

Open-Source PowerBuilder Foundation Classes

Bruce Armstrong, Integrated Data Services
December 2-3, 2024



DISCLAIMER

This presentation was authored by volunteer(s) in the Appeon community. This is not a work for hire by Appeon. The views and opinions expressed in this presentation are those of the author(s).

Its contents are protected by US copyright law and may not be reproduced, distributed, transmitted, displayed, published or broadcast without the prior written permission of Appeon. All rights belong to their respective owners.

Any reference to third-party materials, including but not limited to Websites, content, services, or software, has not been reviewed or endorsed by Appeon. YOUR USE OF THIRD-PARTY MATERIALS SHALL BE AT YOUR OWN RISK.

Appeon makes no warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. Appeon assumes no responsibility for errors or omissions.

Session Agenda

- Why use the Open-Source PowerBuilder Foundation Class (PFC) libraries?
- Why upgrade to the new PFC?
- How to upgrade to the new PFC?
- How to migrate an existing application to use PFC?
- How to contribute to PFC?

Presenter Profile



Bruce Armstrong



<https://www.linkedin.com/in/brucearmstrong/>

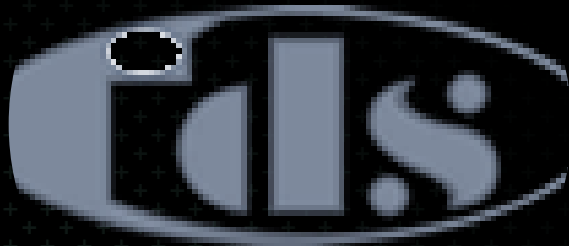
Recent Projects

- [OpenSourcePFCLibraries](#) - The open-source PowerBuilder Foundation Class libraries

Key Skills

- PowerBuilder
- Oracle PL/SQL
- Java
- C#
- C++

Company Profile



Integrated Data Services

IDS has delivered innovative program and financial management information system solutions to the Federal Government for nearly three decades. Our flagship Enterprise Requirements Management product, CCaR™ is used by the DoD and Federal Agencies including the Air Force, Space Force, Army, USSOCOM, DSCA, DHA, JSF and DOE.

Why use the PFC framework

History

- Originally developed by Sybase and released with PowerBuilder 5.0 in 1995.
- Released by Sybase as an open-source project with the release of PowerBuilder 10 in 2005.
- Originally maintained on Sybase CodeXchange
- Moved to Microsoft CodePlex in 2014 when SAP/Sybase abandoned CodeXchange
- Moved to GitHub in 2017 when Microsoft abandoned CodePlex
- Currently located at <https://github.com/OpenSourcePFCLibraries>

Why use the PFC framework

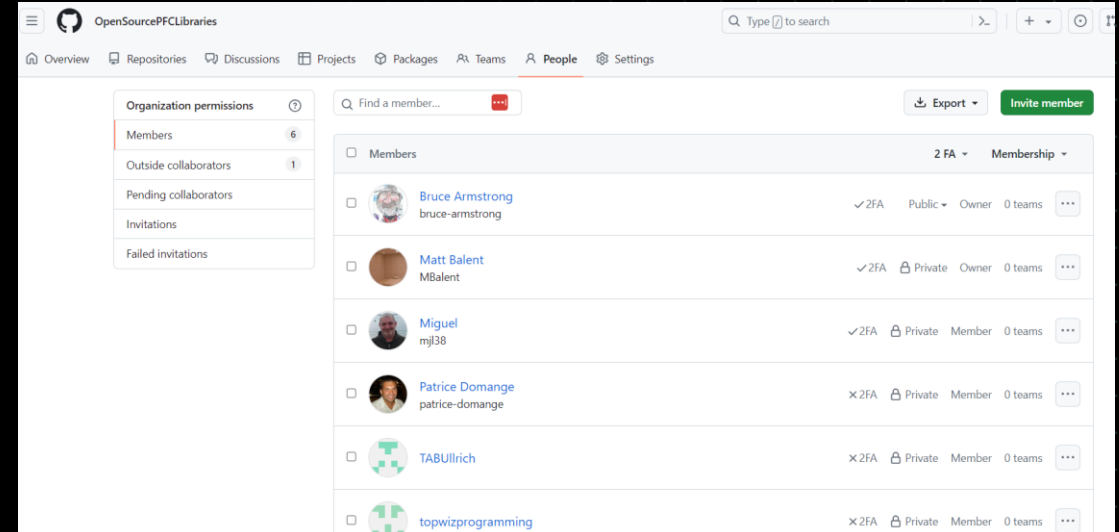
Advantages of open-source software¹

- Agility
 - You don't have to wait for the vendor to develop capability, you can add it yourself
- Speed
 - You don't have to develop the base infrastructure for your project, you build on the infrastructure already provided
- Cost-effective
 - You don't have to pay your developers to develop the base infrastructure, only for what you build on top of it
- Lower learning curve for new hires
 - They are likely to have used the framework in prior projects
- Share maintenance costs
 - You can report issues you've encountered and members of the open-source community will often provide a fix

¹ Adapted from <https://enterpriseproject.com/article/2015/1/top-advantages-open-source-offers-over-proprietary-solutions>

Why use the PFC framework

- Supported by a community of developers
 - One of the members of the community leaves the project it is less likely to impact the longevity of the project
- Open-source projects that are maintained by only one developer are at risk of collapse of that developer stops supporting the project



Why use the PFC framework

- Permissive licensing
 - The open-source PFC libraries are licensed under one of the most permissive open-source licenses available: The MIT License.
 - Other so called open-source libraries may provide a permissive license that they indicate they are licensed under, but examination of the copyright notices in the source code itself may indicate otherwise.

```
//  
////////////////////////////////////  
//  
/*  
 * Open Source PowerBuilder Foundation Class Libraries  
 *  
 * Copyright (c) 2004-2017, All rights reserved.  
 *  
 * Redistribution and use in source and binary forms, with or without  
 * modification, are permitted in accordance with the MIT License  
 *  
 * https://opensource.org/licenses/MIT  
 *  
 * =====  
 *
```

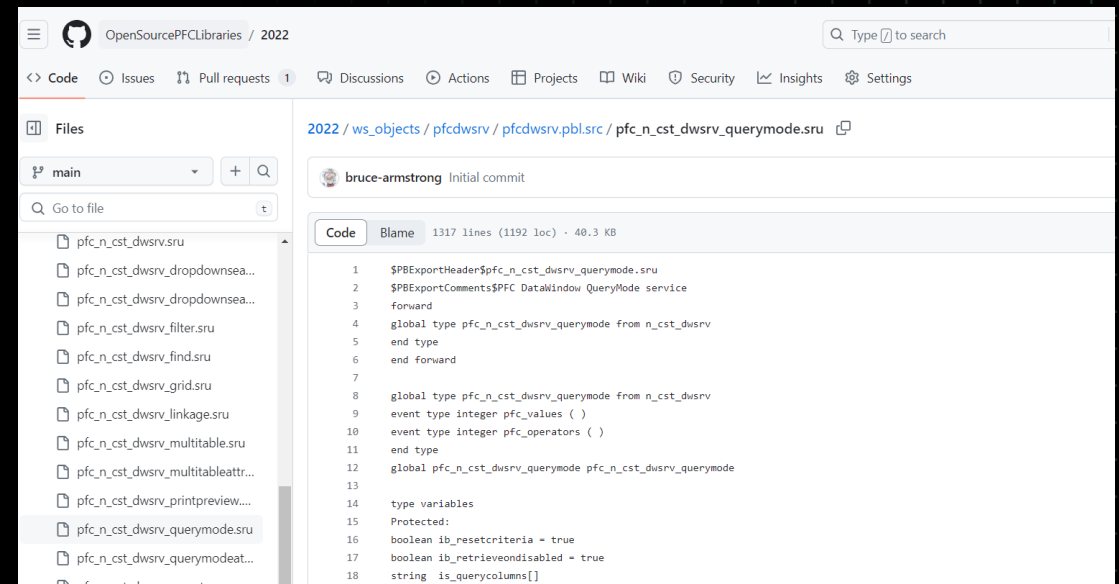
Return Value : long

```
----- Copyright -----  
Copyright © 2020 by Software Tool & Die Inc. here in known as STD Inc. All rights reserved.  
Any distribution of the STD Foundation Classes (STD_FCL for PowerBuilder®, InfoMaker®,  
PowerServer Web® or PowerServer Mobile® source code by other than by STD Inc. is prohibited.  
----- Revisions -----  
1.0 Initial Version - 2020-08-02  
*/
```

// Declarations

Why use the PFC framework

- The source code can be read directly on the internet
 - Such transparency helps ensure that any contributions to the project are not somebody else's intellectual property



The screenshot displays a GitHub repository page for 'OpenSourcePFClibraries / 2022'. The file 'pfc_n_cst_dwsrv_querymode.sru' is selected, showing its initial commit by 'bruce-armstrong'. The code is written in a query language and includes the following content:

```
1 $PBExportHeader$pfc_n_cst_dwsrv_querymode.sru
2 $PBExportComments$PFC DataWindow QueryMode service
3 forward
4 global type pfc_n_cst_dwsrv_querymode from n_cst_dwsrv
5 end type
6 end forward
7
8 global type pfc_n_cst_dwsrv_querymode from n_cst_dwsrv
9 event type integer pfc_values ( )
10 event type integer pfc_operators ( )
11 end type
12 global pfc_n_cst_dwsrv_querymode pfc_n_cst_dwsrv_querymode
13
14 type variables
15 Protected:
16 boolean ib_resetcriteria = true
17 boolean ib_retrieveondisabled = true
18 string is_querycolumns[]
```

Why upgrade to the new PFC

- Sample databases now provided in SQL Anywhere 17 and SQL Server 2022 exports
- Application code has been updated to ensure it works with both of those databases
- Removed obsolete objects intended to support AIX, HP-UX, Solaris and 32 bit windows
- Added PFC wrapper objects for new PowerBuilder objects and controls such as CoderObject, CompressorObject, CryptorObject, ExtractorObject, HttpClient, RestClient, Animation control, Datepicker control, MonthCalendar control, RibbonBar control and WebBrowser control

Why upgrade to the new PFC

- n_cst_apppreference – added option to allow application configuration to be stored in INI file, XML file or registry
- n_cst_color – added 50+ additional color constants based on PowerBuilder values
- n_cst_debug – extended to allow DataWindow data to be viewed as JSON for debugging
- n_cst_colorconv – new object which performs conversion of color values between rgb, xya, yxy, hunterlab, cielab, deg2rad, hsl, hal, hsv, rad, hue, cmy and cielch
- n_cst_conversion – extended to handle 240 more PowerBuilder types

Why upgrade to the new PFC

- `n_cst_environment` – new object to return information about host system
- `n_cst_filesrv` – extended with methods to create directory trees, get system directory, get short path name, get long path name, get windows directory, move files, get temp file path and get temp file name
- `n_cst_inifile` – extended with method to create new empty INI file
- `n_cst_json` – new object for handling JSON text and files (largely obsolete now because of PowerBuilder native `JSONParser`, `JSONPackage` and `JSONGenerator` objects)

Why upgrade to the new PFC

- `n_cst_measureconversion` – new object to do conversions between metric to English, metric to metric and English to English for distance, weight, temperature, pressure and force
- `n_cst_metadata` – extended to add methods to return initial values, scripts, functions, events, variables, arguments, etc.
- `n_cst_numerical` – extended to add methods to do bitwise operations and handle hex, octal and binary numbers
- `n_cst_pbunitconversion` – new object to handle conversions between inches, twips, pbunits, pixels and centimeters

Why upgrade to the new PFC

- `n_cst_platform` – extended with methods to get system folder paths, system colors, dpi, system metrics, free memory and physical memory
- `n_cst_string` – extended with methods to determine if an email address is well formatted, perform a global search and replace, remove punctuation or other special characters and to URL encode a string
- `n_cst_resize` – extended with options to position controls in center, center top, center bottom, center left and center right
- `n_cst_winsrv_style` – new object to provide extended styling capability for windows
- `n_ds` – extended by adding `of_printsetup` function

Why upgrade to the new PFC

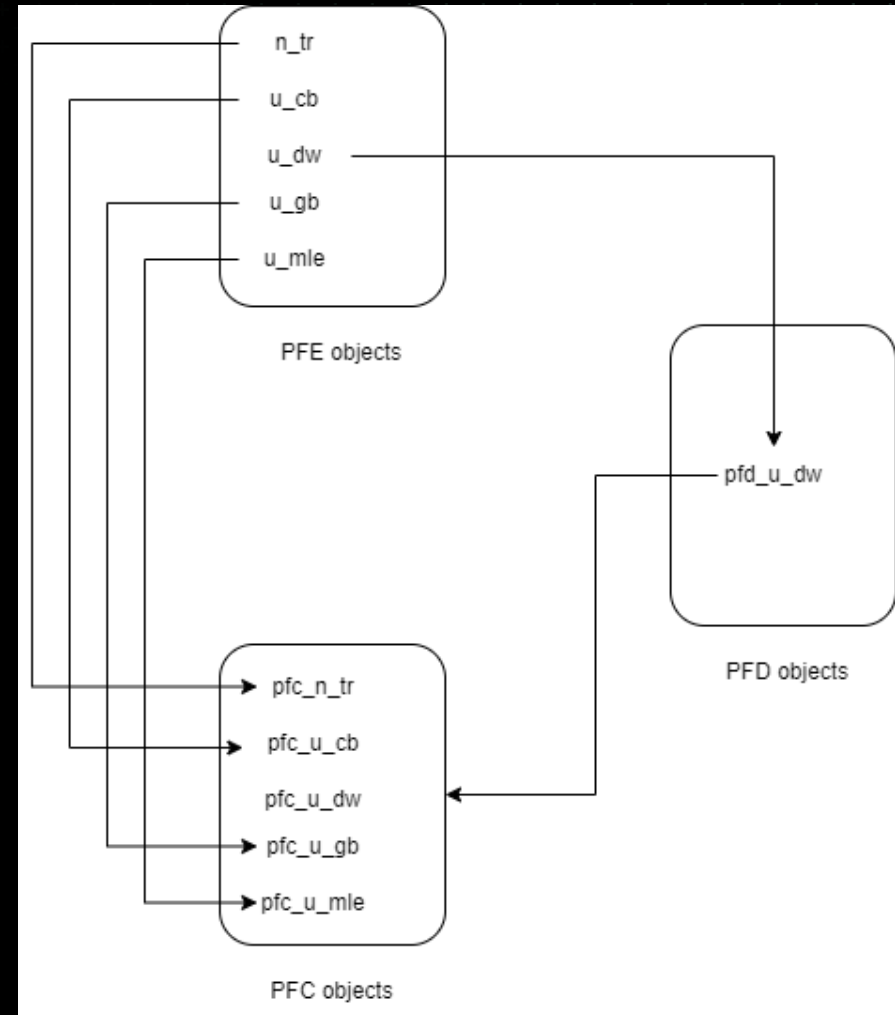
- n_st_splitbar – enhanced to handle new PowerBuilder control types and improve movement handling
- u_htb – extended by adding pfc_positionchanged event that can be scripted to respond to changes in position of the track bar
- u_vtb – extended by adding pfc_positionchanged event that can be scripted to respond to changes in position of the track bar
- w_master – extended by adding pfc_preprint and pfc_postprint events
- w_response – extended by adding user resize capability
- Numerous bug fixes

Why upgrade to the new PFC

- Database files, images, help files moved into separate repositories
- Help files recompiled in CHM and PDF
 - ShowHelp calls in PFC project updated to open CHM files.
- Visual Expert run on the project and the majority of “Critical” and “Major” issues addressed

How to upgrade to the new PFC

- PFC layer objects have all the main source code
 - Your application should not refer to these object and you should not edit them
- PFE (extension) layer objects inherit from PFC layer objects but have no source code.
 - These are the objects that should be referenced in your application code.
- PFD layer objects are created as needed to extend PFC objects
 - This is where you add your framework (not application specific) customizations



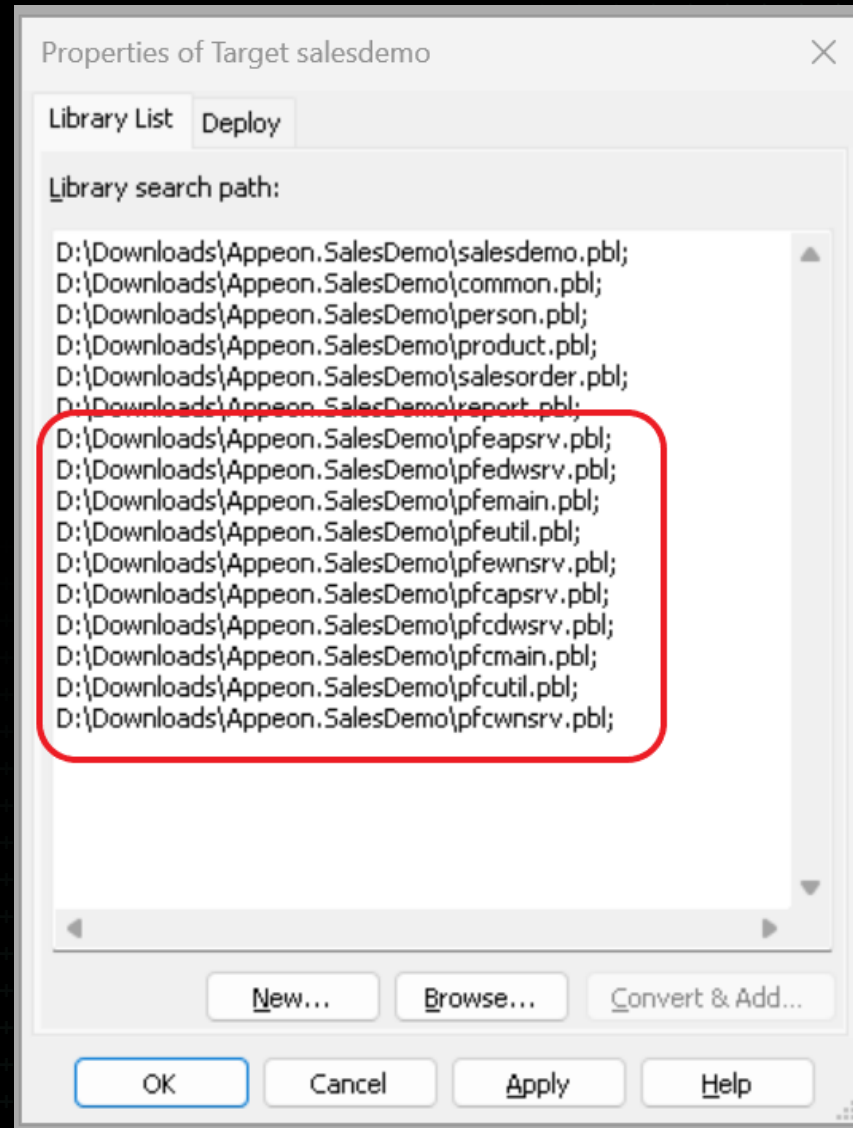
How to upgrade to the new PFC

- If you created a PBD layer
 - Replace the PFC and PFE layers with the current version
 - Modify the objects in the PFE layer for which there are ancestors in PFD
- If you edited the PFE layer
 - Replace the PFC layer with the current version
 - Compare your PFE layer with the current version using a tool such as [pblcompare](#) for added/removed/modified objects
- If you edited the PFC layer
 - Use a utility such as [pblcompare](#) to compare the PFC layers and apply changes
 - Consider making such changes in the PFD or PFE layer to facilitate later updates from the main Open-Source PFC project

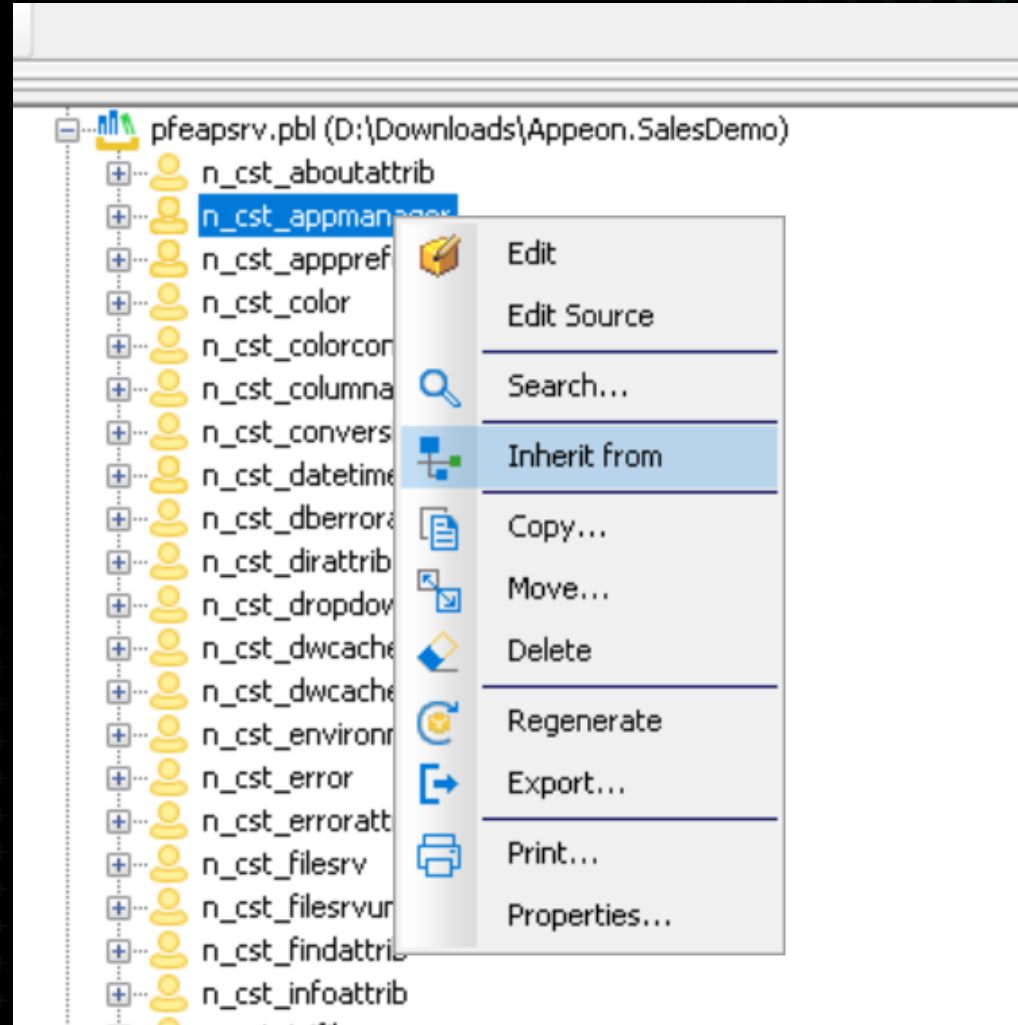
How to migrate an existing application to use PFC?

- Add PFC libraries to the application library list
- Inherit an application specific application manager from the `n_cst_appmanager` object in PFC
- Declare the application specific application manager as a application global variable
- Update the application variable types to use PFC classes
- Deal with any object name collisions
- Update window and user object ancestries
- Modify application object events
- Code the application specific application manager constructor and `pfc_open` events
- Start using PFC services

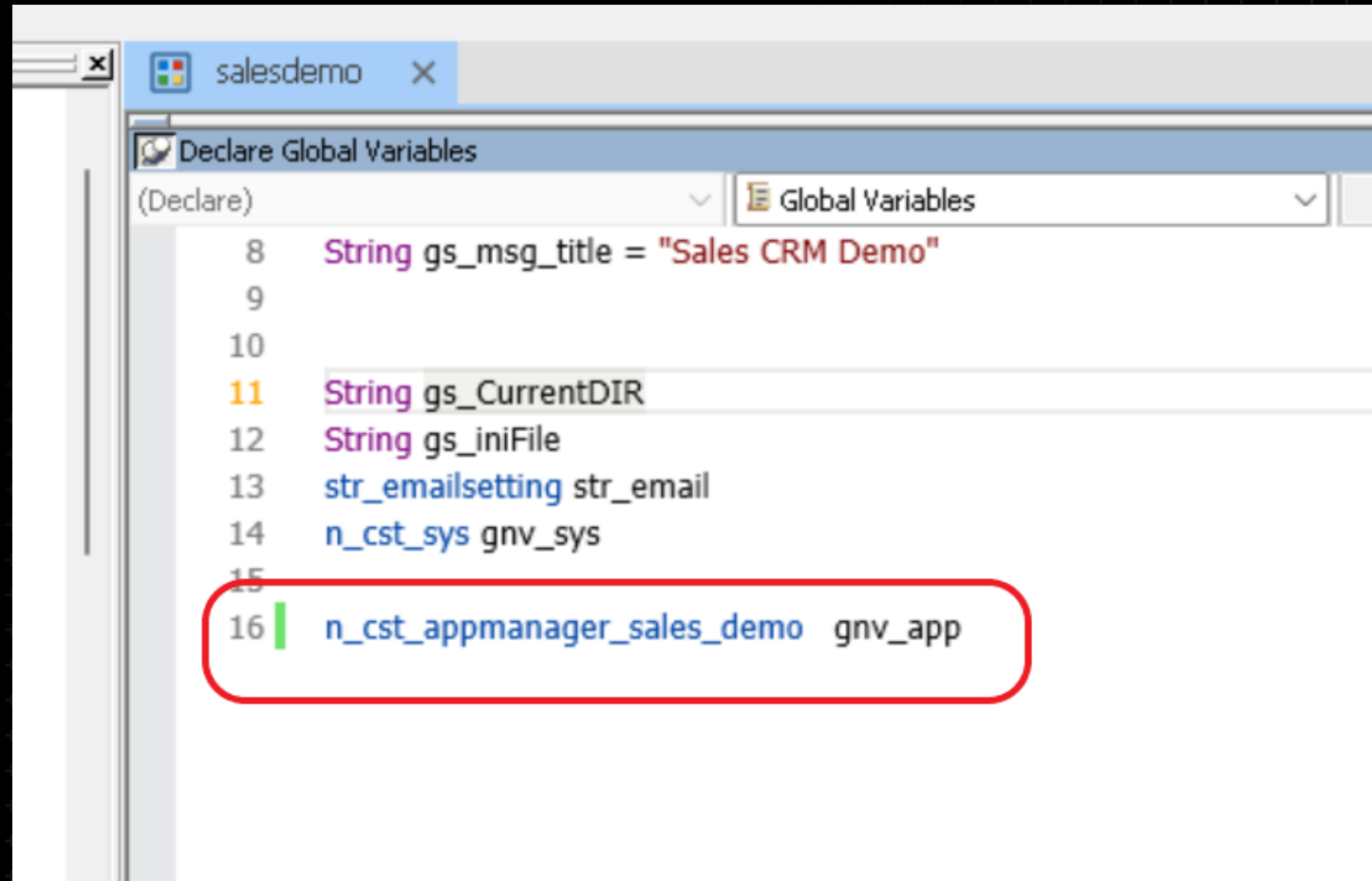
Add PFC libraries to the application library list



Inherit an application specific application manager from the n_cst_appmanager object in PFC



Declare the application specific application manager as a application global variable



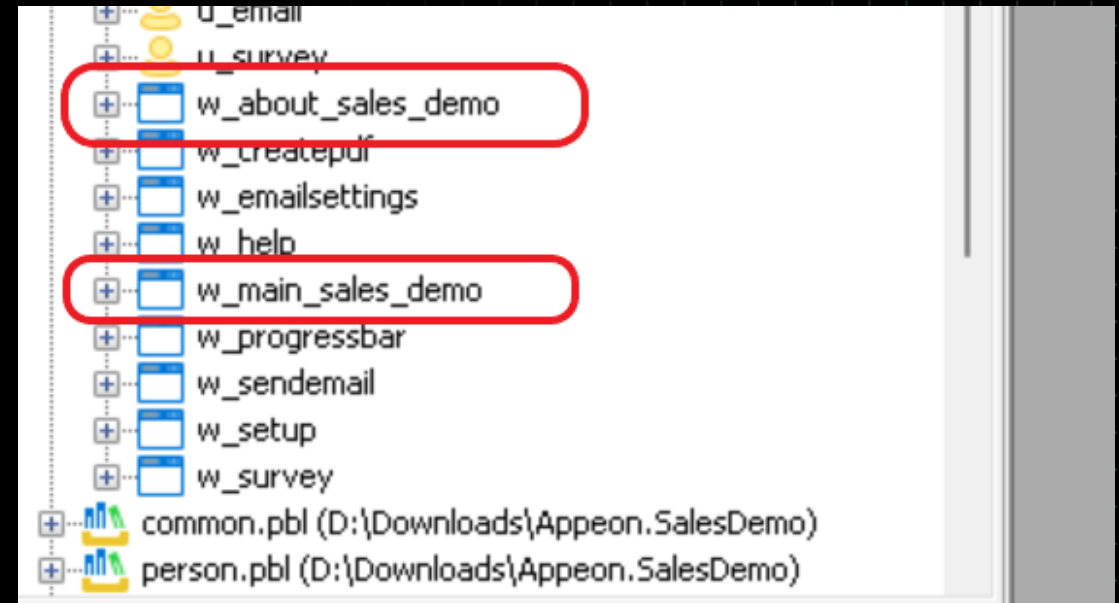
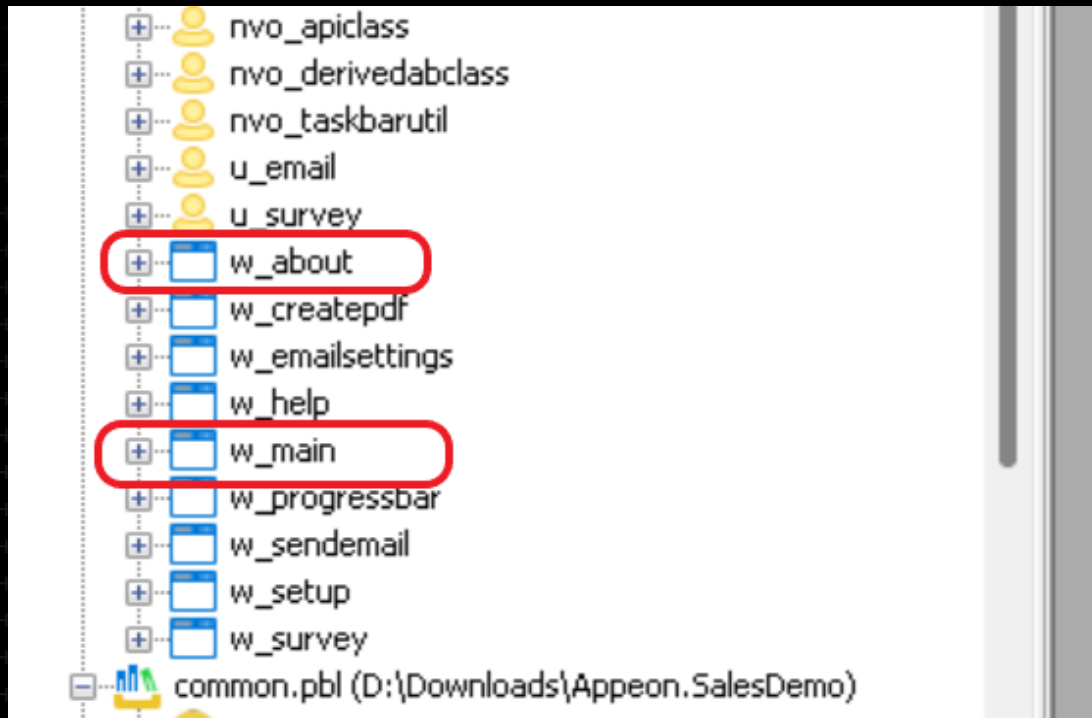
```
salesdemo x
Declare Global Variables
(Declare) Global Variables
8 String gs_msg_title = "Sales CRM Demo"
9
10
11 String gs_CurrentDIR
12 String gs_iniFile
13 str_emailsetting str_email
14 n_cst_sys gnv_sys
15
16 n_cst_appmanager_sales_demo gnv_app
```

Update the application variable types to use PFC classes

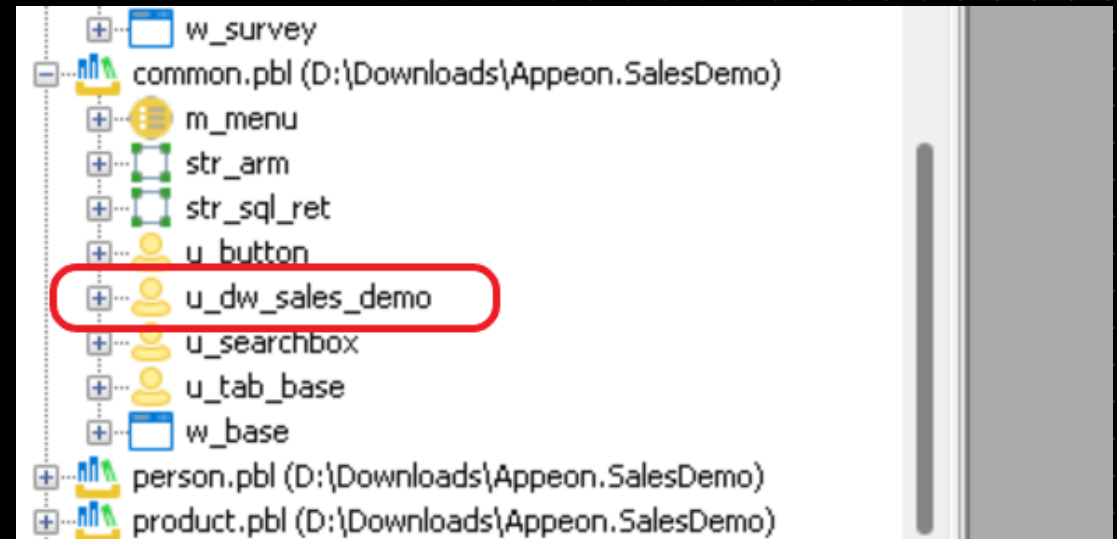
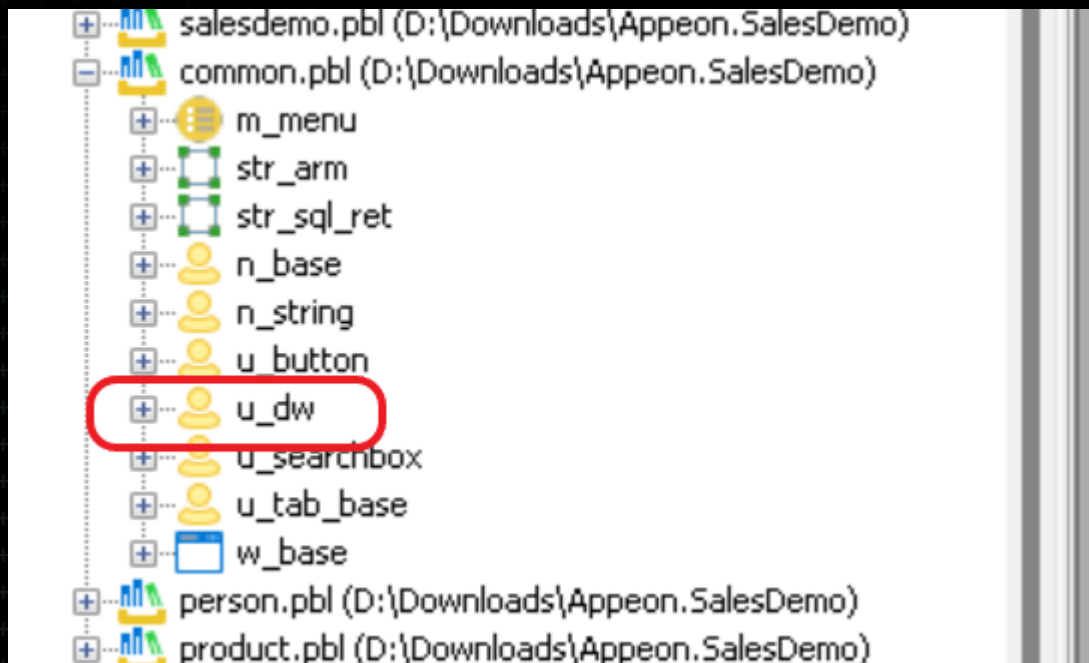
Application

Text Font	Column Font	Header Font	Label Font	Icon	Variable Types	RichTextEdit
SQLCA:						
n_tr						
SQLDA:						
n_dda						
SQLSA:						
n_dsa						
Error:						
n_err						
Message:						
n_msg						

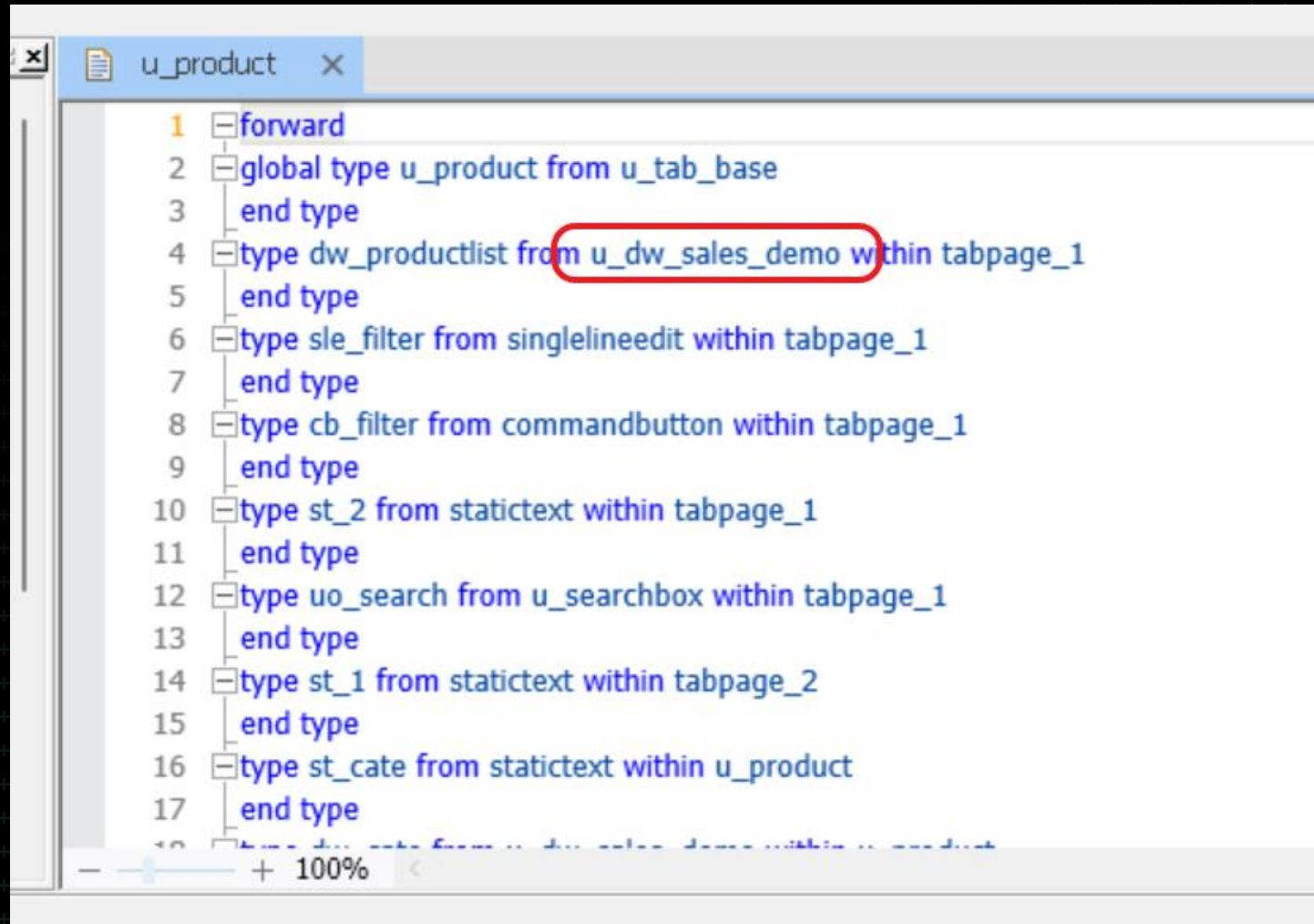
Deal with any object name collisions



Deal with any object name collisions

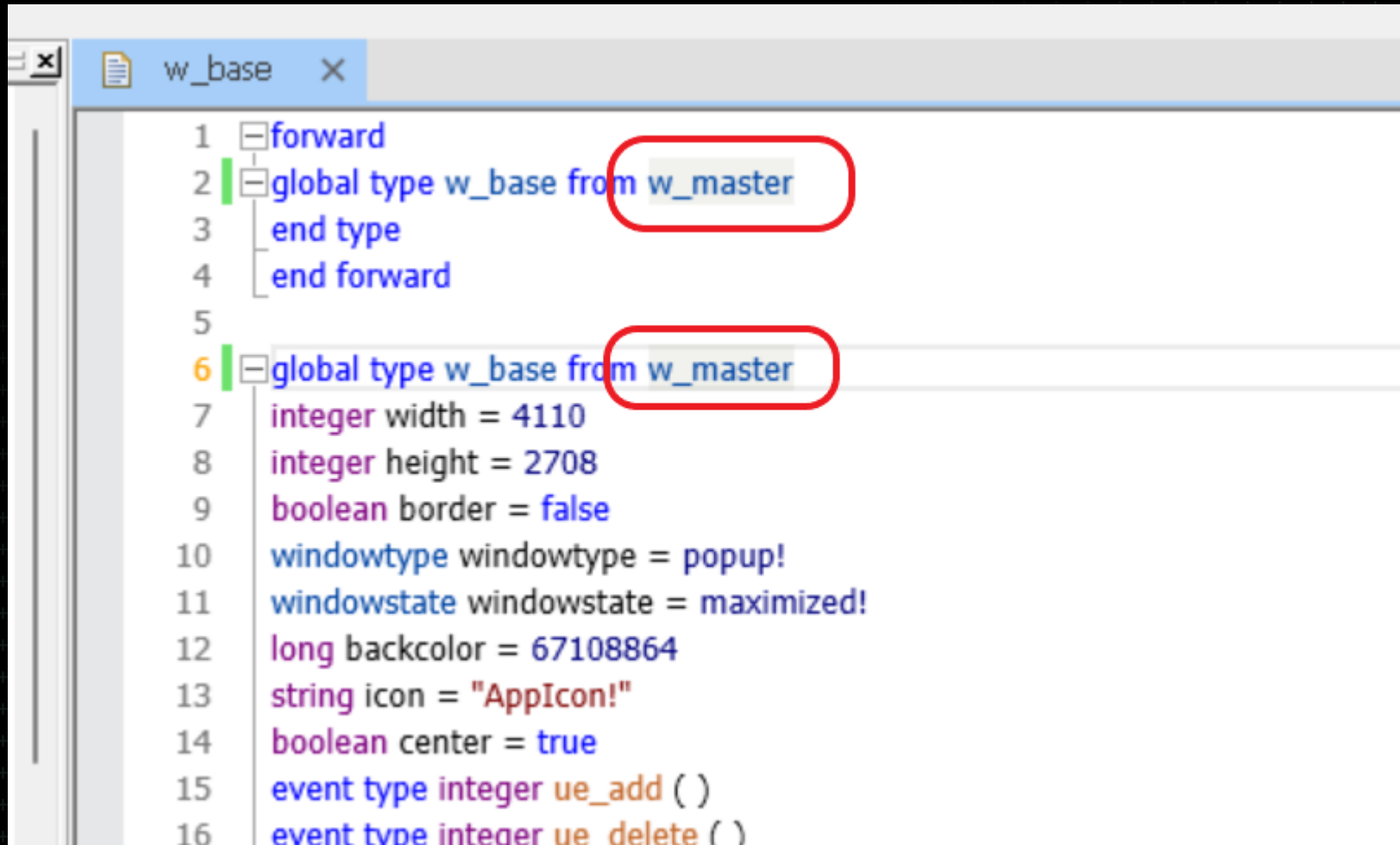


Deal with any object name collisions



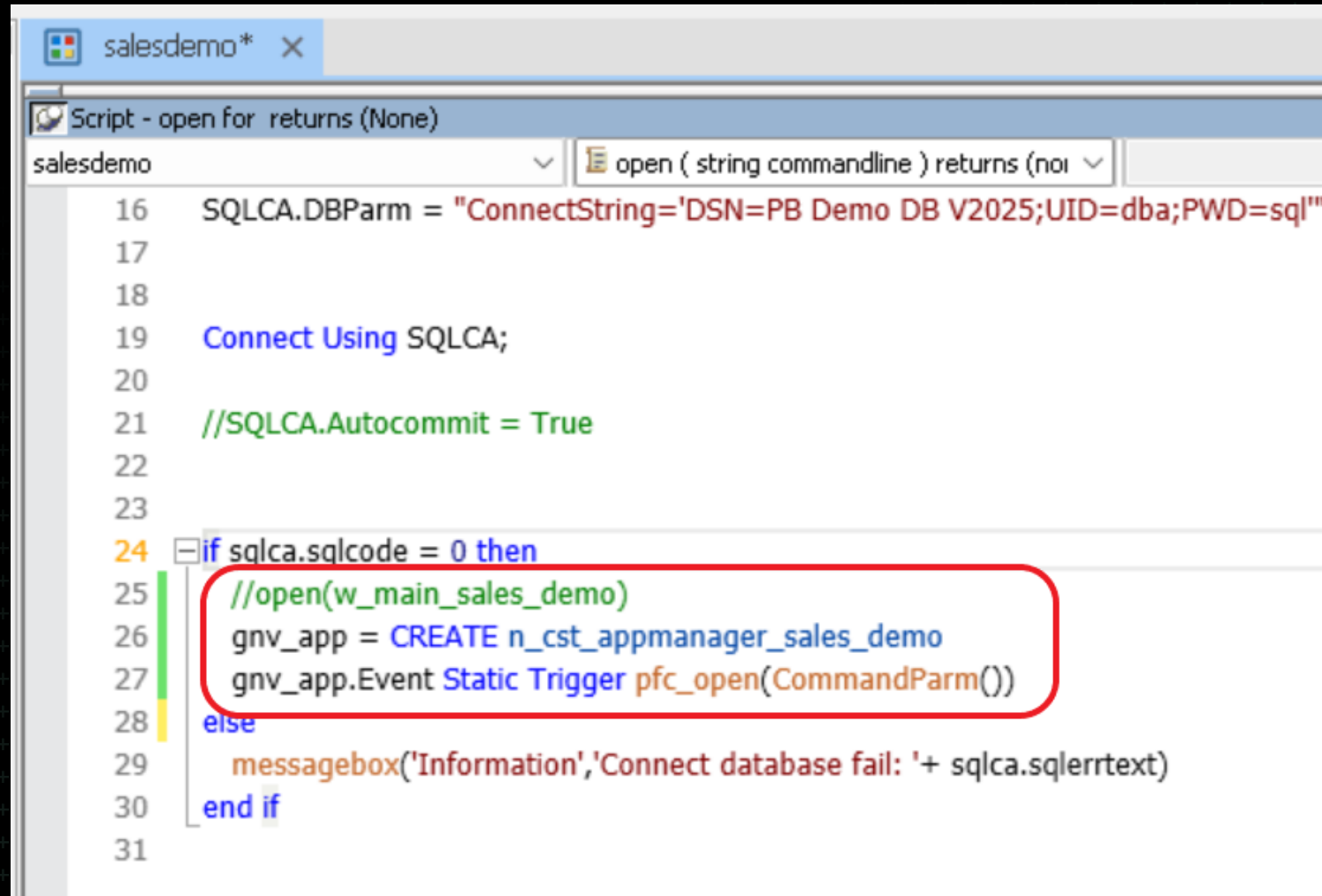
```
1 forward
2 global type u_product from u_tab_base
3 end type
4 type dw_productlist from u_dw_sales_demo within tabpage_1
5 end type
6 type sle_filter from singlelineedit within tabpage_1
7 end type
8 type cb_filter from commandbutton within tabpage_1
9 end type
10 type st_2 from statictext within tabpage_1
11 end type
12 type uo_search from u_searchbox within tabpage_1
13 end type
14 type st_1 from statictext within tabpage_2
15 end type
16 type st_cate from statictext within u_product
17 end type
18 type dw_cate from u_dw_sales_demo within u_product
```

Update window and user object ancestries



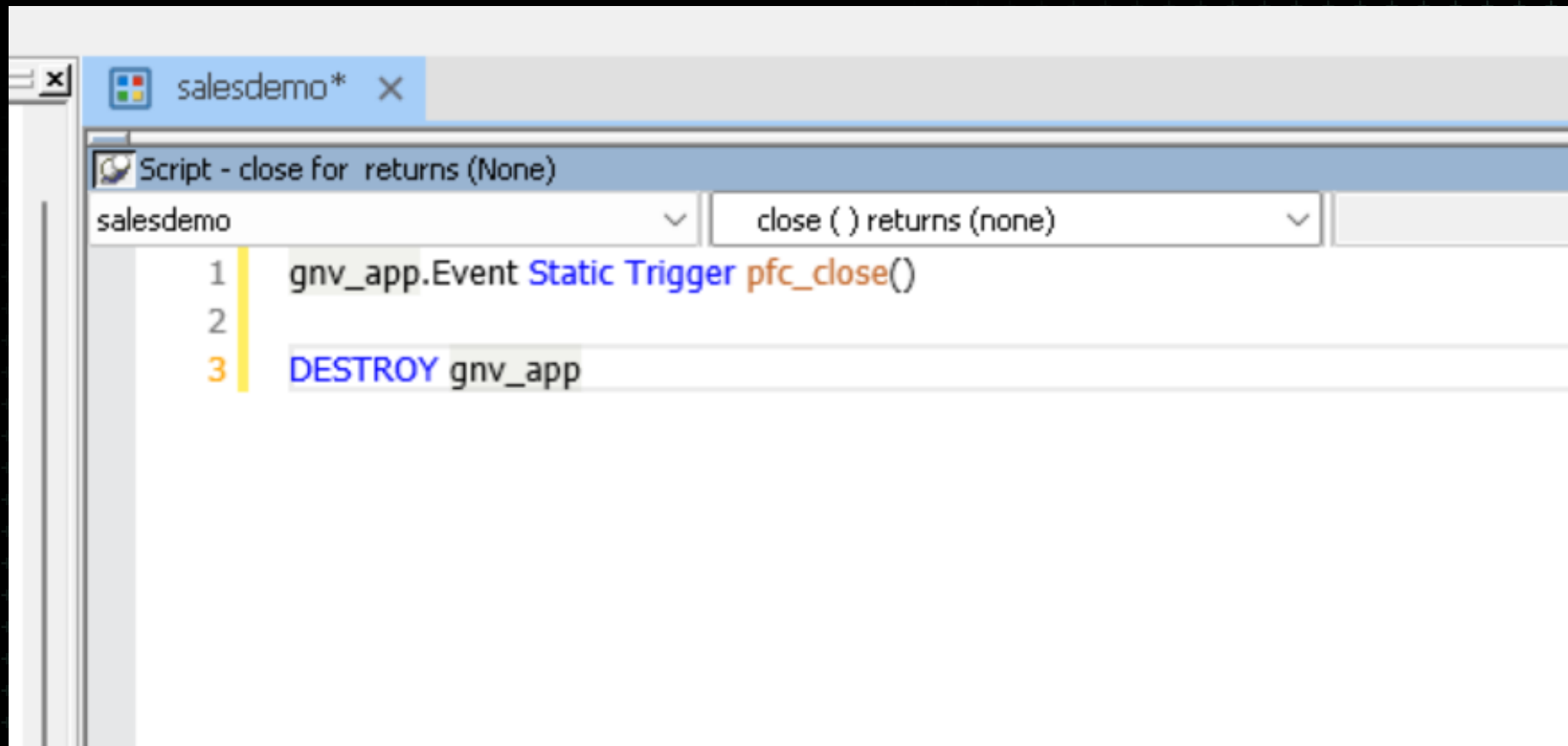
```
1 forward
2 global type w_base from w_master
3 end type
4 end forward
5
6 global type w_base from w_master
7 integer width = 4110
8 integer height = 2708
9 boolean border = false
10 windowtype windowtype = popup!
11 windowstate windowstate = maximized!
12 long backcolor = 67108864
13 string icon = "AppIcon!"
14 boolean center = true
15 event type integer ue_add ( )
16 event type integer ue_delete ( )
```

Modify application object events

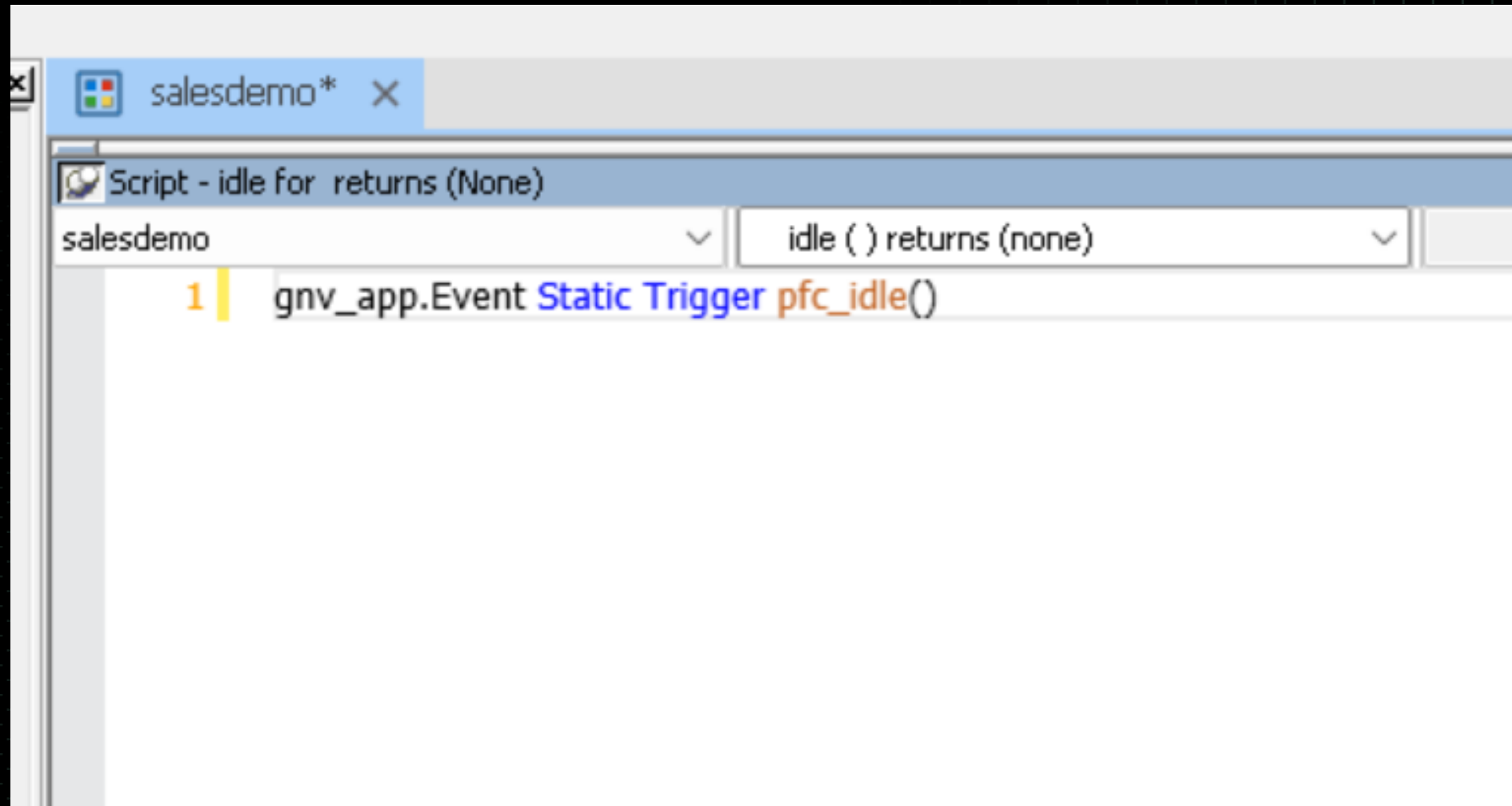


```
salesdemo* x
Script - open for returns (None)
salesdemo open ( string commandline ) returns (noi
16  SQLCA.DBParm = "ConnectionString='DSN=PB Demo DB V2025;UID=dba;PWD=sql'"
17
18
19  Connect Using SQLCA;
20
21  //SQLCA.Autocommit = True
22
23
24  if sqlca.sqlcode = 0 then
25      //open(w_main_sales_demo)
26      gnv_app = CREATE n_cst_appmanager_sales_demo
27      gnv_app.Event Static Trigger pfc_open(CommandParm())
28  else
29      messagebox('Information','Connect database fail: '+ sqlca.sqlerrtext)
30  end if
31
```


Modify application object events



Modify application object events



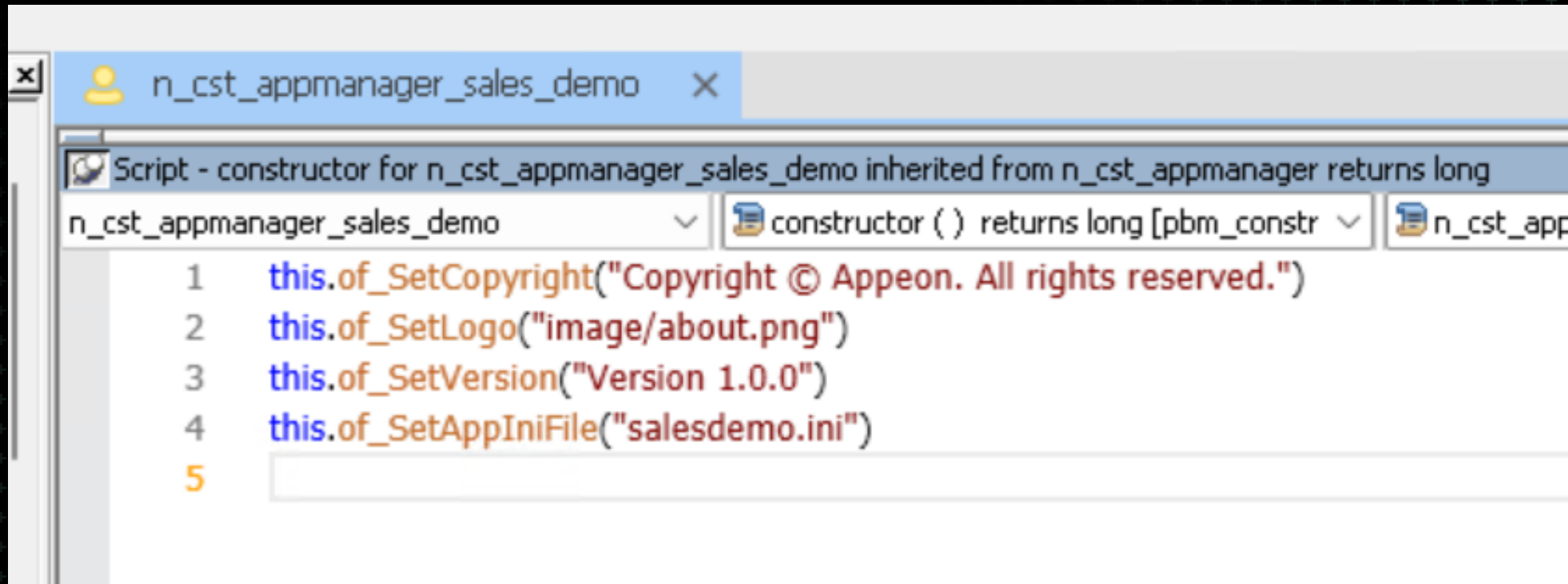
Modify application object events

```
Script - systemerror for returns (None)
salesdemo
systemerror ( ) returns (none)

1 Choose Case error.Number
2   Case 220 to 229 //Session Error
3     | | | MessageBox ("Session Error", "Number:" + String(error.Number) + "~r~nText:" + error.
4   Case 230 to 239 //License Error
5     | | | MessageBox ("License Error", "Number:" + String(error.Number) + "~r~nText:" + error.
6   Case 240 to 249 //Token Error
7     | | | MessageBox ("Token Error", "Number:" + String(error.Number) + "~r~nText:" + error.T
8   Case Else
9     | | | //MessageBox ("SystemError", "Number:" + String(error.Number) + "~r~nText:" + error
10    | | | gnv_app.Event Static Trigger pfc_systemerror()
11 End Choose
12
```

systemerror | Event List | Function List | Declare Instance Variables

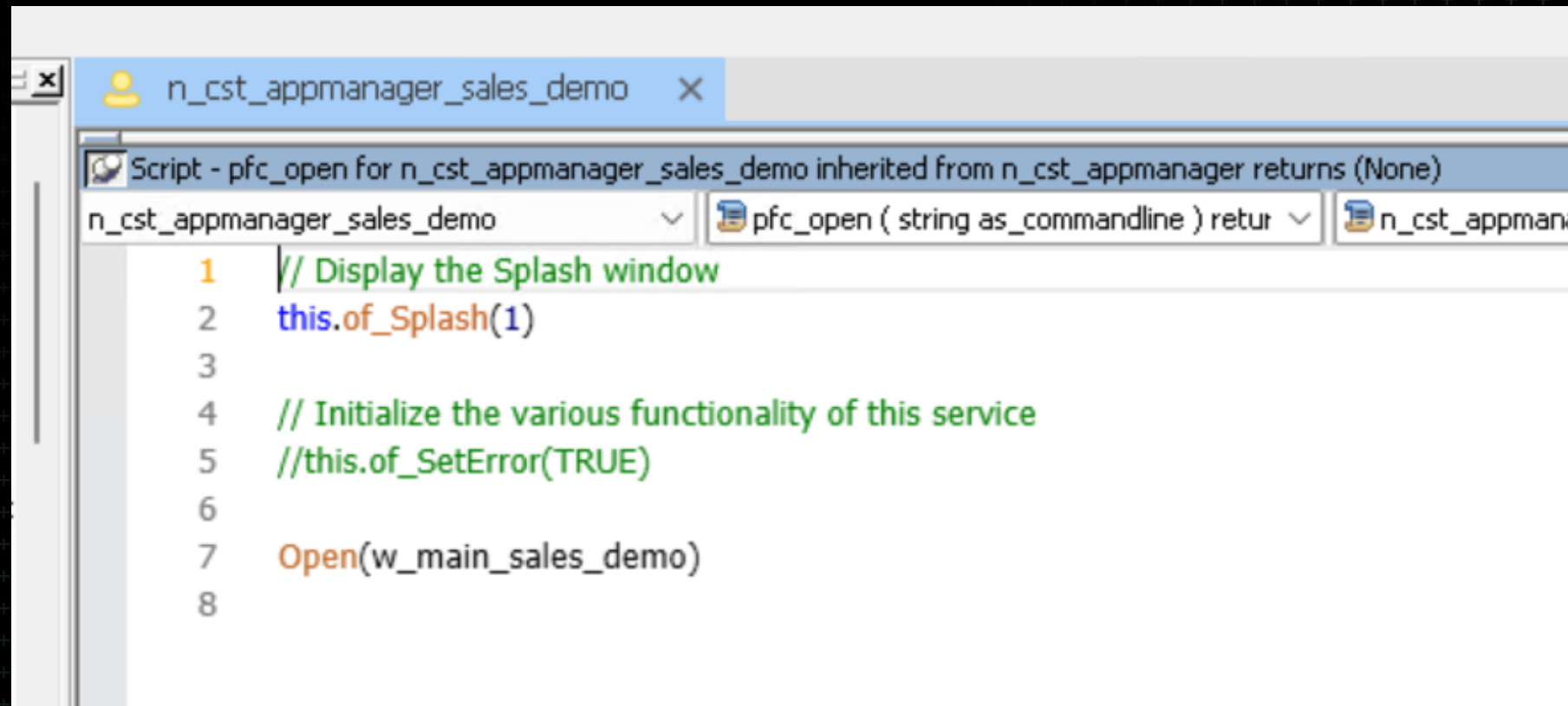
Code the application specific application manager constructor and pfc_open events



The screenshot shows a NetBeans IDE window titled "n_cst_appmanager_sales_demo". The editor displays the constructor code for the class, which is inherited from "n_cst_appmanager". The code is as follows:

```
Script - constructor for n_cst_appmanager_sales_demo inherited from n_cst_appmanager returns long  
n_cst_appmanager_sales_demo constructor ( ) returns long [pbm_constr  
1  this.of_SetCopyright("Copyright © Appeon. All rights reserved.")  
2  this.of_SetLogo("image/about.png")  
3  this.of_SetVersion("Version 1.0.0")  
4  this.of_SetAppIniFile("salesdemo.ini")  
5
```

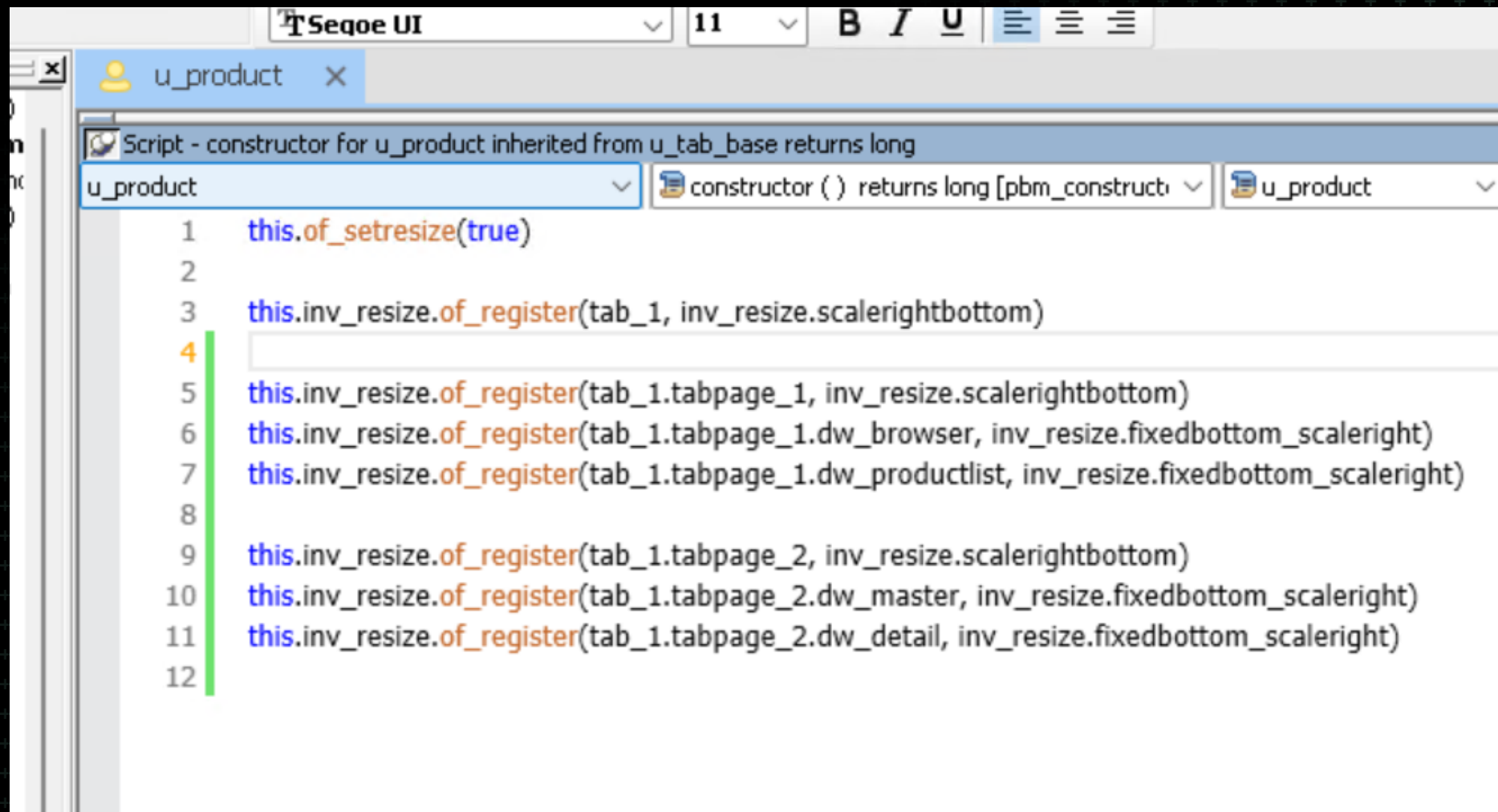
Code the application specific application manager constructor and pfc_open events



The screenshot shows a NetSuite script editor window titled "n_cst_appmanager_sales_demo". The script is for the "pfc_open" event, which is inherited from "n_cst_appmanager" and returns "(None)". The script is written in JavaScript and includes the following code:

```
1 // Display the Splash window
2 this.of_Splash(1)
3
4 // Initialize the various functionality of this service
5 //this.of_SetError(TRUE)
6
7 Open(w_main_sales_demo)
8
```

Start using PFC services

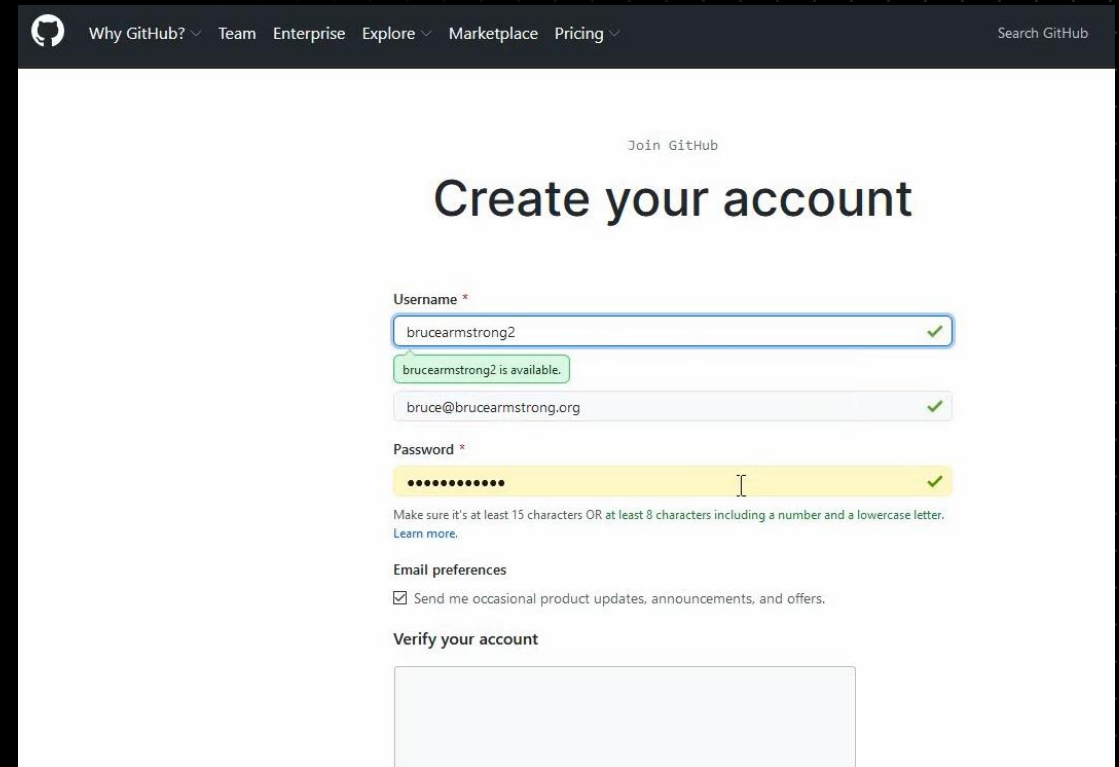


The screenshot shows a Visual Studio code editor window titled "TSeque UI". The editor is displaying a script for the constructor of the `u_product` class, which is inherited from `u_tab_base` and returns a `long` value. The script is written in a C#-like syntax and is located in a file named `u_product`. The script is as follows:

```
1  this.of_setresize(true)
2
3  this.inv_resize.of_register(tab_1, inv_resize.scalerightbottom)
4
5  this.inv_resize.of_register(tab_1.tabpage_1, inv_resize.scalerightbottom)
6  this.inv_resize.of_register(tab_1.tabpage_1.dw_browser, inv_resize.fixedbottom_scaleright)
7  this.inv_resize.of_register(tab_1.tabpage_1.dw_productlist, inv_resize.fixedbottom_scaleright)
8
9  this.inv_resize.of_register(tab_1.tabpage_2, inv_resize.scalerightbottom)
10 this.inv_resize.of_register(tab_1.tabpage_2.dw_master, inv_resize.fixedbottom_scaleright)
11 this.inv_resize.of_register(tab_1.tabpage_2.dw_detail, inv_resize.fixedbottom_scaleright)
12
```

How to contribute to PFC

- Create a GitHub account if you don't already have one



The screenshot shows the GitHub 'Create your account' page. At the top, there is a navigation bar with links: 'Why GitHub?', 'Team', 'Enterprise', 'Explore', 'Marketplace', and 'Pricing'. A search bar is on the right. Below the navigation bar, the text 'Join GitHub' is followed by the main heading 'Create your account'. The form includes three input fields: 'Username' with the value 'brucearmstrong2' and a green checkmark, 'Email' with the value 'bruce@brucearmstrong.org' and a green checkmark, and 'Password' with a masked password and a green checkmark. A green tooltip above the email field says 'brucearmstrong2 is available.' Below the password field, there is a note: 'Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)'. Under 'Email preferences', there is a checked checkbox for 'Send me occasional product updates, announcements, and offers.' At the bottom, there is a section for 'Verify your account' with a large empty box for a verification code.

Why GitHub? Team Enterprise Explore Marketplace Pricing Search GitHub

Join GitHub

Create your account

Username *

brucearmstrong2 ✓

brucearmstrong2 is available.

bruce@brucearmstrong.org ✓

Password *

..... ✓

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

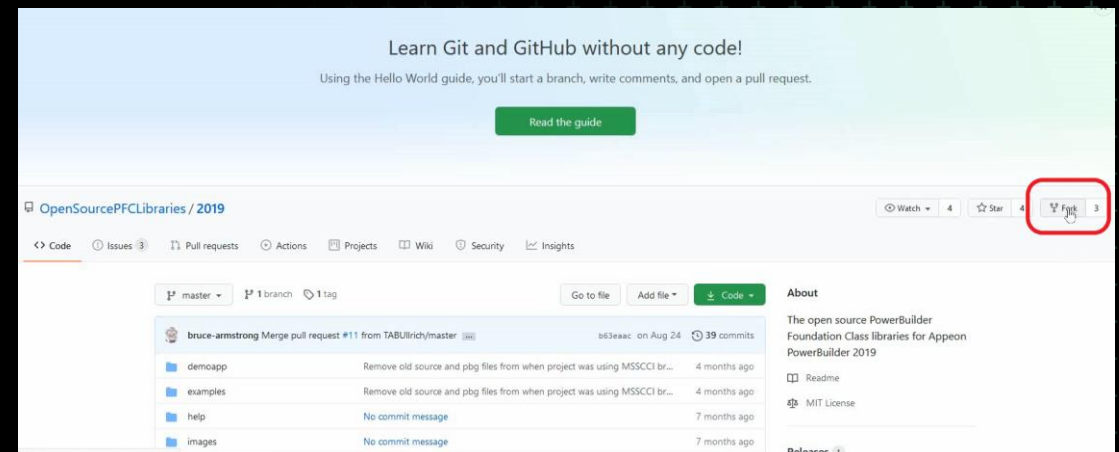
Email preferences

☒ Send me occasional product updates, announcements, and offers.

Verify your account

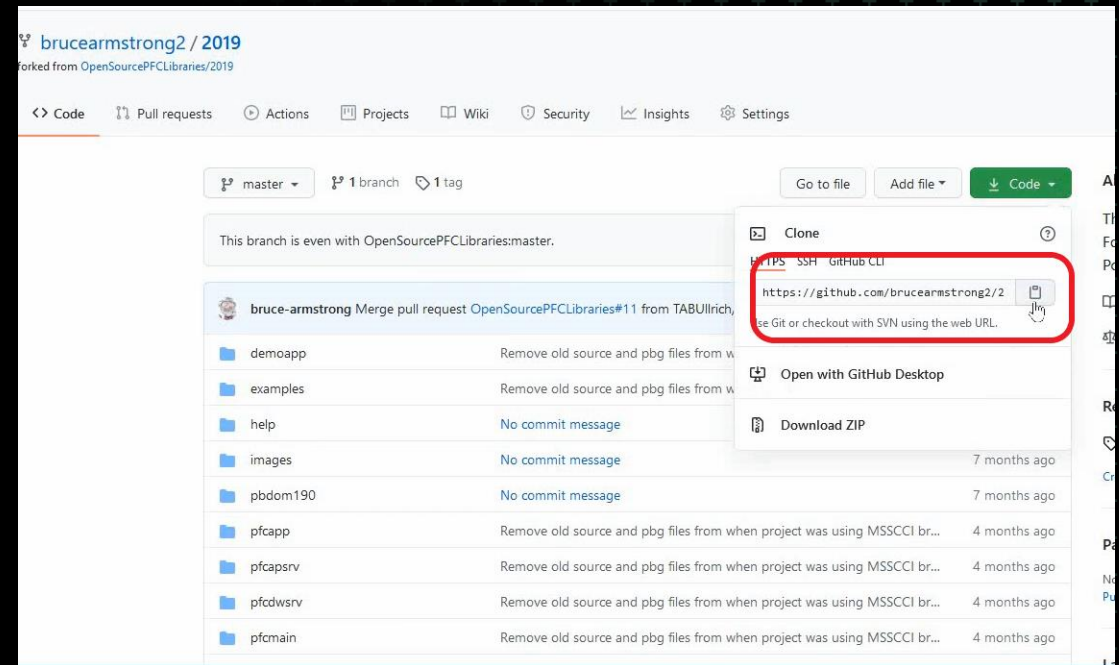
How to contribute to PFC

- Do a Fork from the Open Source PFC Library project in GitHub to create a copy in your own workspace in GitHub



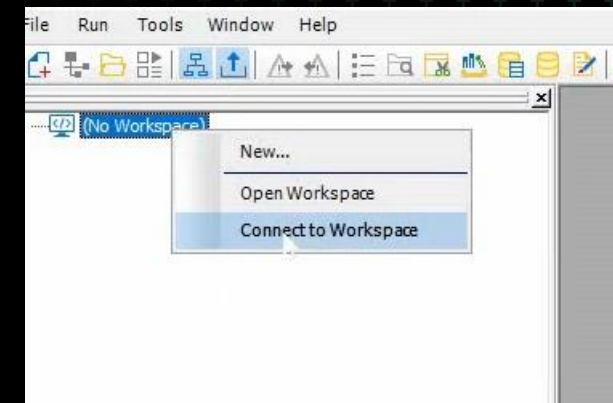
How to contribute to PFC

- Copy the URL for your copy of the project



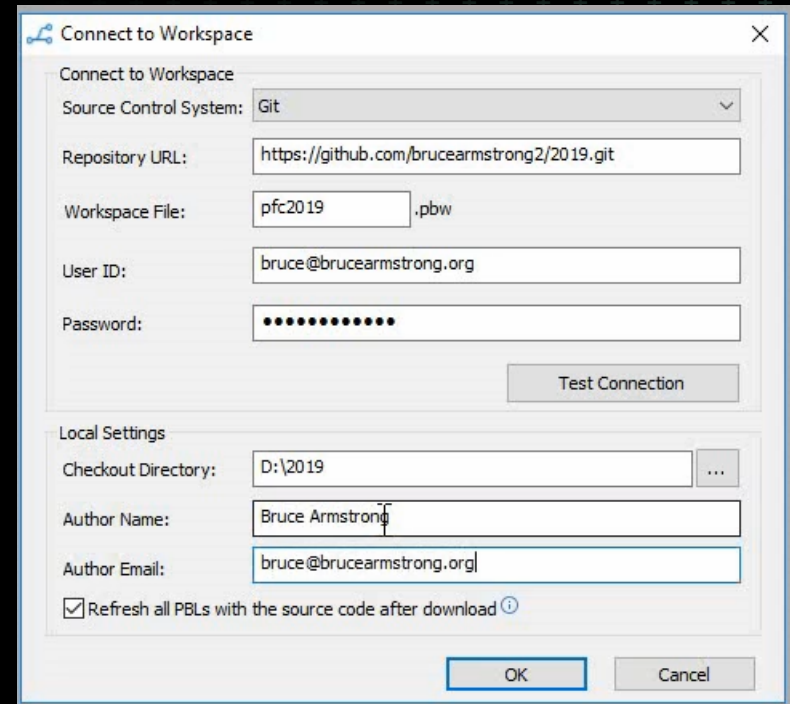
How to contribute to PFC

- Do a Connect to Workspace in the PowerBuilder IDE



How to contribute to PFC

- Paste in the GitHub URL and provider your GitHub credentials
- This will create the workspace, target and PBLs in a repository on your local machine

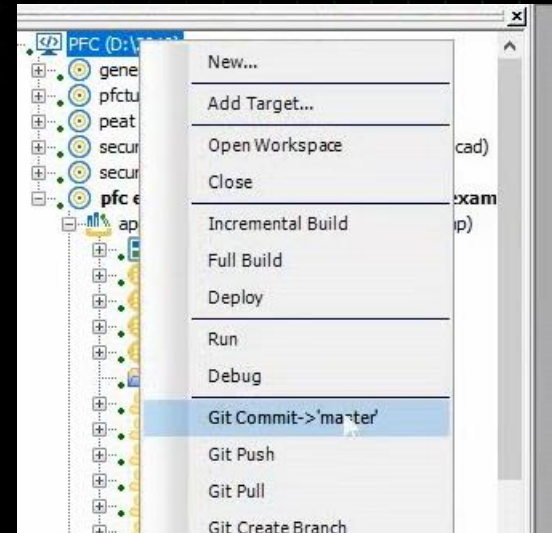


The screenshot shows a 'Connect to Workspace' dialog box with the following fields and options:

- Connect to Workspace** (Section Header)
- Source Control System:** Git (dropdown menu)
- Repository URL:** `https://github.com/brucearmstrong2/2019.git`
- Workspace File:** `pfc2019.pbw`
- User ID:** `bruce@brucearmstrong.org`
- Password:** (masked with dots)
- Test Connection** (button)
- Local Settings** (Section Header)
- Checkout Directory:** `D:\2019` (with a browse button)
- Author Name:** `Bruce Armstrong`
- Author Email:** `bruce@brucearmstrong.org`
- ☒ **Refresh all PBLs with the source code after download** (with an information icon)
- OK** and **Cancel** (buttons)

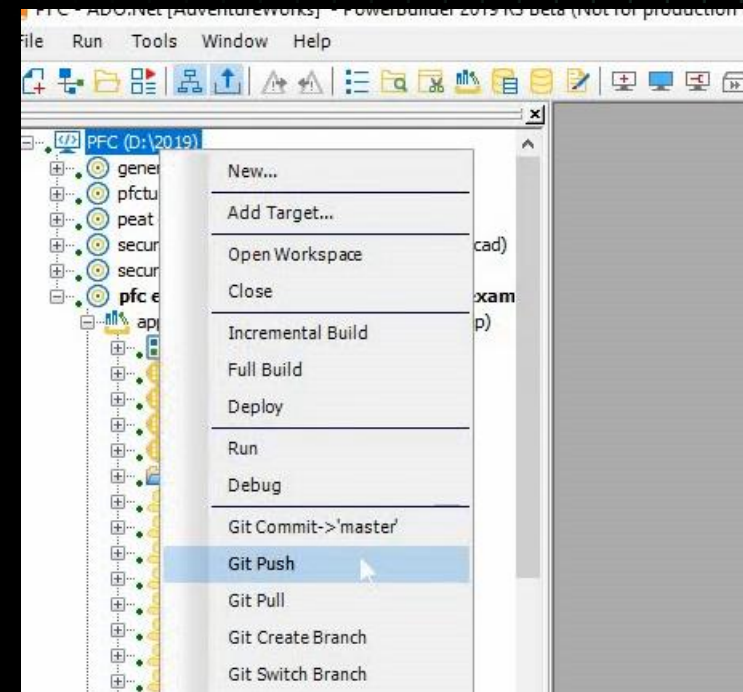
How to contribute to PFC

- After making the changes you want to contribute, do a Commit to your local repository



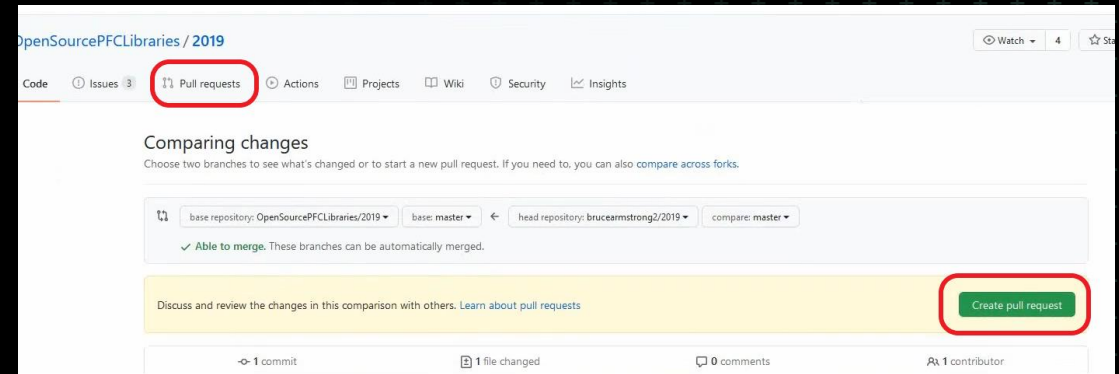
How to contribute to PFC

- The do a push to the project in your GitHub workspace



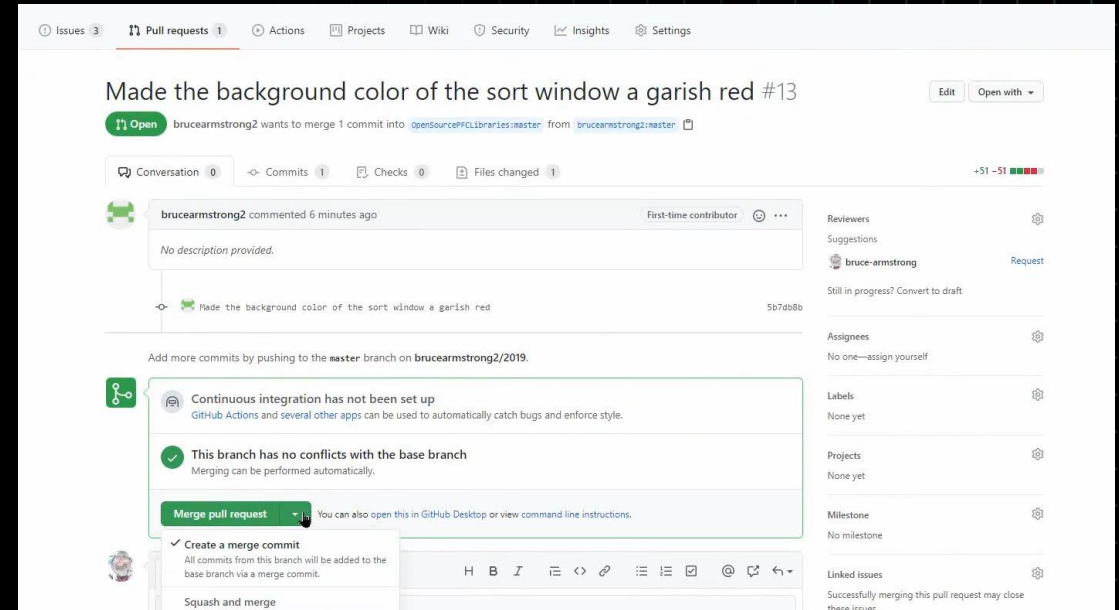
How to contribute to PFC

- Create a Pull Request in your GitHub workspace
- This submits the change to the main PFC project



How to contribute to PFC

- One of the admins in the main PFC project can now review the change, approve it or request changes to it



Session Agenda (recap)

- Why use the Open-Source PowerBuilder Foundation Class (PFC) libraries?
- Why upgrade to the new PFC?
- How to upgrade to the new PFC?
- How to migrate an existing application to use PFC?
- How to contribute to PFC?

Connect with Us



community.appeon.com

Discussions, tech articles and videos, free online training, and more.



facebook.com/AppeonPB

Encourage us with a “like”, see cool pics, and get notified of upcoming events.



linkedin.com

Build up your career profile, and stay in contact with other professionals.



twitter.com/AppeonPB

Follow Appeon and community members to get the latest tech news.



youtube.com/Appeon

Share important Appeon videos with others; no account registration required.

Thank you

Q&A Time

