Version 5.2 Update Highlights

Version 5.2.5 Update

- extended the Lasso `[string]` data type to include the following member tags (improved version of similar `fwpStr` tags that have existed for some time):
  - `->getLeft` : gets the leftmost (n) characters of a string
  - `->getRight` : gets the rightmost (n) characters of a string
  - `->getWords` : gets the first (n) words of a string
  - `->getSentences` : gets the first (n) sentences of a string based on splitting at `'. '`
  - `->getParagraphs` : gets the first (n) paragraphs of a string with paragraph detection including double `br` tags, `p` tags, and line feeds.
- extended the Lasso `[integer]` data type to include the following member tags which return a date string based on an integer. Use these like `$sessionExpires = 30->minutesFromNow`
  - `->secondsFromNow`
  - `->minutesFromNow`
  - `->hoursFromNow`
  - `->daysFromNow`
  - `->weeksFromNow`
  - `->monthsFromNow`
  - `->yearsFromNow`
  - `->secondsAgo`
  - `->minutesAgo`
  - `->hoursAgo`
  - `->daysAgo`
  - `->weeksAgo`
  - `->monthsAgo`
  - `->yearsAgo`
- extended the Lasso `[integer]` data type to include about 300 unit conversions for units of data size (bytes), length/distance, weight, fluid volume, and area. Units include American standard and Metric. Look at file fwpExtn_integer.lgc for details of each available conversion.
- much improved support for custom configurations of template `<head>` `meta` tags, `script` tags for external JavaScript libraries, and `link` tags for external CSS files through a new `$fw_headContent` object with the following member tags:
  - `->addMetaTag` : adds a pair to define the name and content of a meta tag
  - `->addHttpEquiv` : adds a pair to define the http-equiv and content of a meta tag
  - `->addCssFile` : adds a full path name to an external CSS file
  - `->addScriptFile` : adds a full path name to an external JavaScript file
  - `->removeMetaTag, ->removeHttpEquiv, ->removeCssFile, ->removeScriptFile` : corresponding remove methods can be used to modify a baseline of adds for a specific module or page

- eliminated the vars `fw_HTTPexpires`, `fw_HTTPauthor`, `fw_HTTPdesc`, and `fw_HTTPkeyWords` in favor of using the new `$fw_headContent` object.
- _defineCSS.lgc and _defineJavaScript.lgc continue to be options for creating customized logic, but both are disabled by default now.
- eliminated already obsolete `$fw_gUserPermsVars` from _initMasters
- eliminated already obsolete `$fw_pgRepeatLogic`, `$fw_pgRepeatAbove`, and `$fw_pgRepeatBelow` from _initMasters
- fixed typo for Spotlight connector in _fwpAPI_init
- bug fix to `fwpPage_loadTemplate` that was not including _definePageHead.lgc
- fixed a bug in fwpGUI_listRcrds.ctag that could cause lists draw without data
- updated some files in /siteUsers/ to use input names rather than field names in the queries and list config files (not a critical functional change, just a change to example code)

Migrating to 5.2.5

If you have a project started with an earlier release of 5.2, there's some things to rearrange and update for 5.2.5 that warrant some step-by-step details.

Update _initMasters

Generally speaking look for the lines with the //comments and variable declarations for the following variables and delete these lines:

- `fw_HTTPexpires`, `fw_HTTPauthor`, `fw_HTTPdesc`, and `fw_HTTPkeyWords`
- `fw_pgRepeatLogic`, `fw_pgRepeatAbove`, and `fw_pgRepeatBelow`
- `fw_gUserPermsVars`

In section (H), eliminate some of the now obsolete comment lines by modifing the top of the help comments to look like the following:

```
//  $fw_headContent    see docs
//  $fw_pageModes      see docs

//  $fw_client
//  ->decimalChar      ('.') default character for decimal in fwpNum and fwpMath tags
..... etc.....
```

Still in section (H), after the line:

```
//  all these vars can be overidden in _pageConfig
```

insert the following default lines above the `$fw_pageMode` lines:

```
//  $fw_headContent->(addHttpEquiv:   (pair:'expires'='0'));

    $fw_headContent->(addMetaTag:     (pair:'author'=''));
    $fw_headContent->(addMetaTag:     (pair:'description'=''));
```

```
$fw_headContent->(addMetaTag:     (pair:'keywords'=''));

$fw_headContent->(addCssFile:     ($fw_sPath->'css') + 'pb_apiStyles.css');
$fw_headContent->(addCssFile:     ($fw_sPath->'css') + 'styles_main.css');

$fw_headContent->(addScriptFile: ($fw_sPath->'js') + 'scripts_main.js');
```

Assuming you have definitions for those meta tags, transfer the content from the now obsolete vars to the declarations above.

Compare your modifications to the _initMaster.lgc file in the /pbStart/ project folder.

Update /_admin/ Files

If your application was using the styles_admin.css file to modify how the admin sections looked, you'll need to add the following `->addCssFile` line in _pageConfig for the affected modules such as /siteUsers/ and /sitestrings/:

```
if: ($fw_requestPage->'path') >> '_admin';
    $fw_headContent->(addCssFile: ($fw_sPath->'css') + 'styles_admin.css');
```

Eliminate Obsoleted Vars

Depending on which versions of the demo files your app may have initially been created with, there may be instances of some of the now obsoleted vars, so do a global search for these, and eliminate them.

- `fw_HTTPexpires`, `fw_HTTPauthor`, `fw_HTTPdesc`, and `fw_HTTPkeyWords`
- `fw_gUserPermsVars`

Version 5.2.4 Update

- added /siteMngr, /siteUsers and other admin features to pbStart file set.
- updated login procedure in mngr_login2_main.lgc to test for errors from `->authenticate` before trying the `->authorize` step. This will help to accurately report login lockouts and other specific login errors.
- updated error message 5016 in strings_coreErrors_en-us.cnfg
- bug fix for `fwp_recordData->delete` that required an -inputs parameter (which served no purpose)
- added `-formSpec` option to fwpErr_validator.ctyp to make it easier to use the validator on raw POST and GET data. See documentation section *Using fwp_validator on POST and GET*
- bug fixes to fwpEDP_controller.ctyp
- fixed sorting bug introduced to fwpGui_listRcrds.ctyp by a recent code update
- some minor updates to pb_apiStyles.css
- fixed a case where the `fwpLog_err` tag was bing invoked even if `$fw_gLogErr` was set to false
- added trace calls in log tags
- added a trace call to the `->insert` action of `$fw_error` so that we can see exactly when errors occur in the trace

Version 5.2.2/5.2.3 Update

Some minor fixes were release with 5.2.1, but with 5.2.2 there's a a major update to how value lists are created and used. There's a whole new system with new capabillities documented in detail in this updated Guide. The previous value list tags are deprecated but still available with the documentation for those moved to Appanedix A. There's no urgent need to rewrite existing value list code in applications. The old tags will likely remain available for some time. However, using the new value list data type will help improve the logic/display separation of your code.

Additonally, there are new fwpGIS tags for acquiring a U.S. address geocode from the Google API and from the Yahoo API. (5.2.3 added -minPrecision to these tags, and fixed a bug).

Added `fwpNum_dec5` and `fwpNum_dec6` tags.

Version 5.2.x Overview

- New—integrated database migrations management
- New—integrated deployment mode detection
- New—Nemoy search engine (Winner of first Lasso Programming Challenge)
- Enhanced feedback for misconfiguration and developer errors
- Enhanced query abstraction system
- Enhanced API startup process for better 3rd party integration

- Integrated template head and page wrapup tasks into templateLoader to prevent framework updates from overwriting developer code
- Separated all default definitions for CSS styles embedded in framework HTML generators into a separate CSS file for easy modification or overriding
- Several minor fixes, improvements, and a few new tags

- The init process of the API files that go into /LassoStartup/ has been updated. This won't affect any application code, but allows API sets to be initialized individually, included or excluded, and allows application code to test for dependencies of API libraries. See the documentation section *API Startup and Dependencies*
- The page assembly process has been updated to simplify the developer's code, and to help isolate API updates from overwriting developer code in certain areas. In previous versions, the files in /_pbLibs/ were mixtures of API code and developer code. The intent was to allow the developer to modify these core processes. The default processes have now been fully integrated into the template loader, and files in this folder are now either additive or alternative to the built-in processes. This means framework updates no longer require the developer to scour through these files looking for update modifications.
- New multi-deployment management enables resource and behavior modification on-the-fly based on the domain name the site runs as. This allows the use of separate databases when running the site under production, test, development, etc. For example, when the site runs as www.example.com, it will use the database mydb_prod, but when run locally as www.example.dev it will automatically use mydb_test. Declare as many cases as needed to cover production, beta, test, development, etc.
- Added numerous error handlers to catch common misconfiguration and errors in setup and use of PageBlocks tags and types. Prior to these, it was common for misconfigurations to result in obscure Lasso error messages. Now, more specific information is provided with each error, and the tagTracer now displays more error information to help identify exactly when in the page processing the error happened.
- A new `$fw_apiError` object is used to capture programming errors so that `$fw_error` can be focused on providing error feedback to end users. This allows messages to show alternative versions based on whether debugging is enabled or not.
- Added escalation values to `$fw_debug`. Before this change `$fw_debug` was boolean. It is now an integer represented by the constants `fw_kDisabled`, `fw_kEnabled`, `fw_kChatty`, `fw_kVerbose`. This allows debugging messages to be invoked under various degrees of detail. In particular, this is now used to add messages to the tagTracer so that only basic tag calls are logged when debugging is enabled at the base level. Detailed inner workings are logged when `$fw_debug` is set to `fw_kChatty` or `fw_kVerbose`.
- Added new `$fw_criticalLog` to enable certain critical error events to be written to the Lasso SiteAdmin Errors log. The new error handling includes `log_critical` statements when setup, database availability, and config file access errors are encountered.
- Updated all `fwpDate` tags to account for how Lasso 8.5.2 changed the way it treats date casting with empty or non-date-string inputs.

- The `fwp_recordData` custom type has been updated to allow the abstracted input names from tableModel to be used in query phrase fragments in places of real field names. This now allows the abstracted input names to be used universally in application code.
- The `fwp_recordData` and `fwp_rcrdList` custom types have improved options for GROUP BY and ORDER BY.
- Added new `fwp_recordLockStore` type which stores muliple record locking values. Prior to this update, only one record lock ID per session was retained, making it possible to lose track of some locks under certain error conditions. This update will prevent several scenarios that would require locks to expire before re-gaining access to those records. The `fwp_recordData` custom type has been updated to make use of this new lockStore.
- Fixed bugs in fwpActAdaptor_mysql that could prevent some date data types from being saved, and changed quoting of some query strings.
- The `fwpGUI_checkbox`, `fwpGUI_listBox`, `fwpGUI_radioBtn`, and `fwpGUI_popup` value list tags all have a new input to specify the html `id` attribute. Previously it was forced to be the saem as the `name` attribute. This update allows them to be different.
- `fwpUtil_showVars` has improved output formatting for HTTP headers
- `fwpStr_randomID` has new options to force use of uppercase letters and for inserting hyphens at specified intervals. This enables the generation of IDs like J7KL-TY-OPI8.
- Added new tag `fwpGIS_geocoderCoords`.
- Added `isAlphaNumericHyphen` and `isAlphaNumericSpace` validations to `$fw_validator`.
- Added `drawHiddenExtras` and `drawHomeFilterInputs` methods to `fwp_edpView`.
- Added `-makeMap` option to `fwpCnfg_splitPairs`
- Added new `fwpCnfg_splitBlocks` and `fwpCnfg_splitLabels` tags
- Added `forwardedFor` and `forwardedHost` instance vars to `fwp_client`.
- Added `$fw_client->ip` (as a member tag and instance var). Looks for `X-Forwarded-For` in the page headers and returns that value if present, else returns the normal `client_ip` value. In a load-balanced muli-server setup, `client_ip` is usually going to return the IP address of the load balancing machine that forwarded the request. The `X-Forwarded-For` value is often added to headers to reflect the original client IP address (just like you'd normally expect from `client_ip`). You should use `$fw_client->ip` everywhere that would normally use `client_ip` to make your applications ready for load-balanced installations. (All internal API code has been updated this way).
- In `fwpErr_validator`, changed `#thisMsgFirstWord` to `#msgFirstWord` to be consistent with the documentation
- In `fwpErr_validator`, changed `hasLabel` code to be processed if it starts with a `[` so MVS code can be used to display multi-language strings for field names
- In `fwpErr_validator`, update the `date|isDate` validation code to allow an `=euroDate` option to force inpout value to be interpreted as a dd/mm/yyyy date.
- The `fwp_user->testPswd` method has been updated to use the standard validation error strings system.

- In `fwpGui_menuVText.ctag` and `fwpGui_menuHText.ctag` are updated to process menu name fields from the config file if the menu item string starts with a `[` so MVS code can be used to display multi-language strings for menu titles
- adjusted the _atBegin routine to allow default.lasso and index.lasso to be used as "index" files. Either one can be used in any given folder, usage does not have to be one or the other.
- added `useAppStringsDataTables` in `$fw_pageModes` to control whether the appStrings system should look up a requested string in the appstrings data tables or not. This also elminates the version 5.1.x requirement of having the appStrings_en_us table present in a PageBlocks database whether it was needed or not.
- added some improvements to the Debug display output
- cleaned up the auto generated `<label>` html for checkboxes and radio buttons, and added `loop_count` to the auto generated `id` values.

Migrating to 5.2

Prior to replacing your /LassoStartup files, make a backup copy of /LassoStartup/_fwpAPI_init.lgc as you'll need to transfer any adjustments you've made in there to the new version. You may not remember making changes from the defaults, so make a copy and verify that, or you'll be blaming changes in app behavior on incompatibilities instead of misconfiguration.

The vast majority of changes introduced in 5.2 are backwards compatible with 5.1.x code, with some exceptions. First and foremost, all the tags, types, and variables that were deprecated in 5.1 have been eliminated. This is to reduce the extra clutter these created, and to eliminate the processing time they consumed. Refer to the 5.1 Migration Details in the section below, and update all deprecated items in your application code. The majority of these can be done with search and replace.

Variable Name Changes

The following variables (left column) have been replaced by the objects in the right column. These changes have been made with immediate effect. The old variables are not available. (Note that `fw_k` names are constants not variables).

```
$fw_gUrlParamsChar              = fw_kUrlParamsChar
$fw_gServerMode                 = $fw_serverMode
$fw_helpEmail                   = $fw_accountsHelpEmail
$fw_helpPhone                   = $fw_accountsHelpPhone
```

Filename Changes

```
_pageMapper.lgc                 = _pageRouter.lgc
_fwpAPI_initVars.lgc            = _fwpAPI_init.lgc (won't impact app code)
_fwpAPI_initCache.ctyp          = _fwpAPI_initcache.ctyp (won't impact app code)

fwpPage_setCSS.lgc              = _defineCSS.lgc (now optional)
fwpPage_jsScriptsLib.lgc        = _defineJavaScript.lgc (now optional)
fwpPage_templateHead.lgc        = _definePageHead.lgc (now optional)
fwpPage_wrapup.lgc              = _definePageWrapup.lgc (now optional)
```

Updating /_pbLibs/

fwpPage_templateHead.lgc

The file fwpPage_templateHead can usually just be deleted. If you have made modifications to this file to incorporate custom meta tags, unique special JavaScript loading, or other changes, then read through the new docs section on *Using /_pbLibs/* to see how the new options can be used.

If necessary, you can rename the old file to _definePageHead.lgc, and in /LassoStartup/_fwpAPI_init, make sure that the definition for the constant `fw_kUseDefinePageHead` is set to true. (Restart the Site if needed).

fwpPage_wrapup.lgc

The file fwpPage_wrapup can usually just be deleted. If you have made modifications to this file to customize the debug output `topVars` or `clearVars`, these lists are now defined in _initMasters. If you have modified what was section (B) to add vars to the `->addVars` list, you can do that in the new _defineWrapup.lgc file (the example file in the pbStart kit includes that as default code).

fwpPage_setCSS.lgc

Rename your existing file to _defineCSS.lgc, and in /LassoStartup/_fwpAPI_init, make sure that the definition for the constant `fw_kUseDefineCSS` is set to true. (Restart the Site if needed).

fwpPage_jsScriptsLib.lgc

Rename your existing file to _defineJavaScript.lgc, and in /LassoStartup/_fwpAPI_init, make sure that the definition for the constant `fw_kUseDefineJavaScript` is set to true. (Restart the Site if needed). Also read the new docs section *Integrating JavaScript Libraries* for more details on the topic as I think the use of this file has been largely misunderstood in the past.

Updating Master Templates

The new integrated templateLoader cause a few changes to master template files. The good news is that they get simpler.

Previous version templates all had the following first line of code which can now be removed completely:

```
<?lassoscript include: ($fw_sPath->'apiLibs') + 'fwpPage_templateHead.lgc'; ?>
```

Your templates files may also have the following code:

```
[if: $fw_construction || $fw_dataOffline || $fw_maintenance]
    [include: ($fw_sPath->'msthd') + 'alerts.dsp']
[/if]
```

which should be replaced by

```
[fwpPage_loadAlerts]
```

And, lastly, each template file should have ended with a </html> tag. That should be removed as the template loader adds it for you.

So, now, your templates should start and end with the <body> </body> tags.

Updating _initMasters.lgc

New Debug Modes

In Section (E), `$fw_debug` is now used a different way. Prior to 5.2, it was a simple boolean value. This has been replaced by using one of four constants as a value. I find it handy to have all four values assigned like this:

```
    $fw_resetAllCaches      = false;
    $fw_debug               = fw_kVerbose;
    $fw_debug               = fw_kChatty;
    $fw_debug               = fw_kEnabled;
    $fw_debug               = fw_kDisabled;
    $fw_debugLog            = false;
    $fw_debugTimers         = false;
    $fw_debugIPFilter       = '127.0.0.1';
```

Note that all four values are listed. Of course, when run like this, each subsequent assignment will override the former one. So, listed like this, the net result is that `$fw_debug` is disabled. To enabled the chatty mode, I comment out the lower two options, and end up with this.

```
    $fw_resetAllCaches      = false;
    $fw_debug               = fw_kVerbose;
    $fw_debug               = fw_kChatty;
//  $fw_debug               = fw_kEnabled;
//  $fw_debug               = fw_kDisabled;
    $fw_debugLog            = false;
    $fw_debugTimers         = false;
    $fw_debugIPFilter       = '127.0.0.1';
```

The last assignment is now the new value. If this bothers you, then you can certainly use a single line assignment and change the value as needed.

Add the following new variables just below the debug vars section. This is now where you manage the output of vars in the debug display.

```
$fw_debugTopVars        = '';
$fw_debugClearVars      = 'fw_gQueryUser, fw_gQueryPswd, fw_gFilesUser, fw_gFilesPswd, ' +
                          'fw_gUploadUser, fw_gUploadPswd, fw_gPassthruUser, ' +
                          ' fw_gPassthruPswd, fw_gDatabases, fw_gTables, ' +
                          ' fw_gDbTableModels, fw_gHexMap, fw_rollovers';
```

Variable changes

In section (F), cut this declaration:

```
$fw_gServerMode = 'dev';
```

and paste it to section (H) just above `$fw_uploadMIMEs`. Change the name from `fw_gServerMode` to `fw_serverMode` so you have this in section (H) (your value assignments may be different):

```
//  $fw_client->(setVariant:'default');

    $fw_serverMode        = 'http';

    $fw_uploadMIMEs       = array;
    $fw_uploadSizeMax     = 0;
```

In section (H) remove these lines. These are now controlled by application-wide constants in /LassoStartup/_fwpAPI_init.lgc.

```
$fw_pageModes->enableSetCSS;
$fw_pageModes->enableJScripts;
```

New Deployment Management

In Section (E) add the following variables with domain values that make sense for your application. See the document section for this new feature for a detailed explanation.

```
$fw_deploymentHosts->(insert: 'pageblocks.org'   = 'production');
$fw_deploymentHosts->(insert: 'pbtest.dev'       = 'test');
$fw_deploymentHosts->(insert: 'pb.dev'           = 'dev');

$fw_deploymentMode = $fw_deploymentHosts->(find: ($fw_requestPage->'domain'));
```

In conjunction with this change, you'll now want to update the database declarations in section (K) of the _initMasters. Create a `select/case` structure similar to this:

```
select: $fw_deploymentMode;
    case:'production';
        $fw_gDatabases = (map:
            'auth'  = 'pbdemo_production',
            'logs'  = 'pbdemo_production',
            'site'  = 'pbdemo_production');
    case:'test';
        $fw_gDatabases = (map:
            'auth'  = 'pbdemo_test',
            'logs'  = 'pbdemo_test',
            'site'  = 'pbdemo_test');
    case;
        $fw_gDatabases = (map:
            'auth'  = 'pbdemo_dev',
            'logs'  = 'pbdemo_dev',
            'site'  = 'pbdemo_dev');
/select;
```

For this example, the use of production, test, and development (dev) deployments have been created. Based on the domain name the site is running under, the application will use the corresponding database.

These names of production, test, and dev are not mandatory. By declaring domains and labels in `$fw_deploymentHosts` you can name deployments any way you want. Perhaps you want one named beta or demo.

If the sub-domain name needs to be the determining factor, then change this line

```
from:   $fw_deploymentMode = $fw_deploymentHosts->(find: ($fw_requestPage->'domain'));
to:     $fw_deploymentMode = $fw_deploymentHosts->(find: ($fw_requestPage->'subdomain'));
```

and that will allow demo.example.com to run a separate database than www.example.com.

Reorganization of strings_coreErrors_en-us.cnfg

As part of the improved diferentiation between errors intended for developers vs end users, it was necessary to reorganize the internals of the strings_coreErrors_en-us.cnfg file. Error codes that were intended for end users have generally remained the same, but many of the internal errors designed to inform the developer have changed codes and text.

If you have developed, or are using, any alternative language versions of the coreErrors file, you will need to update those files to be congruent to the new organization and text.

New Tags/Types

Database Tags/Types

- fwpActn_encodeLIKE.ctag
- fwpActn_lockStore.ctyp

Config File Tags

- fwpCnfg_splitBlocks.ctag
- fwpCnfg_splitLabels.ctag

Formatting Tags

- fwpDate_isoDateTime
- fwpDate_isValid
- fwpStr_markWebLinks

Nemoy Search Tools

- fwpSrch_nemoyController.ctyp
- fwpSrch_nemoyCriteria.ctyp
- fwpSrch_nemoyQuery.ctyp
- fwpSrch_nemoyAdaptorMySQL.ctyp
- fwpSrch_nemoyAdaptorSqlite.ctyp

- fwpSrch_nreMysqlmod.ctyp
- fwpSrch_nreRobertson.ctyp
- fwpSrch_nreWordcount.ctyp

New Tag Param Names

Several tags have updated parameter names. PageBlocks 5.0 often used a boolean value as a technique to create on/off type parameters. This has been changing with each release to simply having the param present or not. Additionally, the names of many params have been changed to improve readability. In these cases the old param names continue to be supported for backwards compatibility.

One area with several changes is the `fwpCnfg` tags. Several tags supported a `-nosplit` option which has been changed to `-withoutSplit`. For example,

```
fwpCnfg_splitComma: #theseInputs, -withoutSplit;
```

has improved readability over

```
fwpCnfg_splitComma: #theseInputs, -nosplit;
```

While the meaning is certainly not obscure in either case, the former represents the style of readability that PageBlocks APIs are evolving towards.

Consult the details of each tag in the docs or reference database, but the following are some changes that affect several of the config tags:

```
-nosplit       -withoutSplit
-nocache       -withoutCache
-noTrim        -withoutTrim
-nomerge       -withoutMerging
-despace       -removeWhiteSpace or -removeExtraTabs
```

It won't be necessary to change these params as the old names are still supported, but as you work on sections of code, it may be worth updating them to minimize the impact of future deprecation.

New Page Not Found Handler

The files /_apiLibs/fwpPage_errMngr.lgc and /_apiLibs/fwpPage_pageNotFound.lgc are both obsoleted, and no longer included as of 5.2.0. This requires some minor modification to the file template_filenotfound.dsp, and to all _pageConfig.lgc files.

Updating template_filenotfound.dsp

The demo files included a section of code like this:

```
<div id="pgblock2colwnmain">
    [include:($fw_sPath->'apiLibs') + 'fwpPage_pageNotFound.lgc']
    [include: ($fw_sPath->'apiLibs') + 'fwpErr_mngr.lgc']
    [($fw_error->'errorMsgs')]
</div>
```

The div has no importance, but the three Lasso lines inside it do. These should be changed to the following (only the first two Lasso lines are different):

```
<div id="pgblock2colwnmain">
    [fwpPage_pageNotFound]
    [$fw_error->handleAllErrors]
    [($fw_error->'errorMsgs')]
</div>
```

Updating _pageConfig.lgc

The select/case statements that identified each page will almost always end like this:

```
case;
    include: ($fw_sPath->'apiLibs') + 'fwpPage_pageNotFound.lgc';
```

And, the admin sections would have looked like this:

```
case;
    var:'fw_pgAuthRequired'    = false;
    include: ($fw_sPath->'apiLibs') + 'fwpPage_pageNotFound.lgc';
```

Both of these can be replaced with the following:

```
case:
    fwpPage_pageNotFound;
```

The `fwpPage_pageNotFound` tag takes an optional `-pgTitle` parameter to define the string that will show as the page title in the browser window title bar. It is “Page Not Found” by default.