to start retrieving data for drop downs. Allows for comments before data.
- 10) Reserved.
- 11) Reserved.

1	Enabled	Label	Position	Description	Full Path to File	Titles	Delimiter	Ignore	Lines	0	0
2	True	How Paid?	Left	Payment Type	/drop/payType.txt	Code;Type	;	0	0	0	0
3	True	City.	Left	City/S/Z	/drop/\$path/info.txt	City;State;Zip	;	0	0	0	0
4	True	?	Left	AL Yes-No	/drop/yn.txt	Code;Description	;	0	0	0	0

The file referenced by 'Full Path to File' contains the options that will be displayed to the user. A definition for a simple Yes/No field would look like the following:

Y;Yes
N;No

Where the first column is the value, and the second is the description.

The 'Full Path to File' option can also contain the dynamic value **\$path** (see above example, line 3). This is a special variable that can be set from within filePro processing using either the older message-send command, **message send handle (@P,value@)**, or the newer **xx=@GUI.P(value)** command (see Commands section for more details). These commands will replace **\$path** with **value**.

Ex. **xx=@GUI.P("foo")** will change the path **/drop/\$path/info.txt** to **/drop/foo/info.txt**.

Assume two filePro files, named ohio and florida, exist. If we only want to display the zip codes for the state-file the user is in, we could define the following in **/appl/fpoffice/dropdowns/dd0.txt**:

```
118 True ? Left AL Yes & No /appl/fpoffice/$path/zip.txt Code;Description ; 0 0 0
```

And in processing:

```
@once If: ' set $path value to @fi value  
then xx=@GUI.P(@fi);end
```

When the user is in the florida file, the content file would be **/appl/fpoffice/florida/zip.txt**.

When the user is in the ohio file, the content would be **/appl/fpoffice/ohio/zip.txt**.

If the file doesn't exist, there will be either a blank or missing dropdown.

fileProWeb themes

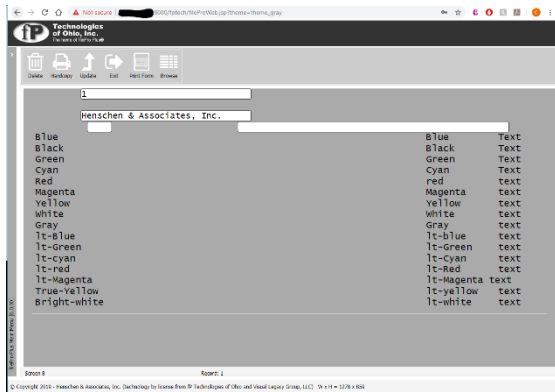
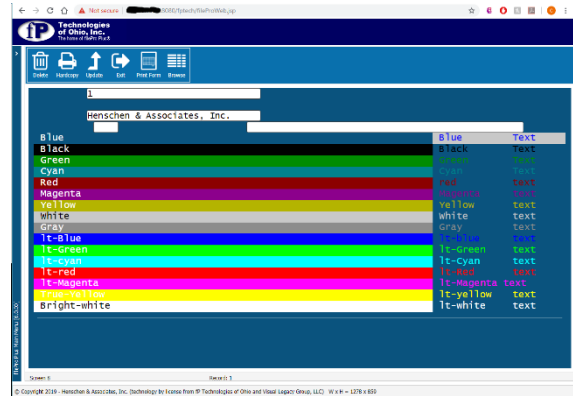
Note: All theme adjustments take place in ~/fileproWeb/webapps/root any in under ~/fileproWeb/fptech will be overwrote in future updates/installs

During installation, fPtechnologies included predefined themes. These Themes are reference by adding **?theme={theme_name}** as in this example:

http://{IP_address}:{port}/fptech/fileProWeb.jsp?theme=buttons

Here's a few examples of these predefined themes:

First, the "**default**" theme - if none is specified. All 16 colors for foreground and background are honored, and honored within FilePro provided you have "PFMORECOLORS=on" in your filePro Config file. (dpromenu option ?-F6) Buttons are displayed in a ribbon at the top of the fileProWeb screen, outside the normal 80x25 window. The bottom of the window is left for normal filePro interaction.

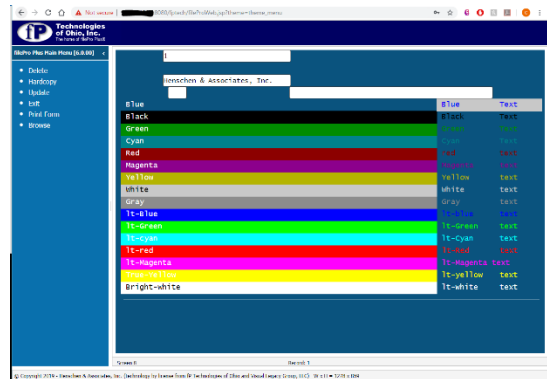


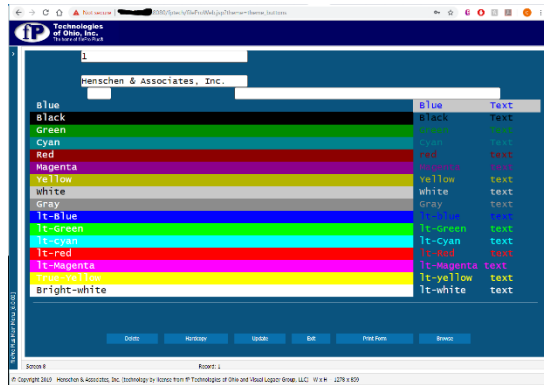
?theme=theme_gray

The Gray theme is just like the default theme in that the buttons display in the ribbon at the top of page. However, colors are masked off to be just gray with black text.

?theme=theme_menu

The Menu theme honors all 16 filePro colors. However, the default button-ribbon format is replaced with a left-side menu for button placement.





?theme=theme_buttons

Another predefined theme is "Buttons". All 16 filePro colors are honored. However, buttons are moved from the ribbon or left-side menu to the bottom of the screen, in the traditional filePro/fileProGI location.

Pretty cool that, at a single site, various session can have different layouts as soon as it's installed. Existing themes can be edited and new themes can be created. We suggest that you keep your theme changes within the `~\fileproWeb\webapps\root` folder. You can change theme-file location, but if the system is upgraded these changes might be overwritten. Staying in the root folder allows you to easily tar/zip up the folder and carry your themes to each of your customer sites.

How to edit/create themes:

Before diving into the specifics for a theme let's look at a more general java configuration file found at `~/fileproWeb/webapps/root/fileproweb/config-overrides.js`. This file defines many of the system configuration variables for the default settings used with the themes. (This is also the location to add buttons, but I'll address those in the Button section.) The parameters are defined directly or by script:

Default login graphic can be changed from `../fileproweb/images/loginPic.jpg` referencing `~/fileproWeb/webapps/root/fileProWeb/images/loginPic.jpg`

```
// Login Screen picture reference
function gavelPic(){
    return '';
}
function gavelPicNewPassword(){
    return '';
}
```

Default background graphic can be changed from `../fileproweb/images/fileProWeb.png` referencing `~/fileproWeb/webapps/root/fileProWeb/images/fileProWeb.png`

```
// Main Menu background graphic reference
centerPanelMenuHTML = '<div style="background-image:
url(/fileproweb/images/fileProWeb.png); height: 100%; width: 100%;">';
```

useDEtoolbar default is **true** to create the Ribbon toolbar; false removes the ribbon toolbar.

showMenuHints default is **false**; if set to true the "Long Description" line from "define menu" contents is also shown on the line.

westPanelWidthMenu default **400** – width for the left-hand menu in pixels.

enableLogger = true; Have logging enabled by default. Logs go into ~\fileproWeb\logs folder

serverLog = true; (needed with enableLogger=true)

processURLdelayMS=50 Receipt of SETEASTURL, command scheduled for processing after set delay

ELFont=true; Expands the edit field to use the entire range by layout positioning, not character. Has the effect of looking like more characters can be typed than field allows.

Note: After making changes like these sometime it requires clearing the cache in the browser and doing a refresh of the page.

Modification creating specific themes:

Themes are in the ~\fileproWeb\webapps\root\fileproweb\themes folder. They are made up of two files consisting of a css file and js file. The css controls the colors and formats of the labels, inputs, and menu background colors for the overall interface. This does not include the filePro application input screens, though could have an effect as with the gray theme example. For a new theme, copy one of the provide theme's css, and js name to the new theme name. Ex: copy theme_menu.css to theme_Var.css and theme_menu.js to theme_Var.js.

Edit the css file; note the three variables for color at the top of the css file.

```
/*
- main colors that will need to be changed for a different color theme are:
BG - #0970ad
BG2- #0a537e
logobar #000062

Modifications to this default css are below it in this file (search for 'Overrides')
*/
:root {
  --theme-bgcolor: #202223;
  --theme-bgcolor2: #996633;
  --theme-logobar: #333333;
}
```

Changing these variables' color can have a large effect on the overall colors.

Theme-bgcolor – changes the background of the menus and windows' outline color

Theme-bgcolor2 – Changes the Window bar below the logo at the top.

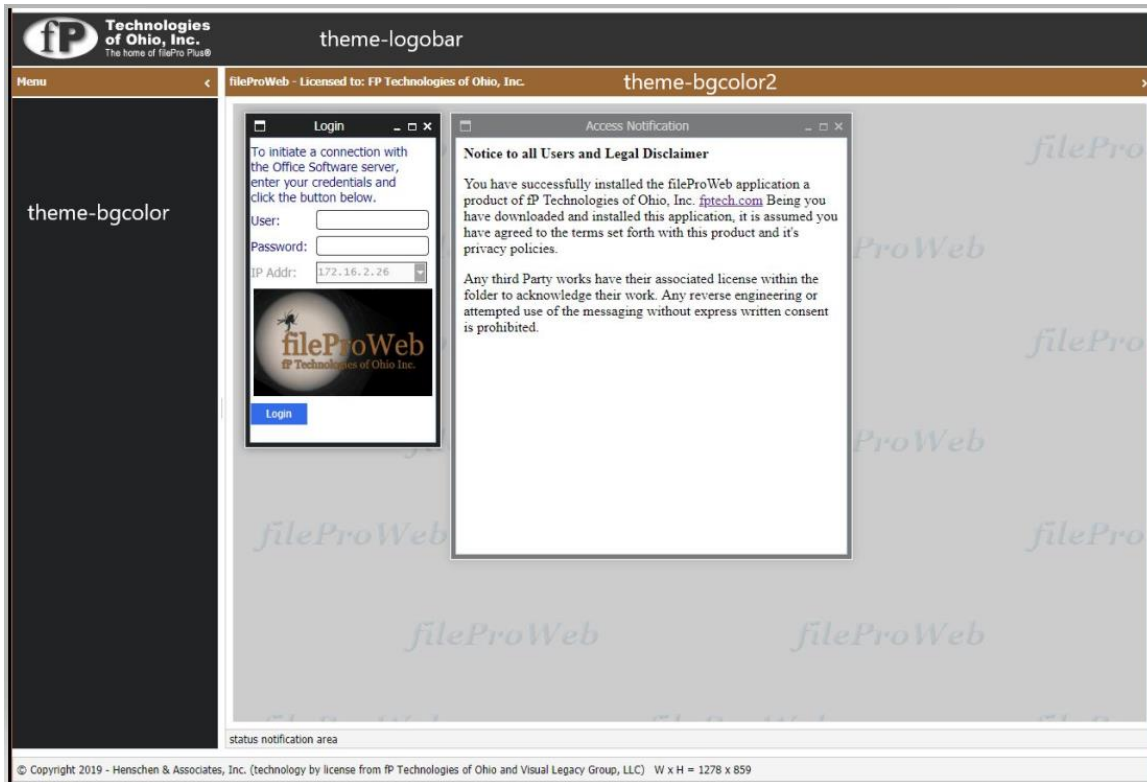
Theme-logobar – Changes the background color of the logo bar.

Also, in this css file is the specified logo used in the upper left corner and the image for the scrolling field indicator.

```
/* Logo image in upper left corner */  
div.my_hdr_logo {  
    content:url("../images/logo-filepro1.png");  
}
```

```
/* Image to indicate scrolling field */  
.fpScrollField {  
    position: relative;  
    background-image: url("../Icons/fpScroll2.png");  
    background-size: contain;  
    background-repeat: no-repeat;  
    background-position: right center;  
}
```

Changing the values as in this example changes the screen's colors to something like the image below. (The color-variables' locations on the screen are labeled for clarity.)



Moving on to the js file for your new theme, there are several other parameter you can adjust. Some of the previous variables mentioned in the config-overrides.js above are also available to use in the theme's js file. When they are in both locations, the theme's js takes precedents over the config-overrides.js file.

Note, there are references to the js file-name *within* the js file. They are located on the first and last lines of the script; be sure they match your file's name.

```
if (enableLogger) {consoleLog("theme_Var.js executed start")}  
.  
.  
.  
if (enableLogger) {consoleLog("theme_Var.js executed end")}
```

useDEtoolbar default is **true** to create the Ribbon toolbar; false removes the ribbon toolbar.

DEBtnsInWestPanel default **true**; creates the left(west) panel menu from the button listing. **false** shows the left (west) panel as collapsed when in database records.

ribbonIconsPath Specify path to find SVG icons used in buttons. Default uses buttons in ~\fileproWeb\webapps\fpotech\ribbon-white, specified as ribbon-white. However, for themes, it's best to use button icons found in ~\fileproWeb\webapps\root\fileproweb\ribbonimages either black or white folder. Specified by path ~\fileproweb\ribbonimages\black\ or ~\fileproweb\ribbonimages\white\. (You can make your own colors as well. See **buttons**)

Color_bkg – is an array of arrays, indexed by the current theme (defaults to 0) which is then indexed by 0-15 (the 16 possible fp codes for background color, possible by setting in your filePro config table PFMORECOLORS=on) and the following html is used for each region that has a background color defined. The original 16 colors you use in filePro's config file can be changed for fileProGI and fileproWeb by "color-mapping" - associating original filePro colors with a color of your choice :

```
color_bkg = [  
  //usual colors  
  '<DIV style="position:relative; width:100%; height:100%; background-color:#000000;z-index:-1;pointer-events:none;"></DIV>',  
  '<DIV style="position:relative; width:100%; height:100%; background-color:#1e1e8c;z-index:-1;pointer-events:none;"></DIV>',  
  '<DIV style="position:relative; width:100%; height:100%; background-color:#008c00;z-index:-1;pointer-events:none;"></DIV>',  
  '<DIV style="position:relative; width:100%; height:100%; background-color:#00828c;z-index:-1;pointer-events:none;"></DIV>',  
  '<DIV style="position:relative; width:100%; height:100%; background-color:#8c0000;z-index:-1;pointer-events:none;"></DIV>',  
  '<DIV style="position:relative; width:100%; height:100%; background-color:#8c008c;z-index:-1;pointer-events:none;"></DIV>',  
  '<DIV style="position:relative; width:100%; height:100%; background-color:#b4b400;z-index:-1;pointer-events:none;"></DIV>',  
  '<DIV style="position:relative; width:100%; height:100%; background-color:#c8c8c8;z-index:-1;pointer-events:none;"></DIV>',  
  '<DIV style="position:relative; width:100%; height:100%; background-color:#8c8c8c;z-index:-1;pointer-events:none;"></DIV>',  
  '<DIV style="position:relative; width:100%; height:100%; background-color:#0000ff;z-index:-1;pointer-events:none;"></DIV>',  
  '<DIV style="position:relative; width:100%; height:100%; background-color:#00ff00;z-index:-1;pointer-events:none;"></DIV>',  
  '<DIV style="position:relative; width:100%; height:100%; background-color:#00ffff;z-index:-1;pointer-events:none;"></DIV>',  
  '<DIV style="position:relative; width:100%; height:100%; background-color:#ff0000;z-index:-1;pointer-events:none;"></DIV>',  
  '<DIV style="position:relative; width:100%; height:100%; background-color:#ff00ff;z-index:-1;pointer-events:none;"></DIV>',  
  '<DIV style="position:relative; width:100%; height:100%; background-color:#ffff00;z-index:-1;pointer-events:none;"></DIV>',  
  '<DIV style="position:relative; width:100%; height:100%; background-color:#ffffff;z-index:-1;pointer-events:none;"></DIV>' ];
```

FGColorFromCode Foreground color (i.e. text color): is an array of arrays, indexed by the current theme (defaults to 0) which is then indexed by 0-15 (the 16 possible fp codes for foreground color) and the specified html color is used. The original 16 colors you use in filePro's config file can be changed for fileProGI and fileproWeb by "color-mapping" - associating original filePro colors with a color of your choice :

```

FGColorFromCode = [
  //usual colors
  '#000000', // black
  '#1e1e8c', // blue
  '#008c00', // green
  '#00828c', // cyan
  '#8c0000', // red
  '#8c008c', // purple
  '#b4b400', // yellow
  '#c8c8c8', // gray
  '#8c8c8c', // drk Gray
  '#0000ff', // lt-blue
  '#00ff00', // lt-green
  '#00ffff', // lt-cyan
  '#ff0000', // lt-red
  '#ff00ff', // Magenta
  '#ffff00', // Yellow
  '#ffffff', // White
];

```

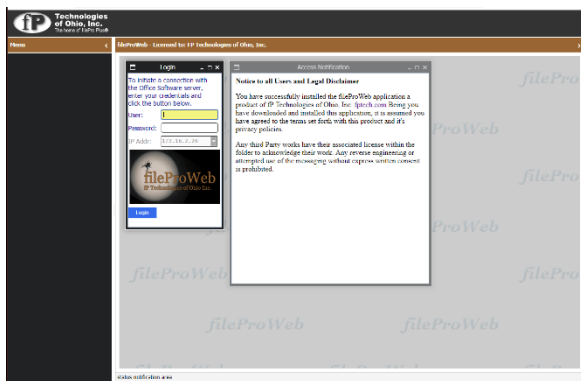
centerPanelMenuHTML – used to specify the graphic used for the main background of the menu pages. '<div style="background-image: url(..fileproweb/images/fileProWeb-t.png); height: 100%; width: 100%;">';

Note: After making changes like these sometime it requires clearing the cache in the browser and doing a refresh of the page.

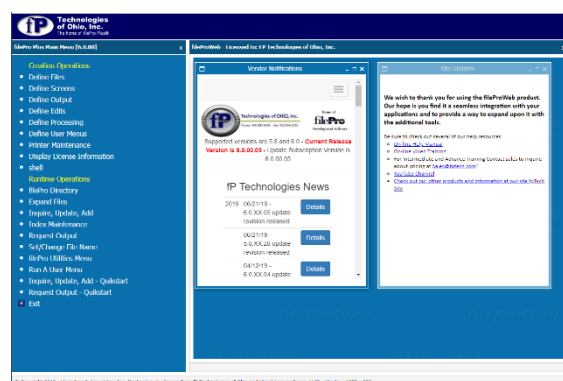
Notification windows:

Another area of the screen layout is "Notices". These notices provide a nice way for the vendor to communicate with the end users any product change, events, sales or whatever they find informational. By default, there's three such notices. The file you need to look at is ~\fileproWeb\webapps\root\fileproweb\infoWindows.js. The file is divided into two sections: 1) loginWindowInfo which displays on the login screen, and 2) menuWindowsInfo which display on the normal menu windows. They are defined in the format of:

loginWindowInfo notification area



menuWindowsInfo notification area



title
url
height,width,left,top

Take for example the above login screen's notice: The window's title is "Access Notification" which gets its value from ../fileproweb/fptech/memos/accessNotification.html. This file is easily modified to have different verbiage, and could be programmed to have filePro code generate a page for its displays each time one gets to the menu. This example defines the size of 500-pixels high by 450-pixels wide, with an offset of 260-pixels from the left. (Note, if you have linked references, it is recommended to use target="_blank" so links open in a separate tab.)

In the ~\fileproWeb\webapps\root\fileproweb\infoWindows.js:

```
loginWindowsInfo = [
  {
    "title":"Access Notification",
    "url":"../fileproweb/fptech/memos/accessNotification.html",
    "height":500,
    "width":450,
    "left":westPanelWidthDataEntry + 260,
    "top":50
  }
]
```

In the above Menu-screen example, external links are defined in the notification's html file as specified in ./infoWindows.js. We set them up in the same manner as local sites pointing to a URL.

```
menuWindowsInfo = [
  {
    "title":"Vendor Notifications",
    "url":"../fileproweb/fptech/memos/fetchVendorUpdates.html",
    "height":520,
    "width":400,
    "left":westPanelWidthMenu + 20,
    "top":50
  }
]
```

Then the html page would reference the remote page as follows, where the URL is http://www.fptech.com/fptech/fp_news.php:

In the ~\fileproWeb\webapps\root\fileproweb\fptech\memos\ fetchVendorUpdates.html:

```
<!--
* Change the URL to point to different external site. For local site pages
* change the URL in ../infoWindows.js
*
* webapps/webfastfp/fpDemo1/memos/fetchVendorUpdates.html
-->
<iframe src="http://www.fptech.com/fptech/fp_news.php" frameborder="0"
style="height:95%;width:100%"></iframe>
```

Note: After making changes like these sometime it requires clearing the cache in the browser and doing a refresh of the page.

Show and Edit Buttons in fileProGI/fileProWeb

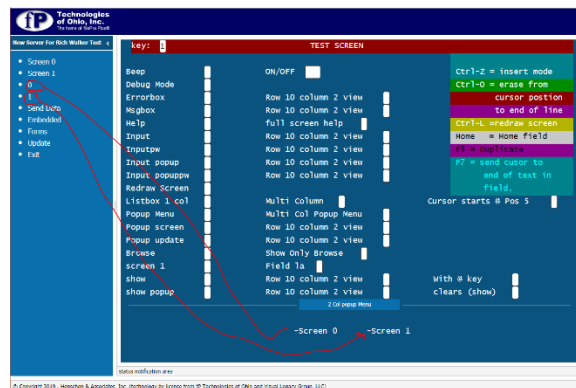
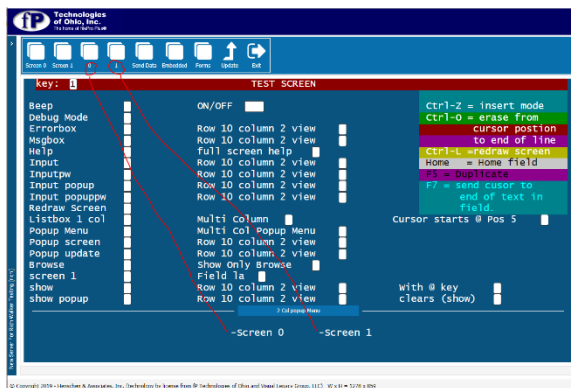
fileProGI and fileProWeb automatically convert the standard filePro keys to action buttons usable by the click of the mouse. Non-built-in, standard, filePro-defined keys however need to be programmed. (Check FP Help for details on command @ENTSEL - "Enter Selection" prompt event trigger.) FileProWeb provides a more typical "Windows" application-look by placing buttons in a ribbon on the top, rather than the default of the bottom rows, 22-24, on the screen. One can change the interface as defined in the themes section back to the traditional bottom labeled-buttons location, or on the left as a menu. Buttons defined in areas 1-21 will show at the defined location. This is not the case with fileProGI where buttons remains on the bottom like the character version today. Keep this in mind when defining the basic show button with a single character. A single syntax example is `show "{C}-Character"` The part in brackets is the clickable character. Though this has worked with fileProGI, using this in fileProWeb can have issue because only the defined clickable part of the button will show up in top ribbon leaving the text part behind. Therefore, it's not recommended to use single-character buttons in fileProWeb when defined between rows 22-24, unless `DEBtnsInWestPanel=false` is set in `~/fileproWeb/conf/webfast/webfastsite.properties` causing buttons to behave as the fileProGI product. (Check topic of themes for more detail)

```
@entsel If:
Then: showctr("22") "\{Screen 0:0} \{Screen 1:1}"
If:
Then: showctr("23") "\{0}-Screen 0 \{1}-Screen 1"
```

Compared to fully-formatted buttons (using `\{Screen 0:0}`), the default format behavior for single-character buttons (`\{0}-Screen 0`) on fileProWeb screens places the button-part on the top ribbon while text would stay on rows 22-24. Examples:

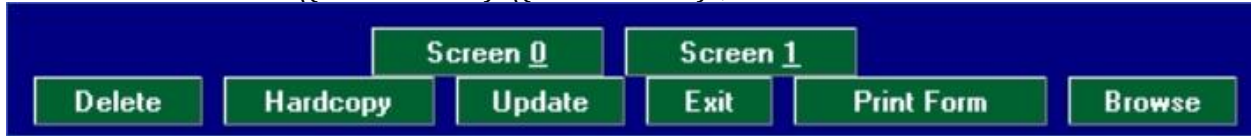
Toolbar effect with single-character buttons

Menu effect with single-character buttons.



In fileProGI there's an option to underline under the associated key use the & for the GIClient. No such association in Web, rather the '&' will show up as a character.

```
@entsel
show "\{Screen &0:0} \{Screen &1:1}";end
```



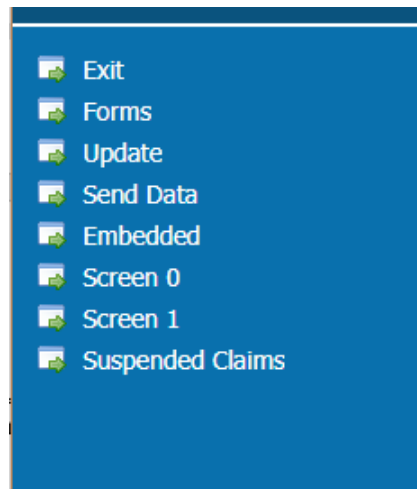
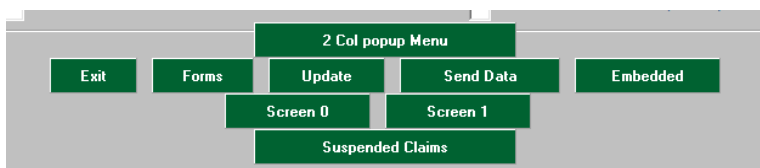
Knowing there's different behavior for button labels between interfaces, how do optimize these differences between interfaces? There is now 4 condition-values provided to determine the interface in use:

- GUI represents both GI and web interfaces
- GUI1 GI only
- GUI2 or WEB represent the web interface
- Not GUI to specify character use

Using the conditions, we can represent the character interface to show single-character letters followed by descriptions, GI with full button titles with underline characters, and Web with full button titles no underline using the following:

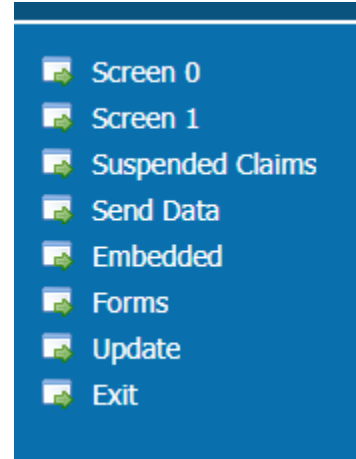
```
@entsel not GUI      ' Character Interface
show "\{0}-Screen 0, \{1}-Screen 1";end
GUI1                 ' fileProGI Interface
show "\{Screen &0:0} \{Screen &1:1}";end
WEB                  ' Web Interface
show "\{Screen 0:0} \{Screen 1:1}";end
```

Placement of the list of buttons is something to consider. Notice the two screen shots below. The listing from the Web page is in the same order as the listing of for GI.



```
■ If: GUI
Then: showctr("22") "\{Exit:X} \{Forms:F} \{Update:U} \{Send
Data:5} \{Embedded:K}"
■ If: GUI
Then: showctr("23") "\{Screen 0:0} \{Screen 1:1}"
■ If:
Then: showctr ("24") "\{Suspended Claims:S:Suspended
Claims}"
```


A more desirable order might be to place the exit button as the last item in the list for consistent location for the end-user to find. The order of the buttons along the bottom, between applications, wasn't as noticeable, but you become more aware of it in the Web interface.



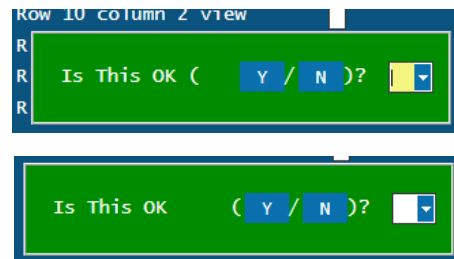
```

■ If: GUI
Then: showctr("23") "\{Screen 0:0} \{Screen 1:1}"
■ If:
Then: showctr ("24") "\{Suspended Claims:S:Suspended
Claims}"
■ If: GUI
Then: showctr("22") "\{Send Data:5} \{Embedded:K}
\{Forms:F} \{Update:U} \{Exit:X} "
    
```

Other uses of the show buttons are within inputs, input popups, message boxes, etc.

```
Input popup ok(1,yesno) "Is this OK (\{Y}/\{N})?"
```

This shows one of the issues you may find with screen layout. Notice the gap between the '(' and 'Y'. Some adaptations had to be made because of scaling and position. To cover the same area, we are forced to add space between 'OK' and '('. A couple of spaces between characters can cause spacing to change to display a more acceptable prompt.



```
Input popup ok(1,yesno) "Is this OK  (\{Y}/\{N})?"
```

All the special keys are honored and can be display using something like:

```
show "\{F1 Key:[INSC]}"
```

Some things to think about when creating buttons are how and when they should appear. To make things easier, a couple of flags, similar to '-D', were added to dclerk for execution in FilePro version 5.0.05. These are:

- D While not new in 5.0, this flag suppresses all the filePro default buttons: Delete, Hardcopy, Update, Exit, Print Form, and Browse Lookup.
- DE This suppresses the filePro default buttons, but leaves the Update-mode default buttons as Record, Cancel, and Browse Lookup. This flag is useful when the button descriptions exceed the one show line or if you want the description of the default buttons changed.
- DU This flag suppresses the default buttons while in Update mode, such as Record, Cancel and Browse, leaving the Delete, Hardcopy, etc. buttons when not in update mode.

Example of command on menu: /fp/dclerk demo -s0 -DE

Using the examples above, what happens when one goes into Update mode, with @entsel buttons defined?

Answer: 'Screen 0' & 'Screen 1' buttons stay on the screen.



If that result is not desirable, simply execute a 'blank' show command to your input processing table.

@update If:
Then: show "";restart

The buttons now clear in Update mode, as seen here, and reappear when canceling or recording from Update when @entsel is run again.



Using '-DE' command options: 'Edit fields' (4 - Define Edits) can be made into buttons in a similar manner. To accomplish this, make sure the "Suppress Status" check box in the "Application Configuration" is unchecked. (See Page 57.)

Place the following edit field in your file. (Note: support in fileProWeb has been disabled/GI3 will appear or disappear based on aspect ratio of the window.):

```
SEX '\{M:M:Male}-Male \{F:F:Female}-Female' ^~ "M" | "F"
```

(Note syntax: create the GUI button with {M:M:Male}; add -Male to display in Character.)



When the "Suppress Status" box is checked in the "Application Configuration" (in the fileProGI Developer's Toolkit, now included with version 6 as a download), the line with the "Male" and "Female" buttons will not appear, nor will anything else sent to this line, such as the "next match" button in index mode. You do not have to worry about losing command original text, since this line is not beyond the normal filePro display.

Buttons using Virtual Keys:

Starting with FilePro version 6.01 "virtual keys" have been introduced. Virtual keys allow the expansion of @key-related functions for an additional 16 "virtual keys" (<VK00>-<VK15>). Avoid using <vk00> if you want to pass values using @VF using the @wvf (explained later). They can only be addressed through processing in fileProGI/fileProWeb products, as there's no keyboard equivalent. It is addressed in the same manner one would address one of the user keys (function keys F1-F10) \{label:keys[:gui_label]} or \{label:keys} where key is [VK01]-[VK15]. The resulting value when clicking a 'button' is returned in @SK ("Special Keys").

```
@entsel If: @sk eq "VK01"      ' Check's if Virtual value was key 01 on button click non-update.
          Then: msgbox "@entsel VK01"
          If:                  ' Create button for Screen
          Then: showctr("20") "\{VK01 Test:[VK01]:VK01 Test}"
          If:
          Then: end
@update
          If:                  ' Clears button when going in update mode.
          Then: showctr ("20") " ";restart
@wuk* If:                      ' Grabs the virtual key click while in update mode @wuk
          Then: msgbox "Key is:"<@sk{"
          If:
          Then: end
```

Top Button Icon Association in Web

Moving on to the fileProWeb interface, the default interface creates a toolbar ribbon, but as described in the theme section you have 1 of 3 types of button placement. So far, the button commands work with any of these interfaces. However, the default ribbon toolbar has the added benefit of associating a graphic to the displayed button. The default button "default-icon.svg" has been assigned to the filePro standard navigation controls represented here:



However, if you want to take the time, you can represent your own icon association by modifying either the overall representation defined in `~\fileproWeb\webapps\root\fileproweb\config-overrides.js` or the individual theme's

`~\fileproWeb\webapps\root\fileproweb\themes\{theme-name}.js` file. If both files have reference to a button, the theme's file takes precedence. At the bottom of the .js file you will find a section for Application Buttons. For each image associated with a button an entry must be added or it will default to the image above. Also, note if you reference an image that doesn't exist it will not have any icon representation just a box where the image would go. The format to reference these is `ribbonIcons['associated_button_label_lowercase'] = 'image_name'`; The lowercase for the text is a must, even if the programmed label has capital letters in it.

Section pertaining to icons in `~/fileproWeb/webapps/themes/root/fileproweb/config-overrides.js`:

```
// Button Icon svg go into ~\fileproWeb\webapps\root\fileproweb\ribbonimages\white
// Format ribbonIcons['lower_case_label'] = 'representing_icon.svg';
// ** Application Buttons Begin **

ribbonIcons['costs'] = 'ios-link.svg';
ribbonIcons['photos'] = 'ios-images.svg';
ribbonIcons['notes'] = 'md-clipboard.svg';

// ** Application Buttons End **
```

In the above example, it will take a button referenced to "cost" and represent it with the "ios-link.svg" icon, the "photos" button labels referenced with "ios-images.svg icon", and "notes" with the "md-clipboard.svg".icon

ribbonIconsPath - Change the default location for buttons/ribbon icons

(~\fileproWeb\webapps\root\fileproweb\ribbonimages\white) in either the config-overrides.js or the theme's js file if desired. (Example: ribbonIconsPath = /fileproweb/ribbonimages/black/.) A different folder can be specified in either the config-overrides.js or the theme's js file you created if using a theme. As noted in theme's, though there are svg icons defined outside of ~\fileProWeb\webapps\root directory, it's highly recommended to keep this work within the root folder so upgrades don't affect your changes. (See themes for more details.) However, you can create your own set of svg images in a separate folder within the root folder and reference them if you wish.

This can also be changed within processing using the TBPATH value. This command associates a directory of icon to be used with the association buttons. The BLANK option clears the path and reverts back to the default path.

TBPATH processing reference:

```
message handle SEND "(@TBPATH,ribbonImages/white@)"
message handle SEND "(@TBPATH,BLANK@)"
x=GUI.TBPATH(ribbon_location)
```

Why use svg files? There's a couple of good reasons: 1) They are vector scalable, so they work well with different browser window sizes, 2) Can have a transparent background, 3) The colors can be adjusted rather easily. Let's look at shape-circle-white.svg in your favorite text editor. When you open it, the first thing you'll notice it's in an XML type of format. I wish to point out the tag fill="white" option. Fill can be set to any hex represented color so it could be represented as fill="#FFFFFF", so the world of colors is possible to change these. If the fill reference isn't there just add it with the color representation.

shape-circle-white.svg:

```
<svg fill="white" role="img" focusable="false" aria-hidden="true"
xmlns="http://www.w3.org/2000/svg" viewBox="0 0 14 14"><path d="M 13.5,7 A 6.5,6.5 0 0 1
7,13.5 6.5,6.5 0 0 1 0.5,7 6.5,6.5 0 0 1 7,0.5 6.5,6.5 0 0 1 13.5,7 Z"/></svg>
```

If you don't wish to create your own icons there's some good sources of images available on the internet that can be used for commercial and personal use. One such source is Diemen Design <https://github.com/DiemenDesign/LibreIcons> Be sure to read license agreements and acknowledge the creators as set forth in their terms if you wish to use these.

Processing Changes

We briefly touched on some processing options in the button section explaining the show command. Before going more in-depth with the commands, let's review the available conditions to evaluate the platform the processing is running on. These can be extremely helpful in customizing your code base on the interface platform you are running.

GUI – is used on the If line to test if one is running GI or Web used. Adding a not in front of it one can check if this process is running a character interface.

GUI1 – Test for GIClient running code

GUI2 or WEB – Test if code is being run from web browser interface.

Ex code:

```
IF: GUI1
Then: show "Running from GIClient";end
If: WEB
Then: show "Running in Web browser";end
If: not GUI
Then: show "Running in character mode";end
```

Along this same line one can test the value of the environmental variable FPCLIENT_TYPE. If the value is "1" it's running within the GIClient, whereas "2" means it's running within the web interfaces, and if "blank" it's running in character.

```
IF: getenv("FPCLIENT_TYPE") eq "1"
Then: show "Running from GIClient";end
If: getenv("FPCLIENT_TYPE") eq "2"
Then: show "Running in Web browser";end
If: getenv("FPCLIENT_TYPE") eq ""
Then: show "Running in character mode";end
```

Message Send Command:

There are several client-side functions which can take place in fileProGI and fileProWeb. For these to be accessed, the program must be setup with the message send command. (In version 6, you can use the associated @GUI commands without the handle requirement for V6). This command only has to be initialized once in order to use, and then called throughout the processing. Note: This feature is only available with the purchase of a developer toolkit license in pre-GIserver 6 versions running GIClients. The toolkit activated the message send command for all their customers under the developers account when requesting license files. For fileProWeb clients, it's built into the subscription model price.

To initialize the command a handle has to be created with the size and edit of (8,0,g). This variable then needs to retrieve a message handle value to send commands. This part is only done once within the processing table

```
@once If: ' *** first a variable must be defined
Then: declare global handle(8,0,g)
If:
Then: handle=new message ();end
```

Similarly, for a report and forms setup, these values once require a check variable:

```

If: yy eq ""
Then: Declare global handle(8,0,g)
If: yy eq ""
Then: handle=new message ();yy(1,#,g)="1"

```

Now that we have the handle defined, let's enable some of the GI features. All messages are sent through the Message handle SEND "@...@" type format unless using the newer @GUI form of the command. The new @GUI format considers options as variables requiring quotes be used if addressed with the acceptance of @GUI.SEND (**Note:** For some options to work you have to allow browser popup from the address/domain that the web server is set to.)

@GUI.SEND - allows the passing of the message send format flags without the need of setting up a handle, etc. and receive the results.

```
X=@GUI.SEND(@Message_send_option,any_flags@)
```

Example GUI.SEND uses older format between parentheses. @once If: ' Using message send Using GUI.SEND

```

Then: declare global handle(8,0,g)
If:
Then: handle=new message ()
If: ' Sets the $path value to demo
Then: Message handle SEND "(@P,demo@)"
If:

```

```

@once If: ' set $path value to demo
then xx=@GUI.SEND(@P,demo@)

```

@P- Sets Global pathname for client dropdown locations (Web and GIClient)

```

Message handle SEND (@P,{value}@) sets value for $path
Message handle SEND (@P,@) clears values $path
xx=@GUI.P(value) or xx=@GUI.PATH(value) New format for V6 not requiring handle

```

Although the same dropdown label ("entry") is on each file screen, you may want different context depending on the "filePro file" you are in. You can setup the dropdown path as described in the dropdown section, with the use of \$path in it c:\dropdowns\ \$path\entry.txt. When sending messages using the @P option, it replaces the \$path string with the value from @P.

Create two text files called "entry" and place them in directories C:\dropdown\demo\entry.txt and c:\dropdown\test\entry.txt. Now when you set the Message send command to the {"filePro file" name} the \$path will be replaced with the {"filePro file" name}. This might be setup when initializing the @once processing routine. Example of setting demo to the path:

```

@once If:
Then: declare global handle(8,0,g)
If:
Then: handle=new message ()
If: ' Sets the $path value to demo
Then: Message handle SEND "(@P,demo@)"
If:

```

```

Alternatively, in version 6
@once If: ' set $path value to demo
then xx=@GUI.P("demo")

```

Please refer to the dropdown section on setting up dropdowns.

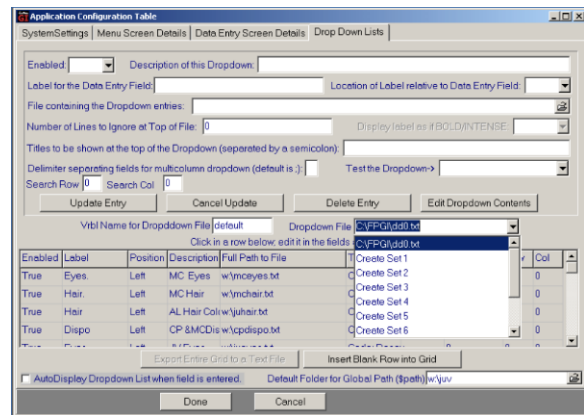
@LDD- Sets alternative drop down list definition file. Where alt_droplist is the name of the alternative dropdown list definitions. (GIClient with toolkit be 6.x)

(@LDD,alt_droplist@)

xx=@GUI.LDD(alt_droplist) Alternatively, in V6

xx=@GUI.LOADDROPDOWNS(alt_droplist)

By default, when one creates drop downs in the toolkit, the definition is going into the global list (dd0.txt by default). However, one can have up to 10 additional drop-down list of which only one can be called at a time. When called, the global and this definition are used together to define the drop downs for the screens that follow.



@keyn if ' alternative dropdown list need handle defined

Then: message handle SEND "(@LLD,c:\\fileproGI\\dd0-1.txt@)";end

Alternative V6 format:

@keyn if ' alternative dropdown list no handle required

Then: xx=@GUI.LDD("c:\\fileproGI\\dd0-1.txt");end

@TBPATH - Specify path to get theme buttons graphic representation (Web only)

(@TBPATH,{path}@)

(@TBPATH,BLANK@)

Button themes provided in fileProWeb: specify the 'path' folder-name within webapps/themes which holds the SVG files associated with the button. By using the value BLANK, it returns to the default location of graphics. (Read more about this in the button section.)

@keyn if ' New path for button graphics

Then: message handle SEND "(@TBPATH,ribbonimages/white@)";end

@keyo if ' Revert to old path for button graphics

Then: message handle SEND "(@TBPATH,BLANK@)";end

@D Display - BUILT-IN VIEWING (GIClient only)

(@D,{row},{col},{full path to display file}@)

(@D,{row},{col},{full path to display file},{row2},{col2},{full path to display file2}@)

xx=@GUI.D(r, c, filename) or xx=@GUI.DISPLAY(r, c, filename)

Alternative format in version 6

xx=@GUI.DISPLAY(r, c, filename)

(no support for second display file)

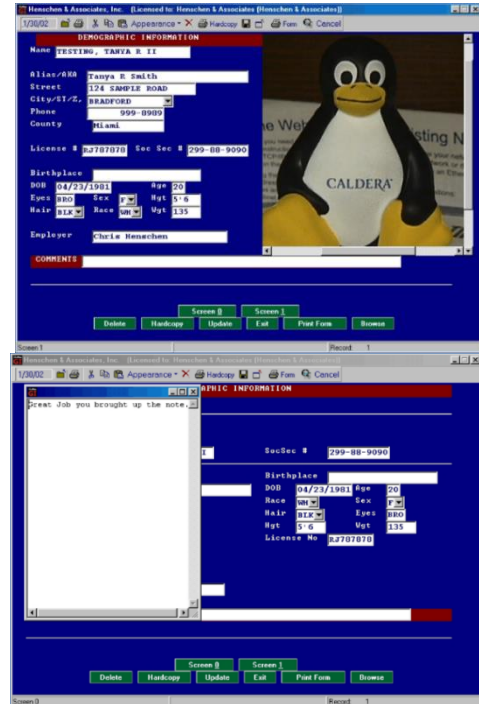
There is built-in viewing available in the GIClient. Using this viewer, one can bring up images and text files local to the client (map drives included). These file types must end in txt, jpg, or bmp. In order to see a picture appear whenever you hit the F8 key within a field, type the following within your processing table. **(Remember the @once processing mentioned above is required for this**

to work.) Since this is a GIClient-specific viewer the @D option does not work in the WEB interface. Please checkout @NW for another method below.

@wuk*

If
Then Message handle SEND "(@D,1,1,C:\{"photos\ {"pic2.jpg@})"
If
Then end

Notice the use of space quote left curly bracket quote to avoid the letter following the backslash being considered literal when dealing with Windows pathing.



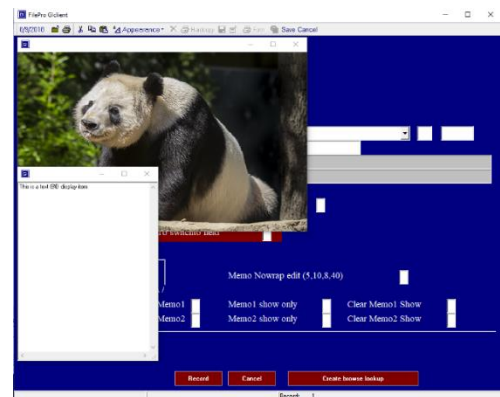
In order to view a read-only note, enter the following to bring up a text notation with the included viewer.

@keyn If ***** 1,1 represent starting row and column for display
Then Message handle SEND "(@D,1,1,C:\ {"office\ {"note.txt@})"
If
Then end

However, @D will also allow for a second item to be displayed as long as the first one has a value. To do this you have to follow the format. (@D,{row},{col},{file},{row2},{col2},{file2}@)

Message handle SEND "(@D,1,1,c:\photos\pic1.jpg,10,1,c:\office\text.txt@)"
As you will notice the two files can be of a different type from one another or of the same type. As mentioned before these can only be file types must end in txt, jpg, or bmp.

Also, note the @D option allows the viewer to be moved around with the mouse from where it is to any part of the screen, but will return to its position when executed again. The person accessing the files can close the windows as well.

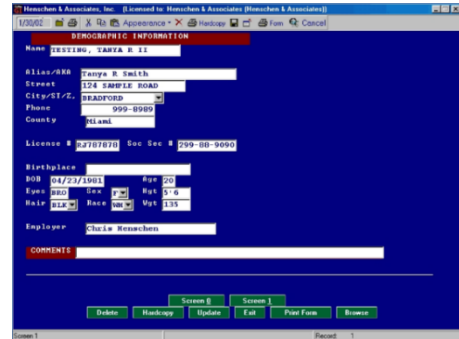


If you wish to have the processing close the viewer, it is possible to use the message syntax. The n would be replaced by a 1 or 2 to tell which window to take down.

@X - Takes down Display (Gclient only used with D)

Message handle SEND “(@X,n@)”
Message handle SEND “(@X,n,n@)”
xx=@GUI.X(“1”[,“2”]) Alternative V6 command
xx= @GUI.HIDE(“1”[,“2”])

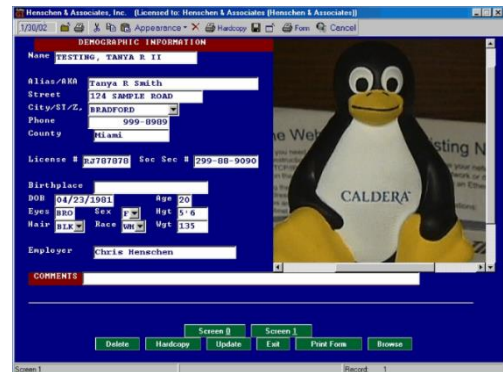
So, to remove Displayed item 1 use:
Message handle SEND “(@X,1@)”
Or to take both down:
Message handle SEND “(@X,1,2@)”



@E Embed - EMBEDDED FILES/GRAPHICS (Web up to 10 [0-10] embedded, Gclients single embed)

(@E,{[0-20]},{URL},{start row},{start column},{width},{height}@)
xx= @GUI.E(frame#, filename, r, c, w,h)
xx= @GUI.EMBED(frame#, filename, r, c, w,h)

The built-in viewer is nice, but what if you want your images to appear as part of the screen? Before doing this, it is recommended to have a cleared-off section of the screen to view the image. Otherwise screen edit fields will not show up. Secondly, once an embedded image is called, it will not disappear unless you exit that dclerk session, you call another image, or you tell it to disappear.



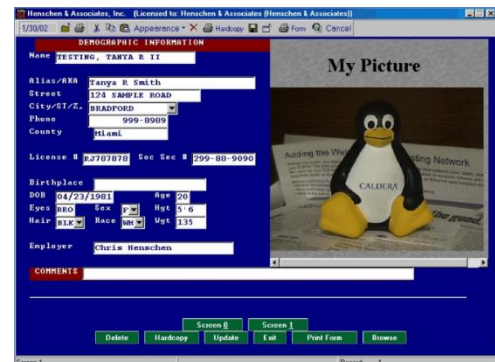
Take a look at screen 1 of the demo example (image on previous page). Notice there is a big clear space on the right-hand side of the screen. This is where we will place the pic2.jpg image we used in the viewer example. As always, with these calls the @once processing must first be within our processing table unless using the newer GIservr V6.

@keyc If
Then: Message handle SEND “(@E,C:\ “{” photos\ “{” pic2.jpg,1,45,40,18@)”
If:
Then: end

Now to close the embedded image, use the following type of process:

@keyc If:
Then: Message handle SEND “(@EX@)” ;end

If you find the image larger than the assigned viewable area, one way to solve this problem is to view the image from an html page, as shown to the right. Type the



following lines into your input processing table and then exit out to view the image. Note the @E just puts it into a 'panel' for the FileProWeb interface. (For local HTML and browser-supported files.)

```
@keyg If: ' display embedded image through Html
Then: Message handle SEND "(@E,C:\ {"appl\ {"photo1.htm,1,45,40,18@)"
If:
Then: end
```

Enhanced with fileProWeb, the embedded has been expanded to support up to 10 Embedded images per html pages. These are referenced by the frame part of the format. All 10 [0-9] can be on the screen at the same time and separately taken down using the @EX,[0-9]. **Please note if the Embedded option is use it's up to you to close it before referencing again. Otherwise you will run out of the range set for the embedded option.**

```
@keyg If: ' display embedded image through Html
Then: Message handle SEND "(@E,1,C:\ {"appl\ {"photo1.htm,11,16,15,10@)"
If: ' display embedded second image through html using V6 method
Then: xx=@GUI.EMBED(1,"C:\ {"appl\ {"photo2.htm",11,31,15,10)
If:
Then: end
```

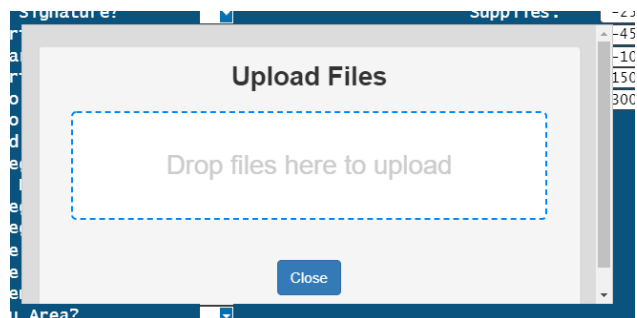
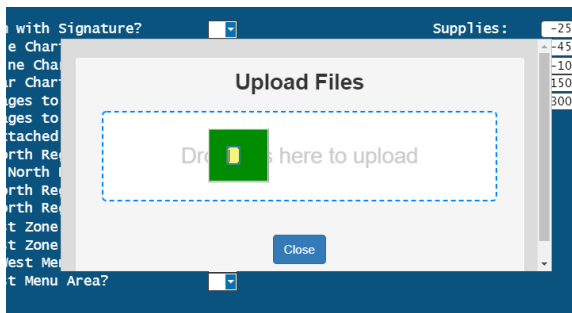
@EZ – Embedded files – Place in an iframe (Web Only)

(@EZ,{z-order [0-20]},{URL},{start row},{start column},{width},{height}@)

This command is much like the @E mentioned above. However, it has the added benefit of the added "z-order". The z-order (stack/zone ranking) can be a number from 0-20 that defines the requested URL's stack ranking above that of the normal fileProWeb one. This could be used if a programmer wishes to cover existing prompts on the screen as a type of pass through using the ahref variables explained in another section.

```
@keyg If:
Then: Message handle SEND "(@EZ,1,"http://video.htm,1,45,40,18@)"
If:
Then: end
```

Ex: Image left used @E on right used @EZ



@EF – Embedded files – Place in and iframe (Web Only)

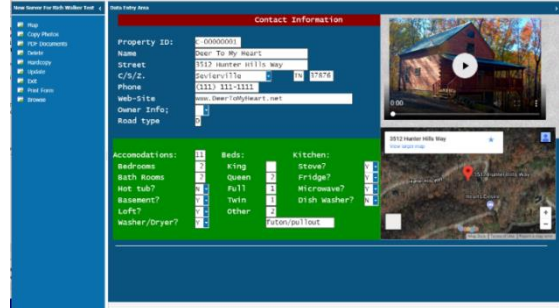
(@EF,{0-20},{URL},{start row},{start column},{width},{height}@)

xx=@GUI.EF(frame#, filename, r, c, w,h)

Giserver V6 new format only

xx=@GUI.EMBEDFRAME(frame#, filename, r, c, w,h)

Enhancing even farther, fileProWeb has @EF for embedded frame. The @EF option allows the use of external sites to be embedded within the fileProWeb screens. This has an endless possibilities of supported web tools. Looking at the syntax you'll see the first parameter requires a number from 0-9. This allows for the designation of up to 10 display areas at one time. Notice on the example screen, I have the



google map URL tied to my address so it will display the plot point when I go to the record. Also, I have an mp4 file in the top display that's playable if clicked. This also changes while I move between records. **Note: Be sure to use @EX between calls otherwise you'll exceed the number allowed. Even between references of the same page.**

@keyg If: ' display embedded image through Html

Then: Message handle SEND "(@EF,1,C:\ \"{\"appl\ \"{\"video.htm,1,45,40,18@)\"

If:

Then: end

For the display of the google map it's required that all spaces be replaced with %20 and commas with %2c this is easily changed with the "replace" command included in filePro version 5.8.03. Also, notice the use of four slashes required for the URL.

If:

Then: URL=" http://maps.google.com/?q="{city<State<zip{"<StreetAddress

If:

Then: URL=replace(URL," ","%20");URL=replace(URL,",","%2c")

If:

Then: Message handle send "(@EF,2, \"{URL{ \"&t=k&z=17&output=embed,11,50,32,10@)\"

If:

Then: end

@EFZ – Embedded files – Place in an iframe above filePro's z-order (Web Only)

(@EFZ,{Z-order [0-20]},{URL},{start row},{start column},{width},{height}@)

X=GUI.EFZ(zone,URL,start_row,start_column,width,height)

This command is similar to the @EF with the exception it doesn't follow FP's normal stacking order. Instead it appears in z stacking order of 0-20 above that of fileProWeb. This is useful to have a URL over command prompts, etc. And be able to pass responses through to filePro. Like @EZ, but within a frame. Note: if used with @EZ the z-order is related with one another for closure. **Note: Be sure to use @EX between calls otherwise you'll exceed the number allowed. Even between references of the same page.**

@keyg If:

Then: Message handle SEND "(@EFZ,1,http://video.htm,1,45,40,18@)"

If:

Then: end

@EX - Takes down the Embedded displays (Web used with @EF)

Message handle SEND "(@EX@)"

Message handle SEND "(@EX,{0-20}@)"

xx= GUI.EX([0-20]) Giserver versions 6 format. fileProGI ignores frame number [0-9,blank]

xx= GUI.EMBEDHIDE([0-20]) blank clears all.

To take down all the Embedded Displays you can use the @EX with no options.

Message handle SEND "(@EX@)"

Or to take down specific embedded images using version 6 as:

xx=@GUI.EX(2)

@EXZ - Takes down the Embedded displays (Web used with @EFZ)

Message handle SEND "(@EXZ@)"

Message handle SEND "(@EXZ,{0-20}@)"

xx= GUI.EXZ([0-20]) Giserver versions 6 format. fileProGI ignores frame number [0-20,blank]

To take down all the Embedded Zoned Displays you can use the @EXZ with no options.

Message handle SEND "(@EXZ@)"

Or to take down specific embedded images you can specify the one you want not displayed ex: zone 2:

Message handle SEND "(@EXZ,2@)"

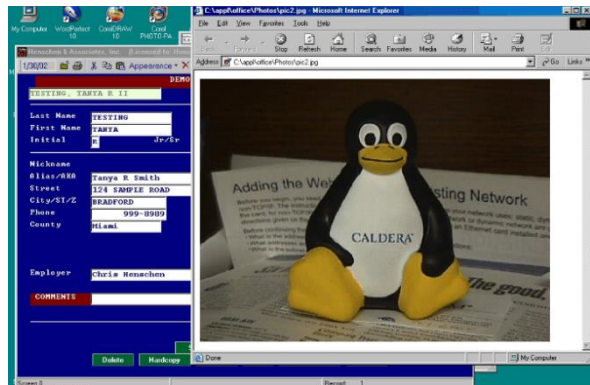
@R - Run program based on default extension (GIclient only)

Message handle send (@R,[command,]param@)

xx= GUI.R([command,]param)

Giserver versions 6 format.

For those using the GIclient you have the @R option to launch local commands on the client. This command uses the default application associated to the extension for display. For example, an htm file would bring up the page in your default browser, or an image in its associated imaging application in Windows. With the command option you can specify the applications to bring the file.



@keyr If:

Then: Message handle SEND "(@R,C:\ {" photos\ {"pic2.jpg,@)"

If:

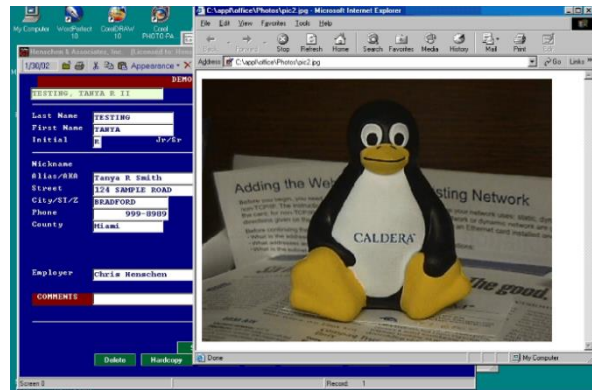
Then: end

@RH – Run program based on default extension in background (GIClient only)

Message handle send (@RH,[command,]param@)
xx= GUI.RH([command,]param)

The @RH option is much like the @R option, but launches commands in the background of the PC client. This command also uses the default application associated to the extension for display. As with the example above, an htm file would bring up the page in your default browser, or an image in its associated imaging application in Windows. With the param you can specify applications to bring the program up with if the application allows parameters.

Giserver versions 6 format.

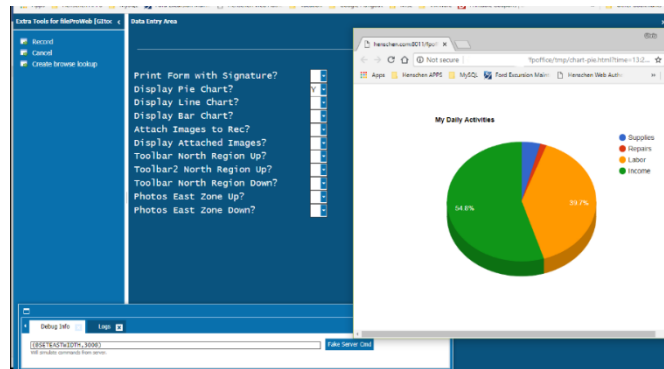


@keyr If:
Then: Message handle SEND "(@RH,C:\{" photos\ {"pic2.jpg,@})"
If:
Then: end

@NW – New Window popup (Web only)

Message handle SEND (@NW,URL,target@)
xx= @GUI.NW(url) Addition to version 6 opening in new tab
xx= @GUI.NEWWINDOW(url)

Since there's no @R option in the fileProWeb version there was added a new @NW to open resulting windows in another tab. This requires enabling browser popups from the webserver's address. This works great for displaying charts, other site's pages, output, etc. The pop option for printing was create around this flag and if you wish to have more control over output names this could be used to popup the results once they are finished printing/writing. I'll be providing more examples for this through the documentation. (Note: be sure to allow popups from your webserver to see results)



The message send commands to control these regions are:

@SETNORTHHEIGHT--- Sets north region height in pixels, 0 means no region shown (Web only)

Message handle SEND (@SETNORTHHEIGHT,height_in_pixels@)
xx=@GUI.SETNORTHHEIGHT(height_in_pixels)

Defines the number of pixels in height to set aside at the top of the browser page to display content presented by the @SETNORTHURL command. A value of 0 pixels, means no region shows.

If: Web ' open 40px top

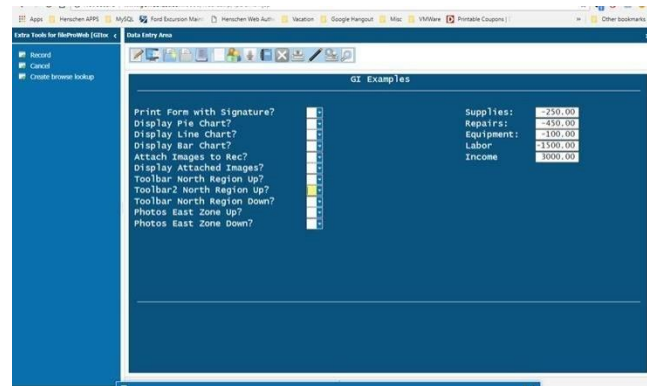
Then: message handle send "(@SETNORTHHEIGHT,40@)"

@SETNORTHURL -- Sets the URL content for the north region (Web only)

Message handle SEND (@SETNORTHURL,[url]@)
xx=@GUI.SETNORTHHEIGHT(url)

Defines the content to place in the provided space set by @SETNORTHHEIGHT height. A blank URL shows no content, but leaves the space unfilled unless told to remove. This is useful to create a html page toolbar of buttons

to use within one's applications. Above the ribbon location or if one wishes to use their own customized one. This is above the 80x25 region defined by filePro.



If: Web

Then: message handle send "(@SETNORTHHEIGHT,40@)"

If: Web

Then: message handle send "(@SETNORTHURL,http://'{WEBIPADDRESS}'/fpoffice/toolbar.html@)"

@SETEASTWIDTH-- Sets east region width in pixels, 0 means no region shown (Web only)

Message handle SEND (@SETEASTWIDTH,width_in_pixels@)
xx=@GUI.SETEASTWIDTH(width_in_pixels)

Defines the number of pixels in width to set aside at the right of the browser entry page to display content presented by the @SETEASTURL command. A value of 0 pixels, means no region shows.

@SETEASTURL-- Sets the URL content for the east region (Web only)

Message handle SEND (@SETEASTURL,[url]@)
(@SETEASTURL,about:blank@) or
(@SETEASTURL,@)
xx=@GUI.SETEASTURL(url)

Additional format in V6.

The previous two commands work hand in hand. @SETEASTWIDTH defines the width in pixels outside the normal 80x25 window to place the content in. A blank or size of 0 will close the panel. Where as @SETEASTURL defines the content to fill into the defined East region. A blank or about:blank URL shows no content, but leaves the space unfilled unless told to remove. This can be very useful to interact with the developers application as will be defined in the html section. The data entry screen area will scale its contents as needed.

If:

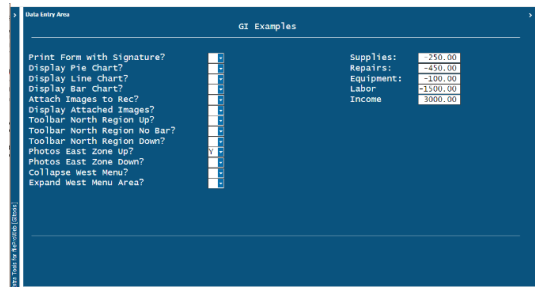
Then: message handle send "(@SETEASTWIDTH,150@)"

If: ' Populates the North Zone with an html page with buttons

Then: message handle send "(@SETEASTURL,/fpoffice/"{@rn{"/photos.htm@)"

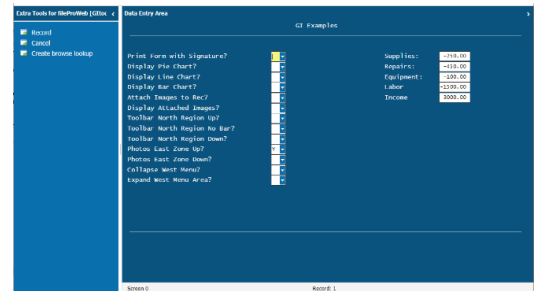
@CollapseWest - Collapse of the west region in a data entry screen. (Web only)

Message handle SEND (@CollapseWest@)
xx=@GUI.Collapsewest() ' additional v6 format.



@ExpandWest - Expand the west region in a data entry screen. (Web only)

Message handle SEND (@ExpandWest@)
xx=@GUI.expandWest() ' additional v6 format



@EMAILFILES - Dumps error log and attaches to email for debugging purposes (GI Client only)

Message handle SEND "(@EMAILFILES@)"
X=GUI.EMAILFILES()

@LOGF - Dumps a snapshot log of the GIclient to a specified location (GI Client only)

Message handle SEND "(@LOGF,c:\logs\debug.log@)"

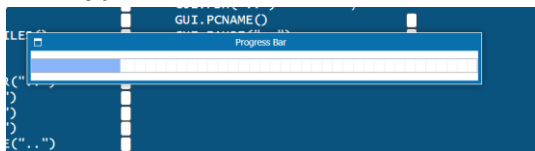
X=GUI.LOGF(desiredlogfile)

@PBS/@PBH - Progress Bar Status/Hide (GI Client only)

Message handle SEND "(@PBS,1,[0-100],title@)"

X=GUI.PBS(percent,title@)

Message handle SEND "(@PBH,1@)"



X=@GUI.PBH()

```

If:
Then: declare percent(3,0);percent="5"
loop If: percent le "100" ' continue showing bar with new value till 100
Then: x=@GUI.PBS(percent,"Progress Bar"); percent=percent+"5"; sleep "200"; goto loop
If: ' Hide Bar
Then: x=@GUI.PBH()
If:
Then: bg="";display;end
    
```

@GUI.DLG- Display a Dialog Box (GIClient)

(@DLG,box-type,{buttons},text-of-message@)

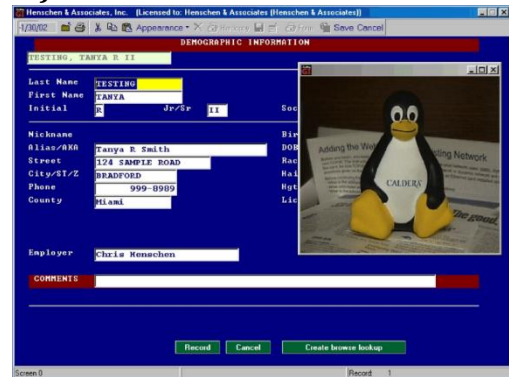
Method above retired to allow passing of values from buttons with new format below:

xx=@GUI.DLG("box-type",{ "Button Description"},"text")

xx=@GUI.DIALOG("box-type",{ "Button Description"},"text")

The "box-type" is an integer and selected from the following:

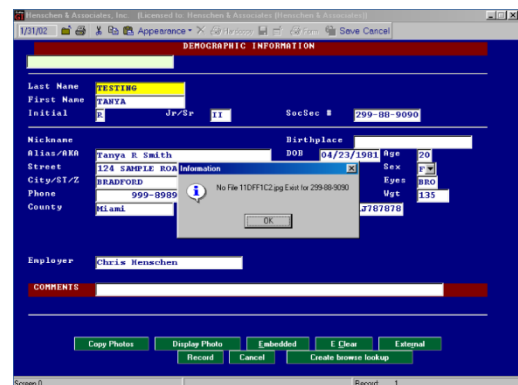
- 1 - Warning A message box containing a yellow exclamation point symbol.
- 2 - Error A message box containing a red stop sign.
- 3 - Information A message box containing a blue "i".
- 4 - Confirmation A message box containing a green question mark.



Clickable message buttons (comma "," delimited) can be created by setting 1-> on, 0-> off. Button Descriptions are listed below and must follow the order shown below.

Button Descriptions

- 1 - Yes A button with a green check mark and the text 'Yes' on its face
- 2 - No A button with a red circle and slash mark through the circle with 'No' on its face.
- 3 - OK A button with a green check mark and the text 'OK' on its face
- 4 - Cancel A button with a red X and the text 'Cancel' on its face
- 5 - Help A button with a cyan question mark and the text 'Help' on its face (unused now)
- 6 - Abort A button with a red check mark and the text 'Abort' on its face
- 7 - Retry A button with two green circular arrows and the text 'Retry' on its face
- 8 - Ignore A button with a green man walking away and the text 'Ignore' on its face
- 9 - All A button with a green double check marks and the text 'All' on its face




```
Then: declare theme;theme=mid(xx,aa,len(xx{""})-aa)
If:
Then: msgbox "URL is"<xx
```

@GUI.PAUSE()/@GUI.RESUME - Pause/Resume output to client. (Gliclients and web)

```
xx=@GUI.Pause()
Message handle SEND "(@PAUSE@)"
xx=@GUI.Resume()
Message handle SEND "(@RESUME@)"
```

```
if:
then: xx=@GUI.Pause() ;sleep "50000";xx=@GUI.RESUME()
```

@GUI.APPDIR - reports back the directory location the Gliclients is running from. (Gliclients only)

```
xx=@GUI.Appdir()
```

```
if:
then: xx=@GUI.Appdir() ; msgbox "The client is installed at:"<xx{""
```

@GUI.WINDOWSDIR - reports back the windows directory location. (Gliclients only)

```
xx=@GUI.WINDOWSDIR()
```

```
if:
then: xx=@GUI.WINDOWSDIR() ; msgbox "The client Window Directory is at:"<xx{""
```

@GUI.FW - Find Window reports existence of window with the specified caption. (Gliclients only)

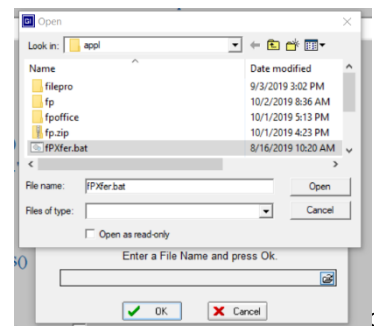
```
xx=@GUI.FW(windowcaption)
```

```
if:
then: xx=@GUI.FW("Inbox - Outlook")
if: xx eq "1"
then msgbox "The client's Mail is open"
```

@GUI.SFN - Select File Name on the Client PC returns the full path of the selection (Gliclients only)

```
xx=@GUI.SFN()
```

```
if:
then: xx=@GUI.SFN()
if:
```



then: msgbox "Location Selected on Client:"<xx{""

@GUI.DELETEFILE - Deletes full pathed files on the Client PC (GIclients only)

xx=@GUI.DELETEFILE (DeleteFileName)

```
if:
then: xx=@GUI.DELETEFILE("c:\logs\file.txt")
if:  xx le "0"
then: msgbox "Request file was unsuccessful deleted error:"<xx{""
```

@GUI.FE - Check for existence of file on the Client PC (GIclients only)

xx=@GUI.FE(FileName)

```
if:
then: xx=@GUI.FE("c:\logs\file.txt")
if:  xx eq "1"
then: msgbox "Request file exists"
```

@GUI.DE - Check for existence of a directory on the Client PC (GIclients only)

xx=@GUI.DE(FileName)

```
if:
then: xx=@GUI.DE("c:\logs")
if:  xx eq "1"
then: msgbox "Request directory exists"
```

@GUI.CREATEDIR - Create Directly full pathed on the Client PC (GIclients only)

xx=@GUI.CREATEDIR (DirectoryName)

```
if:
then: xx=@GUI.CREATEDIR("c:\mynewdir")
if:  xx le "0"
then: msgbox "Request directory was unsuccessful created error:"<xx{""
```

@GUI.ZIP - zips file(s) on GIclient to full path zip location

otherwise goes to ~fileproGI\runfiles folder, optional t includes folders, file list separated by commas. (GIclients only)

xx=@GUI.Zip(**zipFileName,[t|T],filelist[,filelist2]**)

```
if:
then: xx=@GUI.Zip("c:\zipfolder\files.zip","t","c:\documents") ;
if:  xx le "0"
then: msgbox "Request to zip file(s) was unsuccessful error:"<xx{""
```

@GUI.UNZIP - unzips file(s) on GIclient to full path zip location

otherwise goes to ~fileproGI\runfiles folder, first-T honors folder names (blank doesn't) second T overrides existing files. (GIclients only)

xx=@GUI.UNZIP(zipFileName,unzip_location,[t|T],[t|T])

```
if:
then: xx=@GUI.UNZIP("c:\zipfolder\files.zip","c:\documents","t","t");
if:  xx le "0"
then: msgbox "Request to Unzip file was unsuccessful error:"<xx{""
```

@SETCLIPBOARD- Places text in Web client's clipboard for local paste. (Web only)

Message handle SEND (@SETCLIPBOARD,text@) **(added with fileProWeb V360)**

@GUI.SETCLIPBOARD - Places text in Web client's clipboard to be pasted locally.

xx=@GUI.SETCLIPBOARD("text") text can be an expression or quoted.

```
if:
then: xx=@GUI.SETCLIPBOARD("this goes to the clipboard")
if: xx le "0"
then: msgbox "Clipboard copy was unsuccessful error:"<xx{""
```

Building HTML Tools

We discussed some of the ways to present exterior data from the data base with all the message Send commands above. These are all great and enhance the applications, but how do we interact between the data in these external/embedded pages and the filePro? Having interaction can assist in the making of a toolbar or a photo selection list, just to name a couple ways. Creating external html pages using <a href.> tags specifically designed for the GIClient and fileProWeb can greatly expand your world.

atKey:{key} (Use in your own html files with fileProWeb only)

- Pushes a key for @key routines.

```
<a href="javascript:parent.postMessage('atKey:{key}','*');">
```

Much like the button reference to the virtue keys, the atKey option allows one to associate a link to a specific character to trigger an @key event in filePro while not in an editable field. This can be a helpful tool to create something like a toolbar (not related to the automatic ribbon bar). Using the @SETNORTHURL message send command mentioned above let's create a toolbar for an example.

First, we need to setup the processing for the toolbar to appear. If you want it to appear when entering your file use the @entsel processing, remembering to setup the required variables and handles as needed.

@entsel If: ' Setup the webserver location

Then: WEBIPADDRESS="fptech.com/"

If: ' Sets up a 40 pixel zone above data input screen location.

Then: message handle send "(@SETNORTHHEIGHT,40@)"

If: ' Populates the North Zone with html page toolbar.html

Then: message handle send "(@SETNORTHURL/fpoffice/toolbar.html@)"

If:

Then: end

@exit If: ' Closes the North zone when leaving the file.

Then: message handle send "(@SETNORTHHEIGHT,0@)";end

@update If: ' leaves the specified zone, removes the reference html, when going in update.

Then: message handle send "(@SETNORTHURL,@)";restart

An example of the toolbar.html using the **atKey** href option could look like something below. To clarify for those whom aren't familiar with the href tag, the **img src** is the **icon** used and the **title** is the mouse over label that appears for the specified icon. (Color coded for association with code below.)

```
<HTML>
```

```
<style>
```

```
body {
```

```
    margin:0;
```

```
  }
```

```
table {
```

```
    display: table;
```

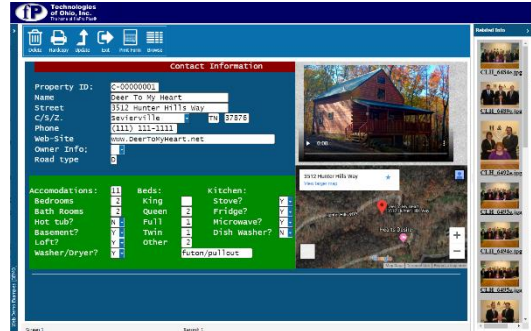
```
    border-collapse: separate;
```

```
border-spacing: 0px;
border-color: gray;
padding: 0px;
background-color: #AED6F1
}
table td {
    text-align: center;
}
table td img {
    height: 32px;
}
</style>
<TABLE>
<TR>
<! *** Delete option D **>
<TD><a href="javascript:parent.postMessage('atKey:D','*');"></A></TD>
<TD>
<! *** Hardcopy option H **>
<a href="javascript:parent.postMessage('atKey:H','*');"></A></TD>
<TD>
<! ** Update option U **>
<a href="javascript:parent.postMessage('atKey:U','*');"></A></TD>
<TD>
<! ** Form option F **>
<a href="javascript:parent.postMessage('atKey:F','*');"></A></TD>
<TD>
<!** Update option U **>
<a href="javascript:parent.postMessage('atKey:B','*');"></A></TD>
</TR>
</TABLE>
</HTML>
```

@wvf with sendVF:{content},'*' - Pass data string to @VF (fileProWeb only format)

```
<A href="\javascript:parent.postMessage('sendVF:{content}','*');\">
```

The sendVF option, much like the atkey, sends data to the server from the html page. However, this command doesn't care if you are in update mode or note. Instead, when it's referenced link is triggered, it runs a new @wvf function and passes the value in filePro as @VF. It can be a string of characters up to 1024 in length at present. This is a snippet of the code used to capture the results from the html files pictured. Note you still have to define variables for message send, etc. In this



processing we first display the photos.htm page using key p. At the time one of the entries links is selected, it triggers @wvf passing the value to @VF. Notice I prefix the links value with a reference to how to handle the response. For example, Fcm- means (field cm-) to me, while C@rn- means (Copy @rn-). These are easy removed using the replace command leaving the value I want to display, or use to rename the image name.

```
@keyp If: WEB ' If run from fileProWeb display html page
Then: Message handle SEND "(@EF, photos\photos.htm,1,60,22,22@)"
If:
Then: end

@wvf If: mid(@VF,"1","4") eq "Fcm-" ' If value begins with Fcm- use msgbox for display
Then: cm="{replace(@VF,"Fcm-","")}" ;msgbox "Image "<cm->"was selected";display
If: mid(@VF,"1","5") eq "C@rn-" ' Rename image {vf_name} to x{vf_name}
Then: vf="{replace(@VF,"C@rn-","")}" ;x=system("mv"<vf->"x"<vf)

@exit If: ' Close html page from @EF after value passed or when exiting file
Then: message handle SEND "(@EX@)"
If:
Then: end

@entsel If:
Then: show "{Photo List:P}";end
```

A shorten version of Photos.html Code for fileProWeb used in process above:

```
<HTML><HEAD>
<TITLE>Select Desired Photo</TITLE>
</HEAD>
<BODY BACKGROUND=rock.gif bgcolor="#C0C0C0" LINK="#000000" VLINK="#000000"
OLINK="#FFEBD">
<P Align="CENTER">
<A href="javascript:parent.postMessage('sendVF: C@rn-1.jpg','*');>
<IMG SRC="image\photos\1.jpg" width="100"><BR>
<B>Rename 1.jpg</B>
```

```
</A><BR>
<P Align="CENTER">
<A href="javascript:parent.postMessage('sendVF: Fcm-1.jpg','*');">
  <IMG SRC="image\photos\2.jpg" width="100"><BR>
  <B>2.jpg</B>
</A><BR>
</P>
</BODY></HTML>
```

Sendfieldleavemsg= (GIClient only format)

```
<A href="\sendfieldleavemsg=E-"{dt{"\ "><B>"&month_name<day{" "<Year"&"</B></A>"
```

The Sendfieldleavemsg is the equivalent to the javascript:fieldleave used in fileProWeb. It allows for the passing of a string of data into an editable field. As such, unlike the example with the @wvf, you must have an input command to receive values from the page.

```
If: GUI1 ' If run from a fPGI client display html page
Then: Message handle SEND "@E,V:\photos\photos.htm,1,60,22,22@"
If: ' Input to receive the image name from html file
Then: declare choice(20,*); input ("23","60") choice ""
If: ' Close html page from @E
Then: message handle SEND "@EX@"
If: ' Display message from results
Then: msgbox "Selected Image"<choice;end
```

Html Code for GIClient:

```
<HTML><HEAD>
<TITLE>Select Desired Photo</TITLE>
</HEAD>
<BODY BACKGROUND=rock.gif bgcolor="#C0C0C0" LINK="#000000" VLINK="#000000"
OLINK="#FFEB CD">
<P Align="CENTER"><A href="sendfieldleavemsg=1.jpg">
  <IMG SRC="image\photos\ 1.jpg" width="100"><BR>
  <B>1.jpg</B>
</A><BR>
<P Align="CENTER"><A href="sendfieldleavemsg=2.jpg">
  <IMG SRC="image\photos\ 2.jpg" width="100"><BR>
  <B>2.jpg</B>
</A><BR>
</P>
</BODY></HTML>
```


Data Tunneling:

Data tunneling was added to 6.0 to GIserv 6.0. From a command line within FilePro you can now pass data to and from the client and server. Note in fileProWeb the client would be considered the webserver not the end device. Note these require setting up the variable for a handle and calling it before issuing these commands.

The following transfer methods do work but are slower than other transfer methods. However, for quick transfers it might be a usable tool. Note these are originally for GIclients; as such, if you are using the fileProWeb, its server location is consider the client location when using these tools not the browser location. (Note: This is designed for small files as it's not fast and has a bit of overhead.)

Send text from a field to a file on the GIclient side (6):

(Client_filename^File_content)

If:

Then: input popup zy "Text to send in c:\\NewGI\\test_file.txt: "

If:

Then: status=@GUI.SENDMSG("6","c:\\NewGI\\test_file.txt^{zy{CHR("13")})

Send file from server to PC location (6):

(Server_filename,Client_filename)

status=@GUI.SendFILE("6","/image/photos/mcssn/no.jpg","c:\\NewGI\\no.jpg")

Receive File from PC to server destination (7):

status=@GUI.RecvFILE("7","C:\\NewGI\\FileProGI.txt","/image/photos/mcssn/FileProGI.txt")

Send tab delimited file to a grid for display on GIclient (8):

(Line1-Column1[TAB]Line1-Column2[CR][LF] Line2-Column1[TAB]Line2-Column2[CR][LF])

If:

Then: declare tab,crlf,line;tab=chr("9");crlf=chr("13"){chr("10")

If

Then: line="GRID3^Ln1-Col1"{tab&"Ln1-Col2"{tab&"Ln1-Col3"{crlf{"line2-Col2"{tab&"line3-Col3"{crlf

If:

Then: status=@GUI.SENDMSG("8",line)

Append to String Grid Contents(9):

(Line1-Column1[TAB]Line1-Column2[CR][LF] Line2-Column1[TAB]Line2-Column2[CR][LF])

If:

Then: line="Ln5-Col1"{tab&"Ln5-Col2"{tab&"Ln5-Col3"{crlf{"line6-Col1"{tab&"line6-Col2"{tab&"line6-Col3"{crlf

If:

Then: status=@GUI.SENDMSG("9",line)

Append data to specified file (19)

(Filename^append_data)
status=@GUI.SENDMESG("19","c:\NewGI\test_file.txt^{zy{CHR("13")})

Send text to a GIClient Modal (20):

(MsgHeightInPixels^MsgWidthInPixels^Caption^Text to be shown)

If:
Then: input popup zy "Text to send to modal: "
If:
Then: status=@GUI.SENDMESG("20","500^300^test modal^{zy{""})

Send text to a GIClient Modal return <CR> to Host on closure (21):

(MsgHeightInPixels^MsgWidthInPixels^Caption^Text to be shown)

If:
Then: input popup zy "Text to send to modal: "
If:
Then: status=@GUI.SENDMESG("21","500^300^test modal^{zy{""})

Cause client to send an email message (25) (GIClient only):

(From^To^cc^bcc^subject^attachmentFileName^bodyFileName)

If:
Then: line="bill@fpotech.com^support@fpotech.com^^^Subject Line^c:\ {"NewGIfp\
{"test_file.txt^"
If:
Then: line=line{"c:\ {"NewGIfp\ {"test_file.txt"
If:
Then: status=@GUI.SENDMESG("25",line)

For email to work by clicking address in screen for GIClient, you need a mail.ini file in your GIClient folder, in the format below:

mail.ini:

[Pop3]
ServerName=pop3.server.com
ServerPort=110
ServerUser=your_login
ServerPassword=your_password
[Smtp]
ServerName=smtp.server.com
ServerPort=25
ServerUser=your_login
ServerPassword=your_password
SMTPAuthenticationType=0

LAUNCHING CLIENT-SIDE APPLICATIONS (GIclients only)

Client-side applications can easily be launched within GI. For launching web sites or emails it can be as simple as a showing statement on a screen and clicking the name:



Example:

@keyw If:

Then: Show "@Go to Web site www.henschen.com";end

Processing changes and additions from GI version Comparison Chart

Message Send assumes in format of message
handle send “(@...” Whereas @GUI doesn’t need the handle, but places value in a field.

Original GI commands	New fileProWeb Commands
Processing Manipulation	
GUI	Have options GUI – GI/Web GUI1 – GI only Web GUI2 – Web only
getenv("FPCLIENT_TYPE") eq ""	Test if session is run as character
getenv("FPCLIENT_TYPE") eq "1"	Test if session is run as GIclient
getenv("FPCLIENT_TYPE") eq "2"	Test if session is run as fileProWeb
Message send(@P,{value}@) Message send (@P,@)	xx=@GUI.P(value) or xx=@GUI.PATH(value) were additionally added to V6- sets value for \$path
	Message send (@TBPATH,{path}@) Message send (@TBPATH,@) Specify path to get theme buttons graphic representation (Web only)
	xx=@GUI.IP() Returns the IP address of Client. (web and GI)
	xx=@GUI.URL() Returns the fileProWeb URL used to access it.
Launching	
Message send (@R (fileProGI only run application based on the extension of the request.)	Message send (@NW,URL,target@) xx=@GUI.NW(URL) xx=@GUI.NewWindow(URL) (Web only) - (Open in New Tab) - Running local app on Browser machine gone be sure to enable popups for the server.
	(@NW,URL@)
Message send(@E - both fileProGI and fileProWeb Message send (@D - fileProGI only Additionally added format for fileProGI only in V6 xx=@GUI.D(r, c, filename) xx=@GUI.DISPLAY(r, c, filename)	(@EF{n},Filename,r,c,w,h@) - (Embedded Frame) (@E,{n},Filename,r,c,w,h@) – (Embedded) Value n is an optional number from 0-9, value 0 when not defined. Embedded with z-orders specified above FP: (@EZ,{z-order [0-20]},{URL},{start row},{start column},{width},{height}@) (@EFZ,{z-order [0-20]},{URL},{start row},{start column},{width},{height}@)
Message send(@EX@)	(@EX{n}@) Expanded to close specific target Value n is optional number from 0-9, blank close all @E, @EF displays
	xx=@GUI.DLG("box-type",{ "Button Desc"}, "text") xx=@GUI.DIALOG("box-type",{ "Button

	Desc"}, "text") Dialog box with return response from click.
Modifications outside of 80x24 window:	
	(@SETEASTWIDTH,w@) – opens up and area to the right of the data input location w pixels wide. FileProWeb only feature.
	(@SETEASTURL,[url]@) (@SETEASTURL,about:blank@) or (@SETEASTURL,@) Places URL page right of data entry page defined by seteastwidth a blank URL removes the page from set area. FileProWeb only.
	(@SETNORTHHEIGHT,H@) much like it's SETEASTWIDTH counter part this sets aside a creates a H number of pixels above the data entry area for html pages.
	(@SETNORTHURL,[url]@) place the defined URL page in the allocated space defined in SETNORTHHEIGHT variable fileProWeb only.
	(@CollapseWest@) command the collapse the west region in a data entry screen. (Menu button section.
	(@ExpandWest@) command the expand the west region in a data entry screen. (Menu button section.
	(@EMAILFILES@) - Dumps an error log and attaches to email for debugging purposes (GI Client only Version 6)
In Html menus send results to fields from page	
sendfieldleavemsg= "\ ""{value}\ ""	javascript:parent.postMessage('sendVF:{content}','*');\ "" – place value in a field run @wvf
	javascript:parent.postMessage('atKey:{value}','*'); - event @key press no field
@GUI.XXX data transfer commands	
status=@GUI.SENDMESG("6","{file_to_create}^{content}")	Same as GI with the exception data only goes to the webserver, not end browser client device.
status=@GUI.SendFILE("6","{server file}."{to client_file}")	Same as GI but goes to webserver
status=@GUI.RecvFILE("7","{PC_file}","{to_Server_file}")	Same as GI but file comes from webserver
status=@GUI.SENDMESG("8",="(Line1-Column1[TAB]Line1-Column2[CR][LF] Line2-Column1[TAB]Line2-Column2[CR][LF])")	Doesn't display
status=@GUI.SENDMESG("19","{file to append to}^{content}")	Same as GI
status=@GUI.SENDMESG("20",HeightInPix^WidthInPix^Caption^shownText)	Doesn't display the message
status=@GUI.SENDMESG("21",HeightInPix^WidthInPix^Caption^HownText)	Doesn't display the message
status=@GUI.SENDMESG("25",	Doesn't work on fileProWeb

Revised: Sep 4, 2020

From^To^cc^bcc^subject^attachmentName^bodyFileName)	
---	--

Setting up a GIclient executable (not Web related)

After modifying all the parameters within your development system, the next step is to configure a client.

- 1) Change directories to the ~\fp\fpGIdev folder.
- 2) Run fpGIdev by double clicking on the appropriate icon.
- 3) Click on the "Setup Configuration" button (see right).
- 4) Click on the "Save Configuration Button" (see right). A statement may appear stating "the deployable configuration file...\fpgiConfig.dat already exists." If so, answer "yes" to override it.
- 5) Once this has been saved, exit from the fpGIdev file.
- 6) Create a directory called fileproGI, or fpgi. (This can be done by right clicking in the fpGIdev folder and selecting New Folder, and then typing in the new name.)

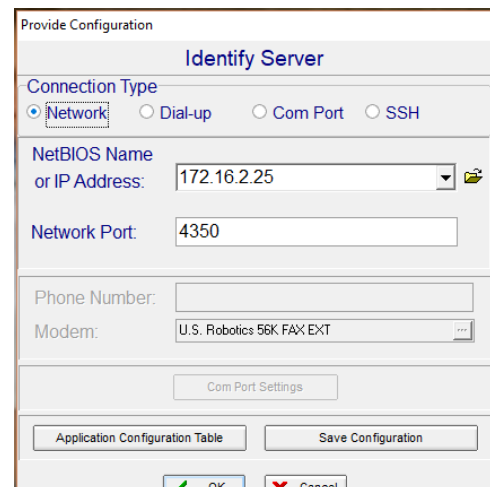


1-"Setup Configuration" button

- 7) Copy fpgiConfig.dat from within the ConfigFile directory over to your new directory.
- 8) Take the GI runtime files, fileproGI.exe and copy to this directory.
- 9) Copy your dd0.txt file to this directory.

You are now ready to run your new runtime version. This directory could be compressed and turned into an executable installation using a program such as winzip. You would then be able to extract files to other computers at the site

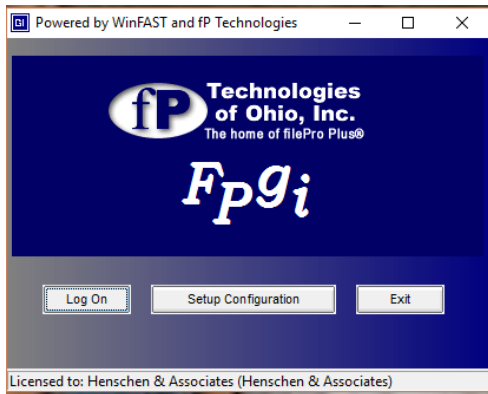
Application Configuration Table - TOOLKIT



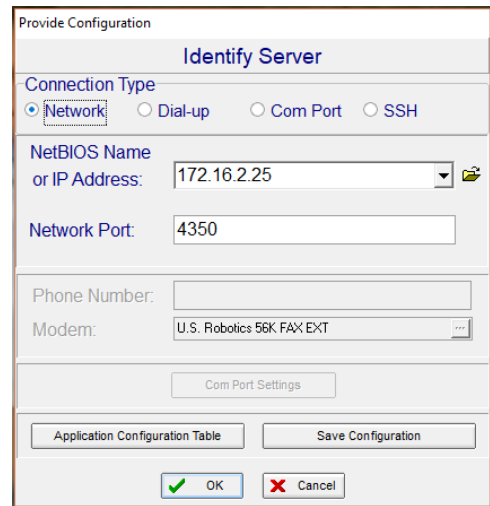
2-"Setup Configuration" button

Application Configuration Table – DEVELOPER (Toolkit)

The Toolkit is the module which opens the door to features similar to those you would find using Windows. When the Toolkit is loaded there is a fpGDev directory within fp with an executable fpGDev.exe. When opening fpGDev.exe, and clicking on the setup configuration you will notice a couple of options the runtime version doesn't have. Clicking the "Setup Configuration" button will bring up the "Identify Server" window. Click on the "Application Configuration Table" button; you will see tabs along the top of the screen: System Settings, Menu Screen Details, Data Entry Screen Details, and Drop Down Lists.



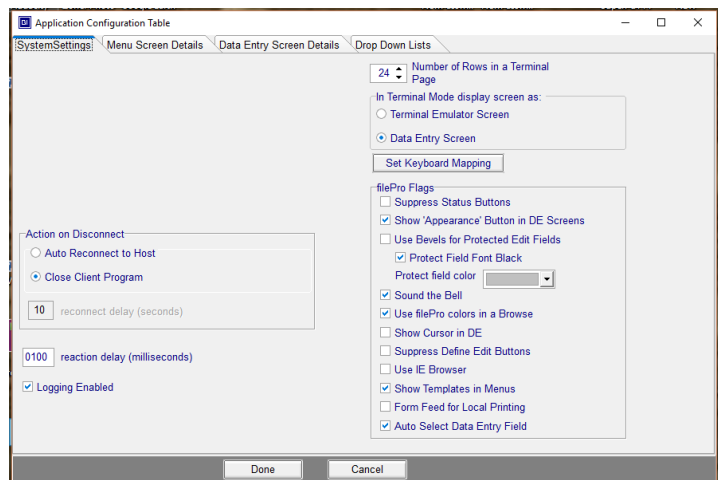
4-Setup Configuration Button



3-Application Configuration "Table" Button

System Settings

To the right, you will see the "System Settings" screen. By checking "Logging Enabled" a file will be created within the local directory of the executable of the GIClients. This file, fileproGI.*.*.*.dat, will only grow to a maximum, so there is no need to worry about filling your drive space. It is recommended not to adjust any of the items on this page except for those located within the "filePro Flags" section. In the "filePro Flags" section, you will see a list of check boxes.



5-"System Settings" screen

Suppress Status Buttons - This will suppress edit descriptions defined with the new \{} style buttons (see edit buttons on page 15). If the edit doesn't contain the button format within the edit description, the text will show on the last line of the screen. This is helpful if you prefer to use drop downs but do not desire to have the edit show in GI; however, in the character version the edits will remain.

Use Bevels for Protected Edit Fields - This enables a three-dimensional look to protected fields on the filePro screens.

Sound the Bell - Enable/disables the error bell sound.

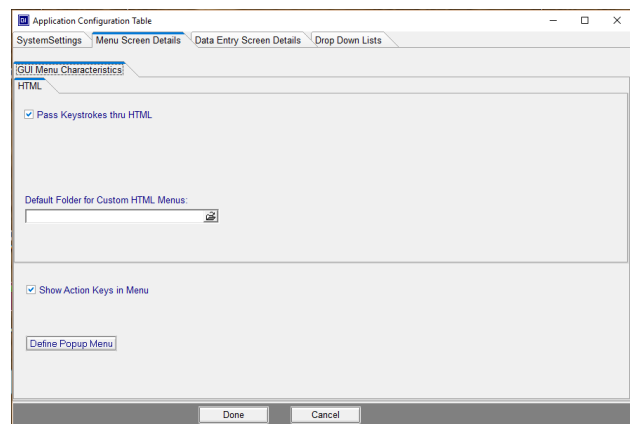
Use filePro colors in a Browse - Enables/disables the use of filePro colors in a browse window.

Show Cursor in DE Screen - Enable/disable the blinking block cursor on the screen usually associated with character screen types.

Show 'Appearance' Button in DE Screens - This allows you to enable the "Appearance" option on the toolbar of the data entry screens, and the "Option" option on the toolbar of the menu screens. Enabling will give the user the ability to change the colors and characteristics of the screens.

Menu Screen Details

To the right, you will see the "Menu Screen Details" screen. To get this view you may have to click on the GUI Menu Characteristics to see the HTML tab. The Default Folder for Custom HTML Menus is located here. If a path is placed here, any calls to html menus are relative to this location. Developers who want to use absolute paths should leave this blank, otherwise, the call to the absolute path will be appended to the default folder.



6-Menu Screen Details" Screen

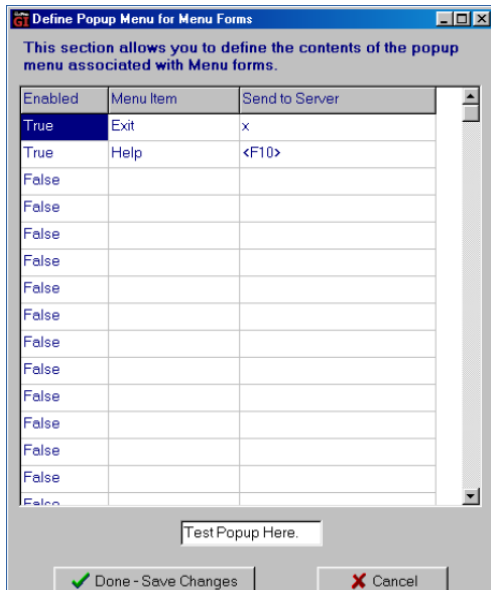
You should uncheck the "Pass Keystrokes thru HTML" box if you don't want the keys passed through. This is used in custom html menus.

Action keys are the labels for menu items, which appear to the left in character versions. Checking the “Show Action Keys in Menu” will show these “action keys”, while unchecked the action keys will be hidden.

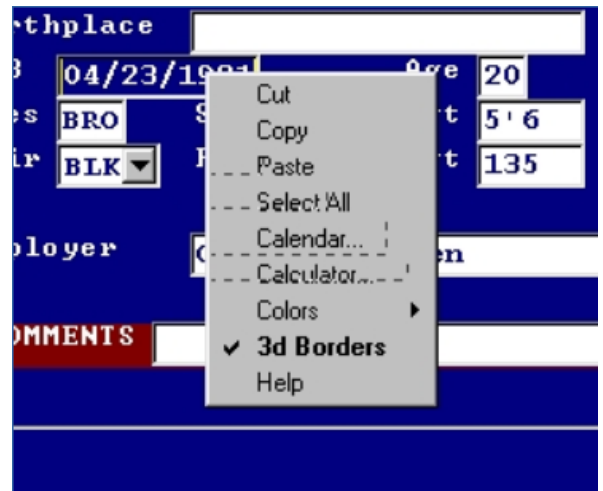
The “Define Popup Menu” button consists of a table of commands, which show up when you right-click on menus.

To add features:

- 1) Under the “Enabled” column, change the statement from “false” to “true.”
- 2) Under the “Menu Item” column, type in the label you want to appear in the right-clicked menu. Example: “Help”
- 3) Under the “Send to Server” column, place the keyboard option you wish to send upon selecting the item. Example: Click “Help” would be the same as pressing <F10> key.
- 4) To preview, right-click on the Test Popup Here edit box.
- 5) Finally, click the “Done-Save Changes” button.



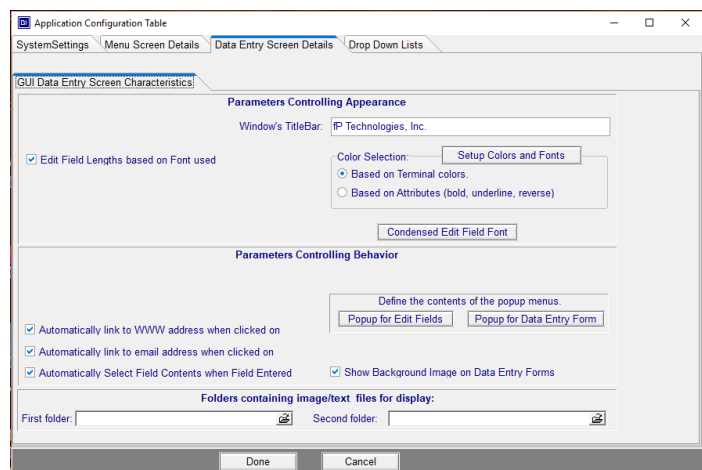
8-“Define Popup Menu”



7-Actual Popup Menu

Data Entry Screen Details

At the right, you will see the “Data Entry Screen Details” screen. As you can see, the developer is able to place their company’s name or other information in the “Window’s Title Bar” field. It is recommended to leave the color selection based on the terminal color, instead of based on attributes. However, feel free to



change the colors under the “Setup Colors and Fonts” button.

9-“Data Entry Screen Details” Screen

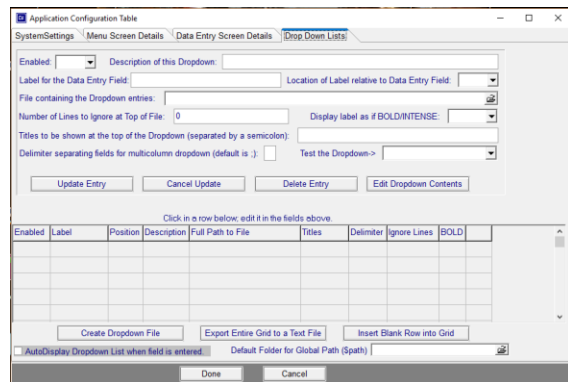
Colors are associated with the color on the character screen. For example, to set the colors of this blue/white screen with red/white reverse, follow the background color for blue down to the foreground color for white in the “Customize Form Colors” table. Click the box where the two colors meet. (See Colors section under appearance for more detailed information)

If we return to the “GUI Data Entry Screen Characteristics” screen, you will see that the check boxes for automatic links to WWW addresses and email are set to “off.” By checking these boxes, text formats such as mail@fpotech.com and www.fpotech.com become executable by double-clicking the mouse on the titles. The “Popup for Edit Fields” button is the same as the “Define Popup Menu” button mentioned above, except that it works only in the edit fields. The “folders containing image/text files for display” have the same conditions mentioned above for html menu paths; however, if absolute paths are desired do not place anything in these locations.

Dropdown Lists

Dropdowns can be added to give you a Windows look and feel. In order to create these dropdowns you must have the toolkit component. Dropdowns consist of text files defined by the programmer. In order for these dropdowns to appear, a consistent set of field labels on the screen must be adhered too. Once the programmer knows the label context used to trigger the dropdown, they will highlight an empty line, or the item to be changed, from the browse list at the bottom of the screen. At this point, the user can define the dropdown using the information at the top of the screen.

To the right, you will see the “Drop Down Lists” screen. You will need to fill in the following fields in order to create a dropdown. If you’d like a different file or location from the default dd0.txt file you can double click in the “Dropdown File” locations edit field. You then will be prompted for the new name and path.



Dropdown Tab in fileProGI Toolkit

Let’s say you wish to have several sets of defined dropdowns. The client allows one to have 1 default list and 1 additional one from the maximum 10 defined ones. The default one is active regardless if you choice other. To define a separate name for each list in the “Vrbl Name for Dropdown File” and an associated file under the Dropdown File. Ex:

Vrbl Name for Dropdown File=default Dropdown File=C:\fileproGI\dd0.txt
Vrbl Name for Dropdown File=extra1 Dropdown File=C:\fileproGI\dd1.txt
Vrbl Name for Dropdown File=extra2 Dropdown File=C:\fileproGI\dd2.txt

Then in programming using the Message Send command use (@LDD,extra1@) and (@LDD,extra2) to reference the desired definitions. More on the message send command later.

Enabled - By selecting “true” it will allow the programmer to have an active dropdown defined. If set to “False” it will make the dropdown inactive.

Description of this Dropdown – A user defined description or comment used to describe the dropdown.

Label for the Data Entry Field – This is the field label used to determine whether or not a dropdown should be placed on the screen. This includes both inputs and input popups.

File containing the Dropdown entries – This is the text file that contains the dropdown text. Clicking on the file folder image to the right of this field allows you to navigate to the desired text file and edit it from this location.

Number of Lines to Ignore at Top of File – Usually this would be “0”; however, if you need an offset from the top of the field which will NOT display headings, it is provided.

Title to be shown at the top of the Dropdown – This is where you would enter the title(s) of the columns that will be shown in the dropdowns.

Delimiter separating fields for multicolumn dropdown – This is a user definable delimiter for cases when the semicolon might be used as part of the dropdown descriptions.
Example: Code;Description NOTE: no spaces

Search Row & Col – Zero being the default allows for the checking of the field label in any row or column. If this is changed to a number, the text for the dropdown must reside on that row or column only.

Test the Dropdown – After defining the dropdown, click the “Update Entry” button. This records the changes to the list shown during the dropdown test.

When all adjustments have been made, click the “Done” button at the bottom of the screen to store your changes.

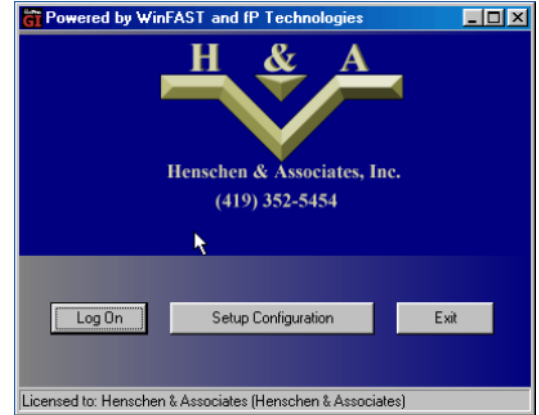
Execution with title format:

If you wish to have your shortcut setup to reference different titles you can add the title tags as shown here:

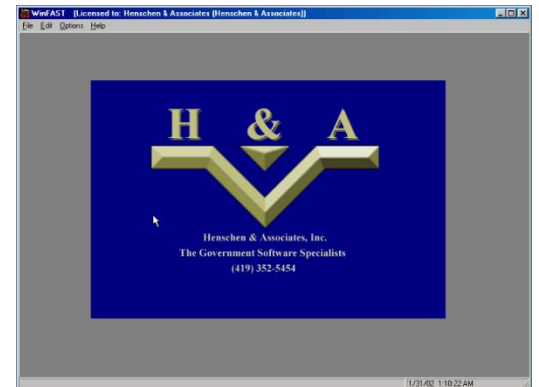
	GIclient version 3x	GIclient version 2x
Add title to launch window	fileproGI.exe "title:MyTitle"	fileproGI.exe "title=My Title"
Specify a non-default ini. Require all default.ini, default2.ini, and default3.ini references as well.	fileproGI.exe ini:whatever.ini	fileproGI.exe whatever.ini

Miscellaneous Changes

Notice that my splash screens have our company logo on them and not the filePro Trade Mark. To change the filePro Splash screen, go to the ~\fp\fpGIddev\runfiles directory. In this directory you will find a file called IntroImage.bmp. This file, if you maintain its size and format, can be changed to be your introductory screen with most paint programs.



Now let's look at the screen you receive if you exit without logging in. This background can be changed as well in the same runfiles directory. This image name is called Splash.bmp. Have fun adding your own.



If you are familiar with html coding you can perform minor changes to the default menus, which come up. However, it would be wise to make backup copies before making any changes to these files. Within the ~\fp\fpGIddev\runfiles directory notice two files:

MenuTemplate1col.htm
MenuTemplate2col.htm.

With slight modifications, the non-filePro related links can be changed to link to other items or images using a text editor. Instead of going into detail, here are a couple of examples one could use.

1) ``



This would bring up the “ \documents\example.doc” in the default Window application associated with the 3 character extension. Example: “.doc”

This could be used for an “on-line help” manual system.

2) ``

This brings up an external html browser and goes to the site mentioned. Great to point users to your site for updates, etc.

3) ``

This brings up the Windows mail program. Following the [mailto:](#) one could add an email address. Example <mailto:support@fpotech.com>.

These changes become universal, in that all menus have the same linking information. Please note that this is a very basic Browser used for the menus. Java and some html codes do not run with this environment.

Changing colors in GIClient (Not Web Related)

There are basically 3 different part of the application you for which you can change colors; the Browse Window colors, the Data Field colors, and the Screens colors.

1st there's the Browse window colors. These are represented by an embedded popup screen with a scrollbar to the right of it as shown:

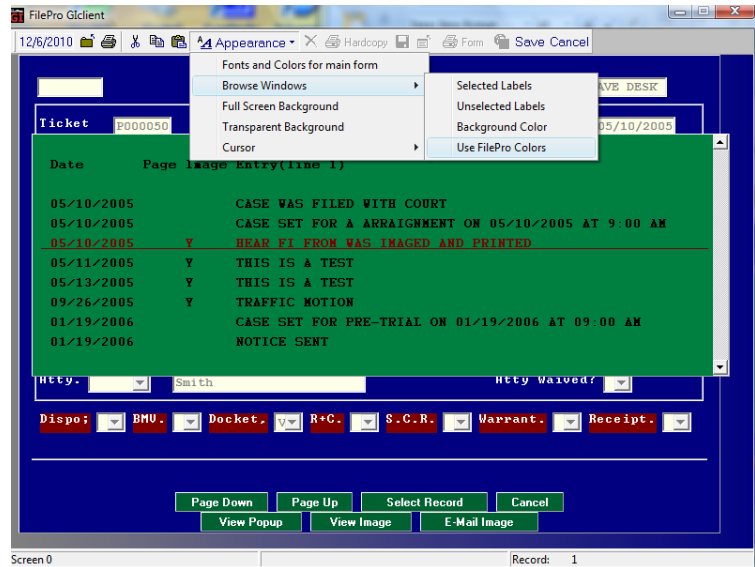
To change these colors, go to appearance -> Browse Window. Here you will be presented with four options.

Selected Labels – the highlighted entry for the record to select text color. (red in this example)

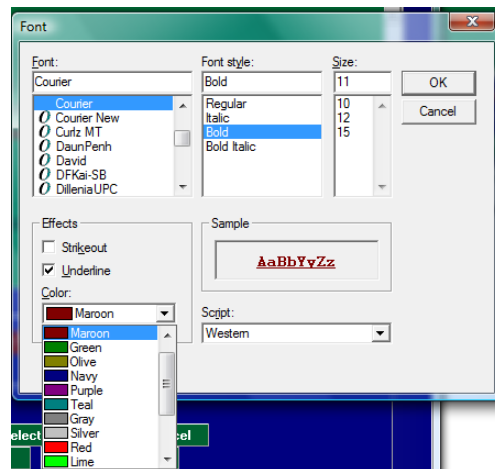
Unselected Labels – The non-highlighted text color. (Black in this example)

Background Color – (Green background in this example)

Use FilePro Colors – makes the browse screen take on the color attributes of the filePro colors from character mode.

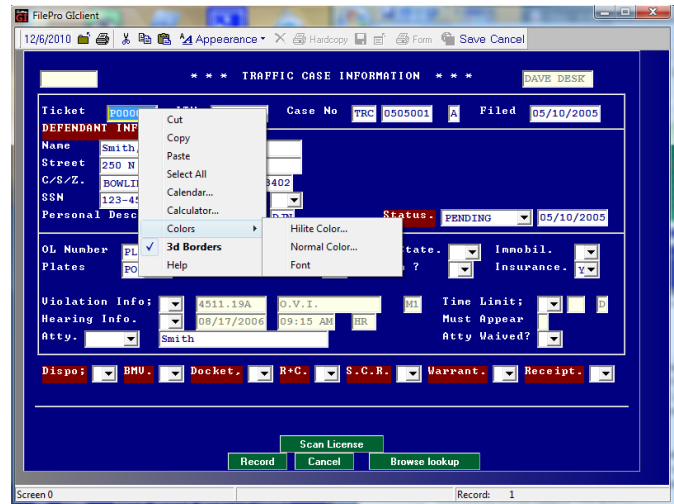


Select any of the non-filePro color options and you are presented with a font adjustment screen. Here you can change the color associated with the entry by selecting one of the dropdown colors as shown here. (Please don't change the font type from the Courier or the size from the 11.)

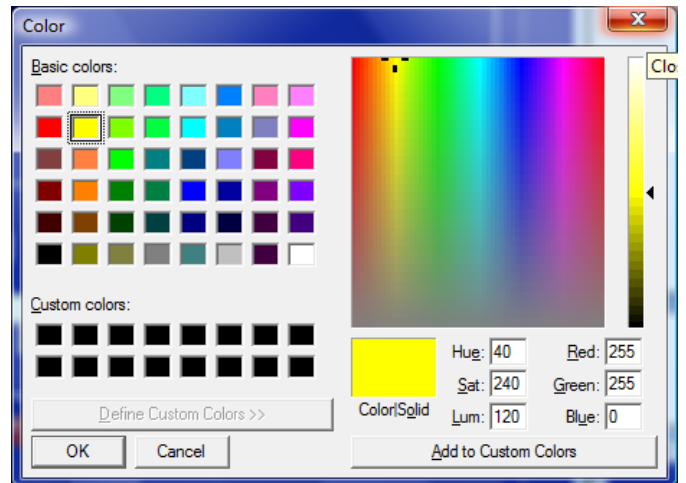


The Last option "Use FilePro Colors", if checked, will revert the color template to that of the character filePro screen. It may best to leave this unchecked to have full control of the browse window's color palette.

2nd Data Field colors options are for those in the data entry fields. Usually these are defaulted to white background with blue letters. To change these, you must go into update mode and right click in the data entry field. Mouse down to colors and you'll be presented with three choices.

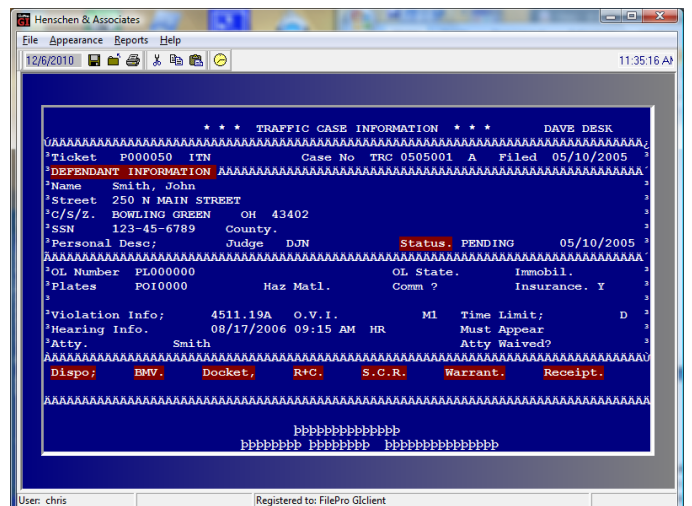


Hilite color - the color of the background where the cursor is located. You are presented with the selection to the right and just select the color.
Normal Color - the color of the background when the cursor is not in the field. You are presented with the selection to the right and just select the color.
Font - the text color. This is presented in the font screen as shown in the previous section. As above only change the color and not the font size of the screen alignment will be distorted.

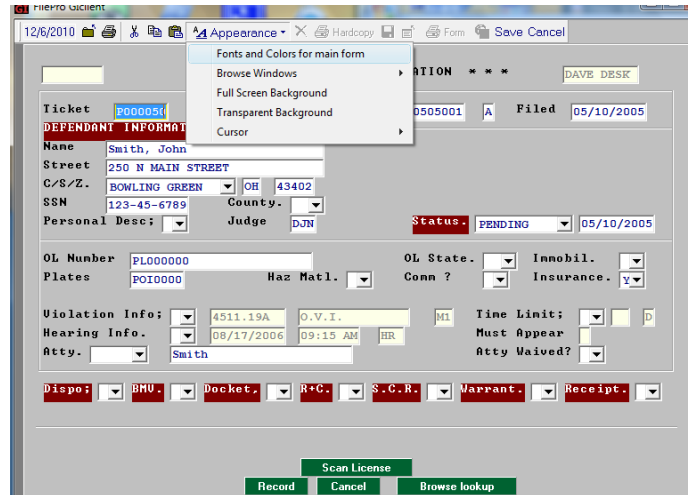


3rd color changing of screens is usually the most confusing to people. To change the colors here, you must first know the character screen color for the section you wish to change. To determine this, it's usually easiest to look at a screen login to a character client for a reference point. If you have no such access you can use the following steps to determine the screen actual screen color.

- 1) Login and go to a screen.
- 2) From the screen press Ctrl-B

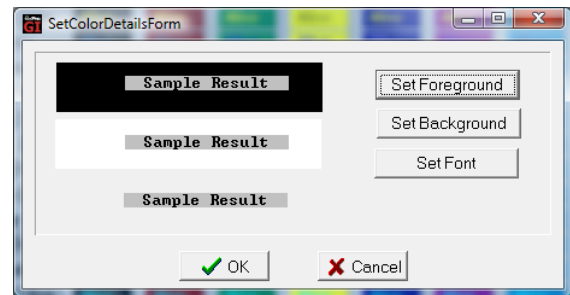


- 3) You will be prompted for a password the first attempt. Type in VLG at this prompt.
- 4) If successful, you will be presented with the underlying layout and colors of the screen you are on. Find the color combination you wish to change. In this example I could either change the highlighted sections (Red background hi-white) or the normal background (Blue background hi_white text)

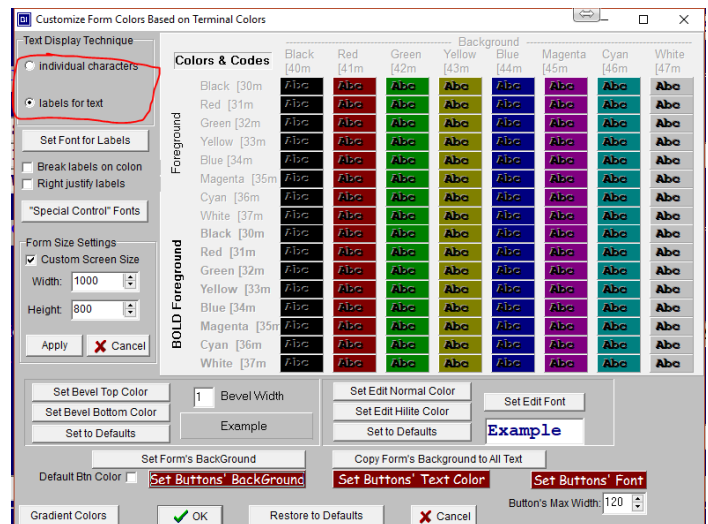


- 5) Once you have noted the associated colors of the screen area you wish to change, press Ctrl-B to return to the previous normal mode.

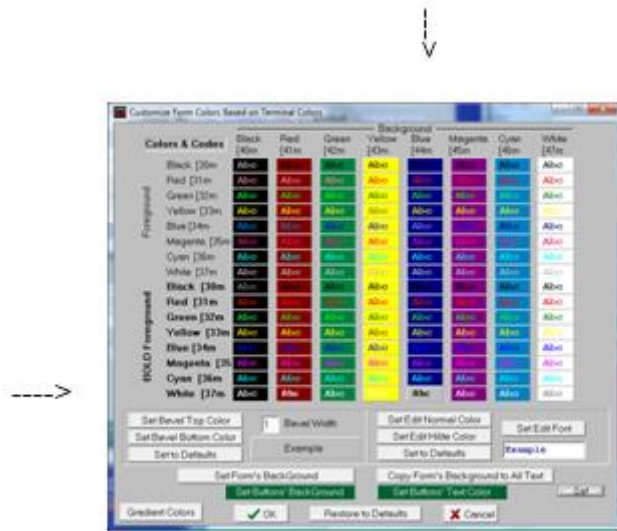
- 6) Now, go to the top of the screen and select appearance -> Fonts and Colors for main form



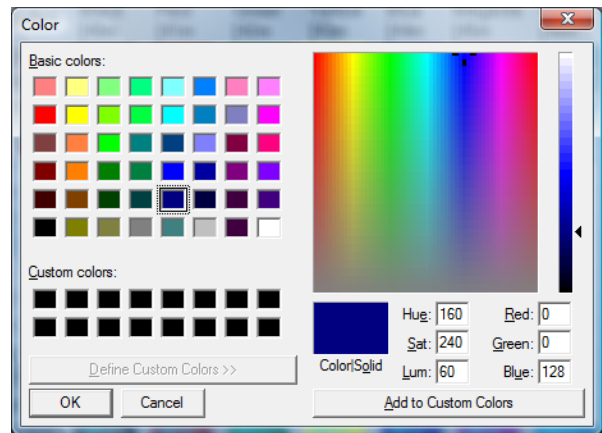
- 7) If you are using version 3 of the GIClient you have to check the radio button for individual Characters. Version 2 of the Client has the selectable grid available by default.



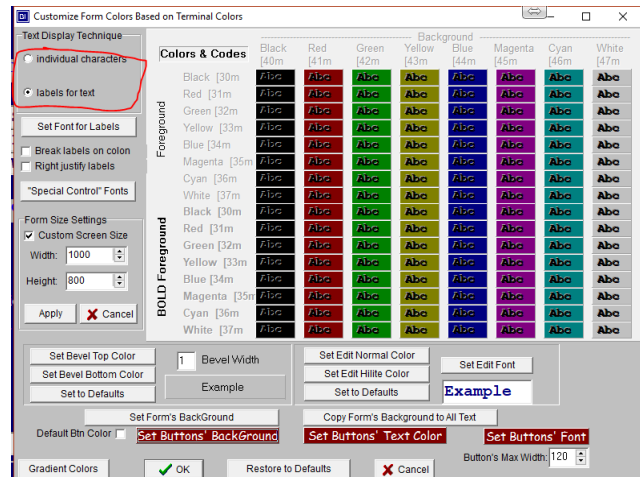
8) Here you will be presented with a grid of the color associations. Using the colors from the character screen you wish to map follow the background and foreground to where they intersect. In the example above the background though the GI interface shows gray the underlying color is blue background with hi_white text. Click on this intersecting button and you will be presented with a color choice window for the background and foreground color.



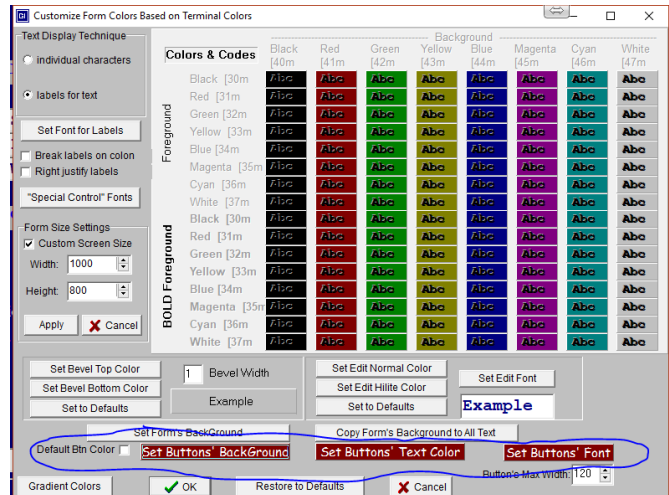
9) Now clicking on the foreground and background buttons will present you with a selection color palette to re-associate the underlying screens. The Foreground being the text color to the GI screen Click OK to approval of each of these color changes.



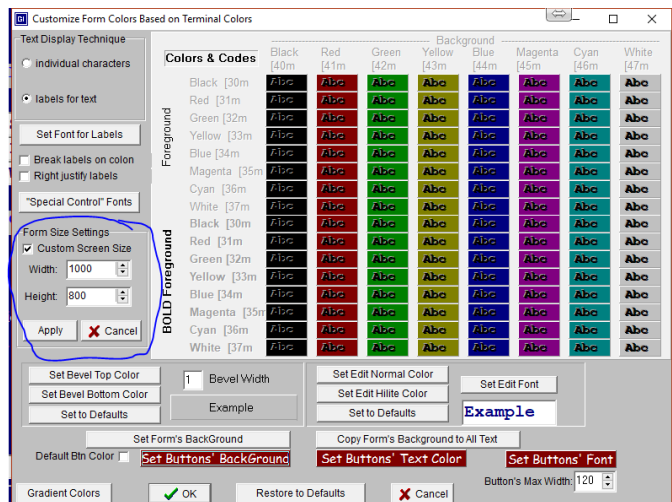
10) Before returning to the input screen check the labels for text radio button if you have V3. Otherwise the text will not scale properly.



11) If you wish to change the color of buttons, it's under the same appearance section of the client described above. However, you simply click the Set Button Background button and select the background color and Set button text color associates itself with as described to the button text. The third option allows one to change the font's text and size.



12) While here I should mention if you wish to change the starting login screen size this can be done with the width and height values in the "Form Size Settings" This has been found to be helpful on smaller screens like laptops.



From Experience

Now that we have gone over the general configuration and setup along with programming options, I'd like to speak about issues in general one might run into.

- 1) Though we know of no issues that should occur between the major browsers most of our testing has been done with Chrome. This was because of the volume of installs it has overall and second that Microsoft's plan is to use chromium source to replace their Edge browser.
- 2) It's not recommended to save your password in the browser's autofill option. Chrome's autofill has been reported to not respect autocomplete="off" as many have expressed frustration. Doing so can cause the data to be passed to other fields throughout the databases edit fields.
- 3) If you wish to not have the normal navigation menu at the top of the browsers it's possible to disable this using the following structure so it would look more like the image on the below. URL line disappears and shortcut could take you directly to site. Change the target to something like:



"C:\Program Files

(x86)\Google\Chrome\Application\chrome.exe" --enable-automation --disable-infobars --app=http://192.168.2.1:8080/fptech/

- 4) If you need to report an issue please see if the same issue occurs using the GIclients as you found in fileProWeb. Note, because of communication issues, OS issues aren't always captured for your review. This includes items that are looking for prompt of OS scripts. There's no notation of an terminal emulator and output from such scripts have been know to in some cases lock the fileProWeb client. Note the logs below rotate so it's best to get the logs sooner than later to have enough information.
 - a. Web server log - ~/fileproWeb/logs/webfast/webfast.log
 - b. GIServer logs on a account level logging- ~/fp/giserver/logs/{login}.cfg
 - c. fileProGI client log if not using fileProWeb. C:\fileProGI\disconnect.###.###.dat and fileProGI.###.###.LOG
 - d. Check your server's root email if on unix.
 - e. FilePro character logs if ~/fp_v6.0.01.07D6/logs/servlog.cfg is enabled ~/fp/logs
- 5) Reminder, if using one of the embedded functions, clear it before providing another value or you may end up with a warning that you exceed limits.
- 6) Poppups, make sure to clear with the clearp/clears commands. The client is keeping the states not the server as such it need to clearly know something is done or the screen may remain in unwanted locations.

- 7) Suggest one keeps the /appl/fpoffice linked to the documentRoot folder for temp files for future tools development.

Throughout this document there's been references to Operating Systems and other products. Each of these product and their company are covered by their trademarks. Including fP Technologies of Ohio, Inc product discussed in depth here within these pages. If you have additional questions about the filePro product line I'd suggest checking out their site at www.fptech.com or contacting their support. Also, another great resource is the beginners tutorial videos on their site http://fptech.com/fptech/Tutorials/beg_tutor.php for more advance knowledge they have their intermediate and Advance videos for purchase. These cover much of the filePro character interface to get someone started. In the future as time permits additional web and GI videos will be added.