

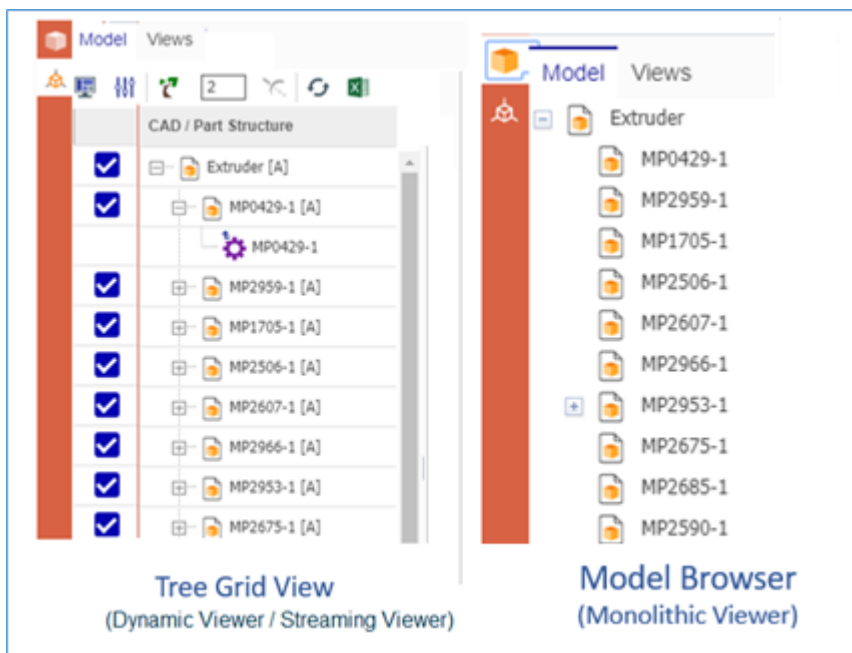
Tree Grid View vs. Model Browser

The concept and vision of the Dynamic Product Navigation (DPN) require an alternate mechanism for displaying business objects (Items) associated with 3D components displayed in the 3D View:

- The *dynamic* part of DPN specifies that some conditional and customizable logic should determine 3D view content. A QD provides this ability.
- The *navigation* part of DPN specifies that the 3D View can be used to access *related* business objects (Items). A TGV Definition provides this ability.

The optimal choice for the Dynamic /Streaming Viewer is the TGV—a user interface component that displays Items and associated properties in a grid.

This section compares the features of the TGV Model Browser of the Dynamic Viewer as well as the Streaming Viewer against the simple Model Browser of the Monolithic Viewer to make it clear how each function and what information each provides.



QD vs. AML



A TGV is populated with a response generated from the execution of an associated QD. In addition to the **CAD** Items that form the Query basis, administrators can add other related ItemTypes whose properties should be displayed in the TGV UI. Using TGV Definitions, administrators can choose which Items to display, what Properties to include, how to format the Property values, static text, what columns to include, and alternative icons. For example, the figure shows the left-most Tree column with nodes for the following Items that display the following properties for a given Item:

- **CAD**: A **CAD** Item **Name** and revision in square brackets.
- **Part**: A Part Item Name.

In addition, two columns were added for the `is_released` and `is_current` Boolean properties that both are displayed as checkboxes.

In contrast, the simple Model Browser content is determined by the execution of a static (hard-coded) AML query on the CAD structure of an opened **CAD** or **Part** Item. Each node in the Tree uses the `keyed_name` Property of the **CAD** Item with the icon for the **CAD** ItemType. There are no other ways to configure the display other than the configuration of the `keyed_name` Property.

Partial vs. Full CAD Structure display

By default, a TGV is populated using only a portion of QD response results. When related content has multiple levels, a user determines which content to query based on selected nodes in the Tree column. Expanding a node re-executes the query from this node and adds new rows to the TGV. On the contrary, the simple Model Browser uses an AML to execute an exhaustive (fully recursive) search for the entire CAD BOM structure. The Tree is populated with the full list of CAD Items from the query.

Each approach has a trade-off. For the TGV, query execution is generally faster, much faster for large results. However, after the simple Model Browser is populated, all data exists on the client and can be readily viewed. This affects data synchronization between the simple or TGV Model Browser and 3D View. For the Dynamic Viewer, when a 3D component is selected in the 3D View, a directed set of sub-queries is performed to populate the TGV with all levels of the hierarchy up to the selected part component. Once the data is populated, the query should not be re-executed, but the initial population can take several seconds.



In general, and especially for large and deep assembly hierarchies, the lazy-loading approach used by the TGV will yield a better user experience and require less network data exchange and less memory for the client browser. Also, the number of peer nodes and depth of query responses is configurable in the TGV Definition—administrators have control over the granularity of query response data.

