Dynamic Pivot

The development of dynamic pivot will go in two independent modes i.e

1. Pivoting Library
2. Container Component

Pivoting Library

The goal of the pivoting library is to simplify the creation of pivot blotters for various in-house projects. Developers should be able to set up a pivot blotter effortlessly by simply providing the following: a data source, column and row pivots, and optional filters. The key features to include are:

- Support both streaming and static data from diverse data sources (Phase 1 will focus exclusively on AMPS).
- Handle datasets of varying sizes: either load all data on the client-side or fetch data on demand in smaller chunks.
- Enable persistence and restoration of grid states, including column layout, widths, filters, and other configurations.

Container Component

The container component can be envisioned as a fully-fledged application, ultimately designed to provide users with a blank canvas for hosting any desired components. However, to maintain focus and simplicity, Phase 1 will concentrate on developing a Pivot Grids host component. This initial implementation will be structured to align with the broader long-term vision.

A primary objective of the container component in Phase 1 is to empower end users to create pivot blotters for their data without writing any code. To achieve this, the container component must include:

S#	Feature	Due date
1.	A container to host collection of grid components, the container can be: 1. *Tab based where only single pivot is visible* 2. **Tiles based where tiles are laid out in the form of table/grid, each tile to be resizable, can change position by drag drop (Phase-1)** 3. **Ability to add a new component (Pivot Grid for the phase-1)** 4. **Ability to remove any hosted component** 5. **Save & Restore layout** 6. **Multiple container instances (example: Let's assume, we rolled this application out. I as a user, want to create 2 different dashboards one with full of pivots of rates risk and another dashboard with credit risk pivots of different risk models)** 7. **Reduce the dependency on dpd for layouts i.e let dpd handle the window state i.e x,y cordinates & window dimension but take charge of what's inside the window. To put in simple words, DPD handles the external part of the window and we manage what's inside**	
2.	**Pivot Host Component**:	

The purpose of this component is to enable end users to create and configure pivot grids without writing any code. It will host the Pivot Grid Library component and pass the required properties dynamically through a user-friendly GUI. This means:

1. Let user pick the data-source, have named sources configured in dpd config
2. Let user pick the columns needed
3. Let user pick the initial row & column pivots
4. Let user pick the initial filters
5. Stream/static, All data on client or on-demand