

# Plex Goes Green: 5250 Techniques in Plex

12D

Bob Minor

ADC Austin

**ADC AUSTIN**

**MKS**

**WEBSYDIAN™**

**cm** FIRST

**desynit**  
**ca**

# Plex Goes Green

---

- > Why Green Screens in Plex?
- > Basic implementation

# Why Green Screens in Plex?

---

## > Plex because

- Productivity
- Model based development
- Standards enforced

## > Plex over 2E because....

- Development environment
- Inheritance
- Meta code
- Multi-platform

# Why Green Screens in Plex?

## > Green screen because

- Host based - ease of deployment
- Ease of maintenance
- Compatibility with other systems
- Familiar look and feel for users
- Green screens can be easier for certain types of data entry

# Implementation - Make your panels green

---

## > One triple

- Panel – mode - DDS



# Conclusion

---

> Thanks

# Not quite....

---

## > History

- Obase vs Patterns
- Experiences

## > Strategies for a comprehensive solution

## > Fixes and enhancements

# History - 5250 in OBASE

---

- > Designed with full support of 5250
- > Not the future direction of Plex
- > Patterns are needed for other target platforms
- > Many Plex developers are now unfamiliar with OBASE

# History - 5250 in Patterns

---

- > Patterns are the basis of current programming techniques
- > Flexible platform options
- > Familiar to most Plex developers
  
- > There is no broad explicit 5250 support
- > Requires some tuning

# History - Experiences

---

- > OBASE development
- > 5250 in patterns
- > Development of standardized 5250 patterns
- > Client specific development
- > Customized function types
- > ADC Migration Studio
- > Enhancements and fixes to standardized patterns

# What is in a comprehensive solution?

---

- > 5250 Variant
- > Low level functions with 5250 variants that can be used as the basis for higher level function types
- > High level functions with addition adjustments for 5250
- > Standard coding, source code, meta functions and utilities that can be used change behavior to RPG and CA 2E norms

# 5250 Variant

---

- > Create a 5250 Variant in inheritance layer
- > Can be an existing standards layer
- > Can be a new pattern
- > Allows to change variants for regeneration to other languages and platforms

# Using the Variant

---

## > Creating the Variant

- Log in to the group model
- Configuration>Variants>Add

> Any variant specific triples go in the 5250 Variant

> RPG language triples

> Test for the variant in any platform specific code

# Low Level Functions

---

- > UI Shell
- > Headers
- > Footers
- > Confirm prompt
- > Various program options
- > Grid
- > Key screen
- > Window
- > Additional layers with added features

# UI Shell

Program Name

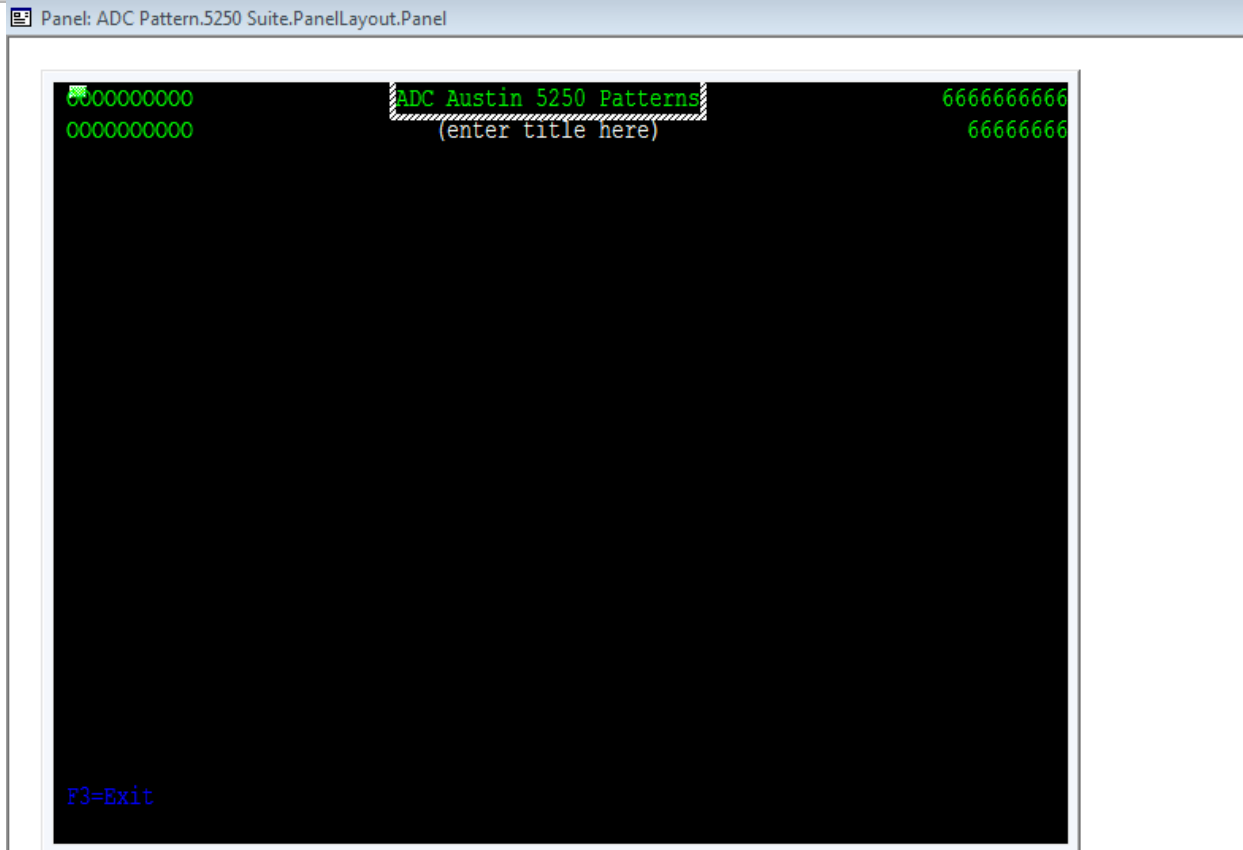
User

```
0000000000 ADC Austin 5250 Patterns 6666666666  
0000000000 (enter title here) 6666666666  
  
F3=Exit
```

Date

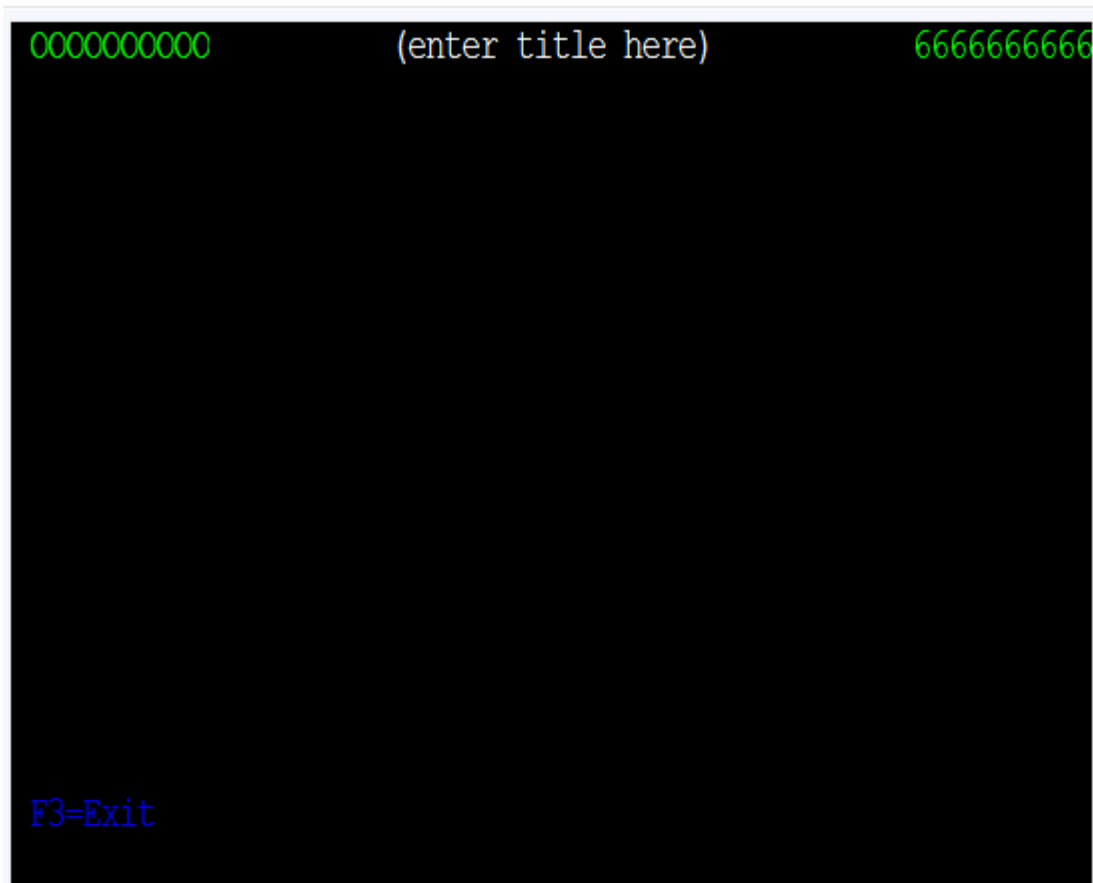
Time

# Panel Layout Function



Single Instance UI Function that Prompt Record, Delete Record, Change Record, Create Record and any other Single instance functions inherit from. Make General Layout Mapping in this Panel and any common Action Diagramming in this function

# UI Window



# High Level Functions

---

- > Edit File
- > Display File
- > Edit Record (Full featured similar to 2E)
- > Add Record (For a Plex-like suite)
- > Change Record
- > Delete Record
- > Suites

# Edit File

```
0000000000          ADC Austin 5250 Patterns          6666666666
0000000000          (enter title here)              6666666666

Type options, press Enter: 2=Change 4=Delete 5=Display

[ ]

F3=Exit  F5=Refresh          F9=Create
```

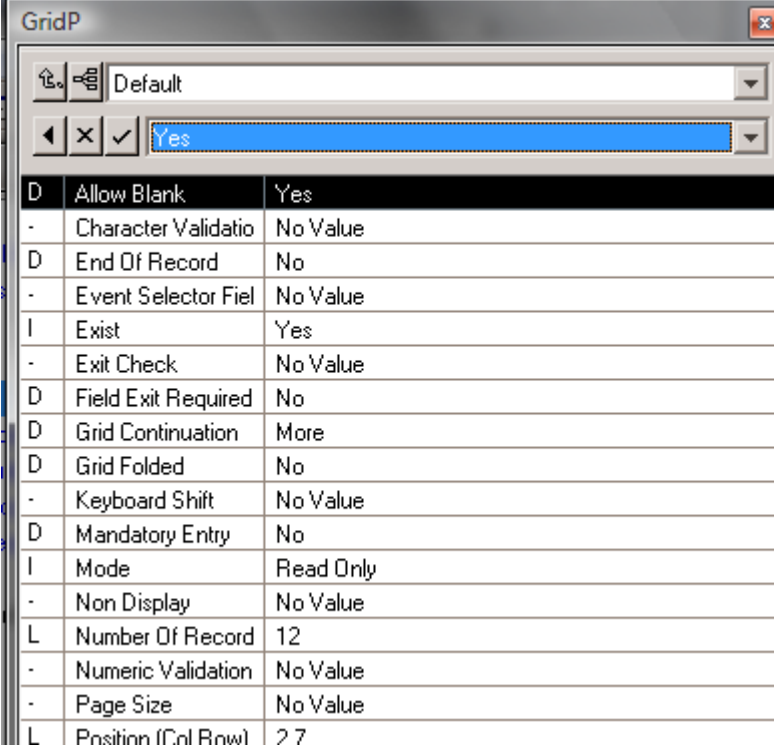
# Additional development and modifications

---

- > Additional programming features
- > Unexpected behavior

# Variant Specific Properties

## > Event Selector Field



The screenshot shows a window titled "GridP" with a "Default" dropdown menu and a "Yes" dropdown menu. Below these is a table with the following data:

D	Property	Value
-	Allow Blank	Yes
-	Character Validatio	No Value
D	End Of Record	No
-	Event Selector Fiel	No Value
I	Exist	Yes
-	Exit Check	No Value
D	Field Exit Required	No
D	Grid Continuation	More
D	Grid Folded	No
-	Keyboard Shift	No Value
D	Mandatory Entry	No
I	Mode	Read Only
-	Non Display	No Value
L	Number Of Record	12
-	Numeric Validation	No Value
-	Page Size	No Value
L	Position (Col Row)	2 7

# Set Focus

---

## > Problem

- After Prompting (F4) on a field and the result returning the cursor would go to the first editable field on the region. The user would have to tab passed the field they prompted on

## > Solution

- Create own meta code function and add the "Set Focus" line to keep cursor on field.

# Set Focus

 +For Each Field

++If Input Capable

++If Focus

+++Set Value To Current Field: FIELDS/+Field

Store the name of the field with focus

++Name Defined Field: FIELDS/+Field, Environment<\*Object name>

Modified

+For Each Field

++If Input Capable

++If Focus

+++Set Value To Current Field: FIELDS/+Field

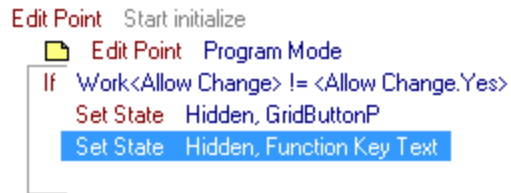
Store the name of the field with focus

++Name Defined Field: FIELDS/+Field, Environment<\*Object name>

++Set Focus

Keeps the focus on field where user has pressed F4- Prompting

# Switch between Edit/Display Mode



Set State to show/Hide GridButtons and Text on Page

# Positioning on Grid



All Programmers have to do is to add the fields to the PositionGrid Variable on the Panel Set the Region to visible = Yes and add the fields to Previous Values to compare values on PositionGrid Region on Panel with Previous Values for positioning. If the values are the same the program ignores this and processes any other events eg 4 - Delete

# SubFile Cursor Support

---

## > Problem

- If roll up and cursor was on line 10 and now there are only 6 records Cursor stays at row 10 even if there were data shown for that row



# Roll Control

---

## > Problem

- Roll up and down works correctly
- Getting More on Subfile when there was no “more” records

# Fixes position of Cursor when Page up/Page Down

```
Source code: [(Read-Only)]Utility Functions.Subfile Rollup
C      select
C      when      ZRD1 > *ZERO
C      if        (ZRD1 + &(1:)) <= ZRX1
C      eval      ZRD1 += &(1:)
C      endif
C      *
C      when      ZRD2 > *ZERO
C      if        (ZRD2 + &(1:)) <= ZRX2
C      eval      ZRD2 += &(1:)
C      endif
C      *
C      when      ZRD3 > *ZERO
C      if        (ZRD3 + &(1:)) <= ZRX3
C      eval      ZRD3 += &(1:)
C      endif
C      *
C      when      ZRD4 > *ZERO
C      if        (ZRD4 + &(1:)) <= ZRX4
C      eval      ZRD4 += &(1:)
```

# Subfile Cursor RRN

---

## > Problem

- If deleting a row on a Edit File the row disappears, but Plex was not aware of it going so
- On Prompt on Edit file was not always working

# SubFile Cursor RRN

```
Source code: [(Read-Only)]Utility Functions.Subfile Cursor RRN
C*****
C*  Force the "Sub File RRN" to be equal to the "Cursor RRN"
C*  This is required to allow the Edit File Prompting to work
C*****
C          select
C          when      ZRX1 > *ZERO and ZRC1 > *ZEROS
C          eval      ZRR1 = ZRC1
C          *
C          when      ZRX2 > *ZERO and ZRC2 > *ZEROS
C          eval      ZRR2 = ZRC2
C          *
C          when      ZRX3 > *ZERO and ZRC3 > *ZEROS
C          eval      ZRR3 = ZRC3
C          *
C          when      ZRX4 > *ZERO and ZRC4 > *ZEROS
C          eval      ZRR4 = ZRC4
C          *
C          when      ZRX5 > *ZERO and ZRC5 > *ZEROS
C          eval      ZRR5 = ZRC5
C          *
```

# Time and Date

---

## > Problem

- Plex calls a CLPGM to get time and date and we did not want all the joblog to log all these calls

## > Solution

- Add RPGIV SRC to do this

# Time and Date

---

```
C      MOVE      *ZEROS      TIME      6 0
C      MOVE      *ZEROS      DATE      8 0
C      MOVE      *ZEROS      MMDDYYYY   8
C      MOVE      *ZEROS      YYYYMMDD   8
C      MOVE      *ZEROS      DATE8MDY   8 0
C      MOVE      *ZEROS      DATE8YMD   8 0
C      EVAL      DATE = %dec(%date())
C      EVAL      TIME = %dec(%time())
C      EVAL      YYYYMMDD = %EDITC (DATE:'X')
C      EVAL      MMDDYYYY = %SUBST (YYYYMMDD:5:4)
C                        + %SUBST (YYYYMMDD:1:4)
C      EVAL      DATE8MDY = %dec (MMDDYYYY:8:0)
C      EVAL      DATE8YMD = %dec (YYYYMMDD:8:0)
C      MOVE      DATE8YMD      &(1:)
C      MOVE      DATE8MDY      &(2:)
```

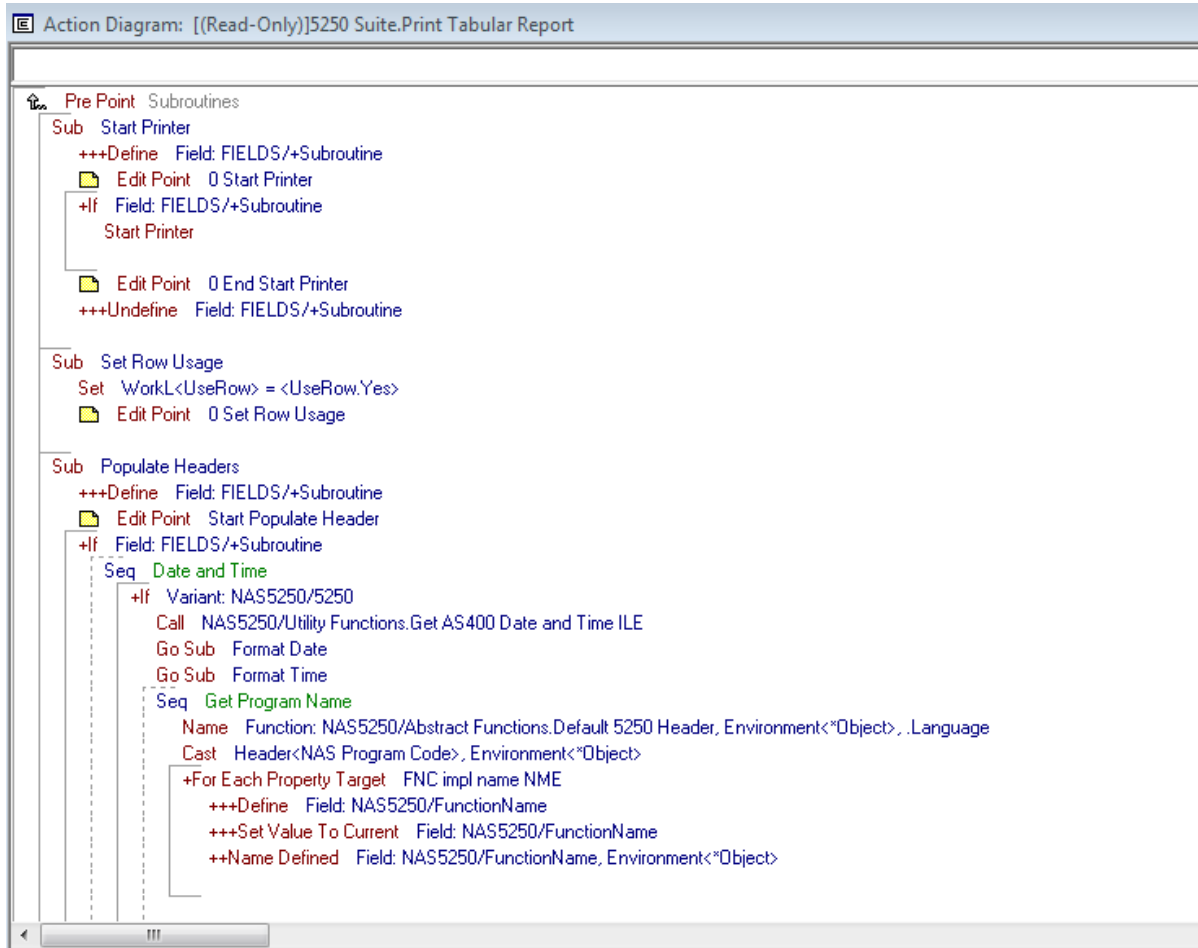
# Print Function

---

## > Replicated Print functions from OBASE

- Tabular Report and Key Break Report
- Added logic to be able to use Print Functions as Print Objects combining for Runtime
- Added Title for Program Name, User, Date, Time, Company Name and Report Title to Report Region
- Added Selections region to print with Print Selection passed into Report Function

# Print Tabular Report



# Solution Summary

---

- > Variants increase flexibility
- > Patterns allow full use of the inheritance power of Plex
- > Detailed behavior needs to be modified to correct defects and satisfy traditional 5250 users
- > Its easy to create a 5250 panel, its harder and more involved to build a comprehensive solution

# ADC 5250 Patterns

---

- > Variants
- > Low level building block functions
- > Utilities and source code
- > High level functions and suites
- > Oriented towards standard 5250 and 2E behaviors

# Summary

---

- > Green screens are a popular option in Plex
- > Green screens can be developed using Patterns
- > A well designed pattern layer keeps future options open
- > Some fine tuning is required
- > ADC has developed a complete solution

# Conclusion

---

> Thanks

> Really