Dynamic Page Concept – UI & Logic Separation

This document describes the architectural concept for dynamic pages in the AmiumScripter framework. The design separates business logic, control structure, and UI behavior for maximum flexibility, reusability, and maintainability.

# Concept Overview

Each Page consists of three files and separates core responsibilities:

1. 1. Page.cs – Page Logic

- Contains business logic, data models, thread or timer logic.  
- Defines and manages all relevant signals (BaseSignal and derivatives).  
- Does NOT contain any direct UI code or WinForms controls.

1. 2. controls.cs – Control Declarations

- Declares only the visual controls (e.g., SignalView, buttons, grids) and their layout.  
- Can be modified by the App UI Editor or by code.  
- Contains no logic or event handling – only declarative UI structure.

1. 3. view.cs – UI Behavior

- Contains all logic affecting visual controls (highlighting, event binding, custom behaviors).  
- Does not define data logic or initialize controls, but orchestrates their behavior and appearance.

# Benefits

- Clean separation of concerns: Business logic, UI structure, and UI logic are separated.  
- Enables dynamic (re-)building of UI without touching core logic.  
- Allows for programmatic and visual UI editing.  
- Promotes code reuse, modularity, and easy maintenance.

# Component Responsibilities

|  |  |  |
| --- | --- | --- |
| File | Responsibilities | Notes |
| Page.cs | Logic, signal declaration, no UI | Handles threads, data flow, signal models. |
| controls.cs | Declares controls and their placement | Generated by UI editor, layout only. |
| view.cs | Manages UI behavior (events, visuals) | Orchestrates control interactions. |

# Architectural Diagram

The following diagram shows the relationships and data flow between the components:

+-----------+  
 | Page.cs |  
 +-----------+  
 | ^  
Signals, Data | | Uses signals, logic  
------------------+ |  
 v |  
 +----------------+  
 | view.cs |  
 +----------------+  
 | ^  
Events/ | | Control instances  
Behavior | |  
 v |  
 +--------------+  
 | controls.cs |  
 +--------------+

# Summary

This concept provides a robust, scalable foundation for dynamic, modular WinForms UI design, where logic, layout, and behavior are clearly separated for maintainability and extensibility.