IBM Software Solutions | Enterprise Content Management Software

Customizing case client using property view designer

IBM Enterprise Content Management Software

IMPORTANT NOTICE

Product plans referenced in this document may change at any time at IBM's sole discretion based on market opportunities or other factors and are not intended to be a commitment to future product or feature availability in any way. All statements regarding IBM future direction or intent are subject to change or withdrawal without notice and represent goals and objectives only. All information is provided "AS IS", without warranty of any kind, express or implied. IBM will not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other documentation. References to IBM products do not imply that they will be available in all countries in which IBM operates.

Each IBM customer is responsible for ensuring its own compliance with legal requirements. It is the customer's sole responsibility to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM DOES NOT PROVIDE LEGAL ADVICE OR REPRESENT OR WARRANT THAT ITS SERVICES OR PRODUCTS WILL ENSURE THAT THE CUSTOMER IS IN COMPLIANCE WITH ANY LAW.

The information contained in this document:
- *Is intended to outline an aspect of IBM's general product direction and should not be relied on in making a purchasing decision;*
- *Is for informational purposes only and may not be incorporated into any contract; and*
- *Is not a commitment, promise or legal obligation to deliver any material, code or functionality.*

The development, release and timing of any features or functionality described for IBM products remain at IBM's sole discretion.

The Properties View Designer

Objectives

This lesson is designed to introduce you to the Properties View Designer feature.

- How property views are used
- Default, system-generated, or custom views
- Using the view editor

Properties Views

- The property view designer was introduced in ICM 5.2.
- It allows customized views of properties to be displayed in case pages
- Custom property views may in some cases replace the need for:
  - Forms
  - Custom Widgets

Properties Views

- A properties view is loaded and used by a properties widget, based on the case type, to know what properties to display and how to display them.
- Each case type has a “default view”. This is the properties view that will be used for this case type if a particular page's properties widget is not configured to use a particular custom view.
- There is a “system generated” view that is available automatically, and is automatically set as the initial default view, when designing a case type.

More about Properties Views

- Properties Views and Forms use a shared "controller" in a Model/View/Controller paradigm
  - The Model is the Model API
  - The View is what you create with the view editor
- Therefore, changing a property in a particular properties view may change the same property elsewhere:
  - Multiple properties views on the same page
  - Forms widgets on the page
  - Same property used in multiple places within a single properties view

Automation of view fields

- There is a Model API for the property controller layer, allowing accessing fields.
- It could be used by a script (Script Adaptor or custom widget), possibly also using External Data Services, to pre-populate fields.
- If you want a value to be displayed but not be a case property, use a Workflow field and populate it using a process that perhaps uses EDS to compute the value.

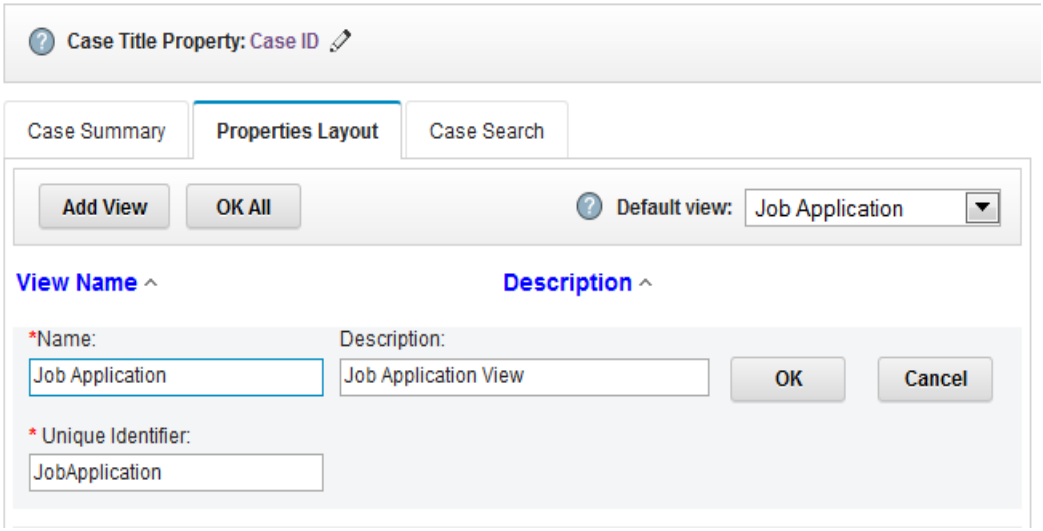
Automation of view fields (con't)

In addition, you can define a field to have a dependency on an External Data Service:

- Any change in the value of the field will be sent to the EDS
- That web service can return a change step that could update that field and/or other fields.
- The updates can also change metadata of fields, such as setting a field to be read-only/not, required/not, hidden/shown, etc.
- For example, the user chooses a state, and that populates a list of cities in that state in another combo box.

Creating a new Properties View

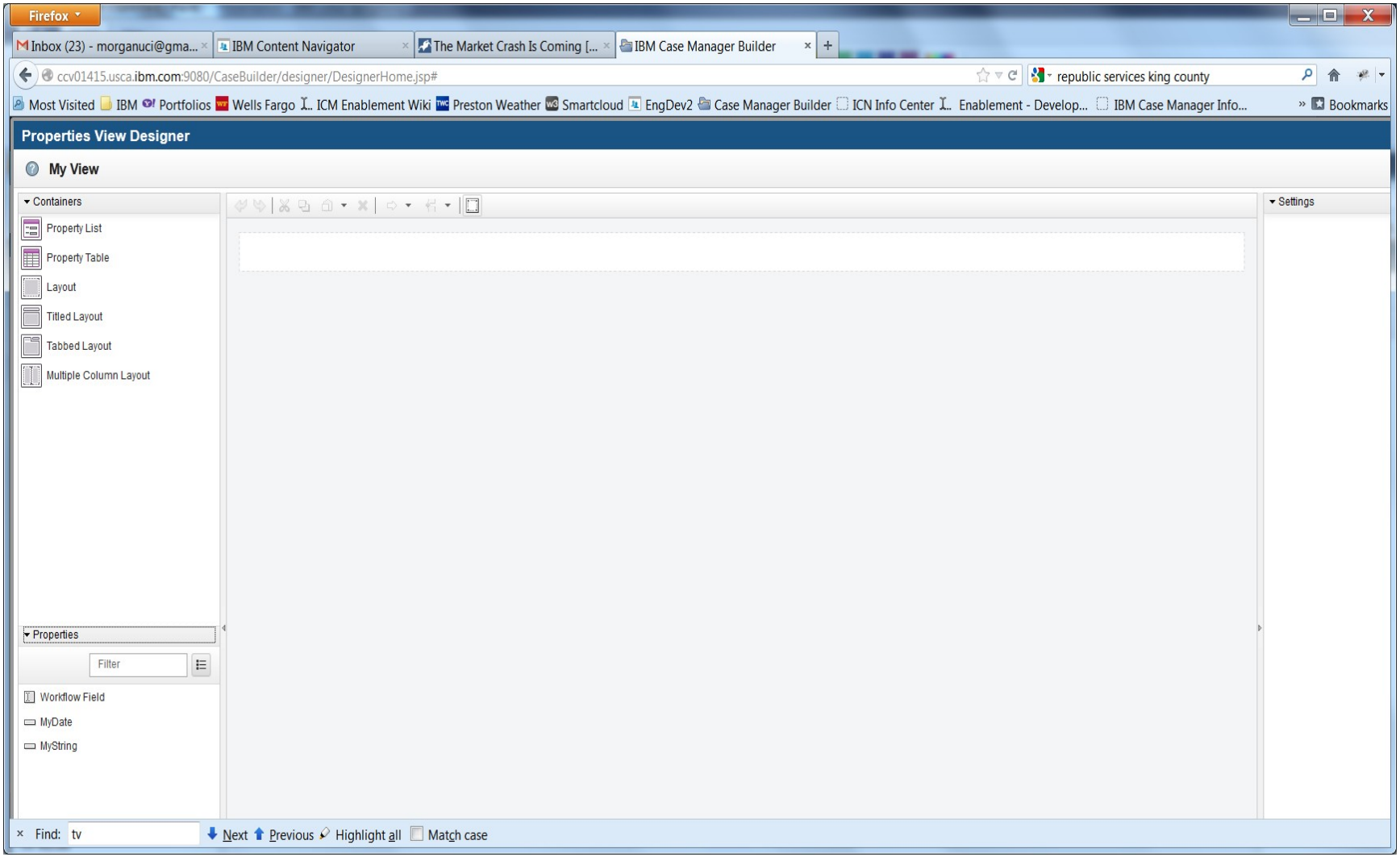
- You give a name to each custom properties view
- To create a new properties view, within Case Builder, open the case type, click on Views on the left, then the Properties View tab. Click Add View. Give the new view a name and description and click OK to save it.



Creating a new Properties View

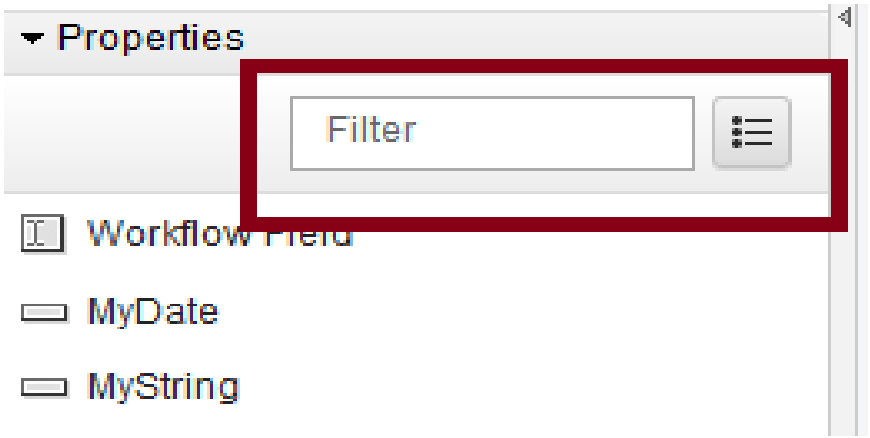
- Once you've created a new view, click on its name to open it in the editor.
- This opens a new screen with three regions:
  - Palettes (Containers and case properties) are listed on the left
  - A canvas area in the middle
  - Settings on the right
- These regions can be “undocked” and float
- One of the listed properties is “Workflow field”. It allows you to place a WF field just like other properties, rather than always listed at the bottom.
- See screen shot on next slide

Properties View Editor



More about Properties View Editor

- The list of properties may be very long. Therefore, there is a filter field. Typing text there limits the list to only properties whose names contain that text.
- Notice also the button to the right of the filter field. Pressing this button removes from the list any properties that have been used in the view so far.
- If that button is not selected, then properties that have been used are displayed in a lighter gray color.

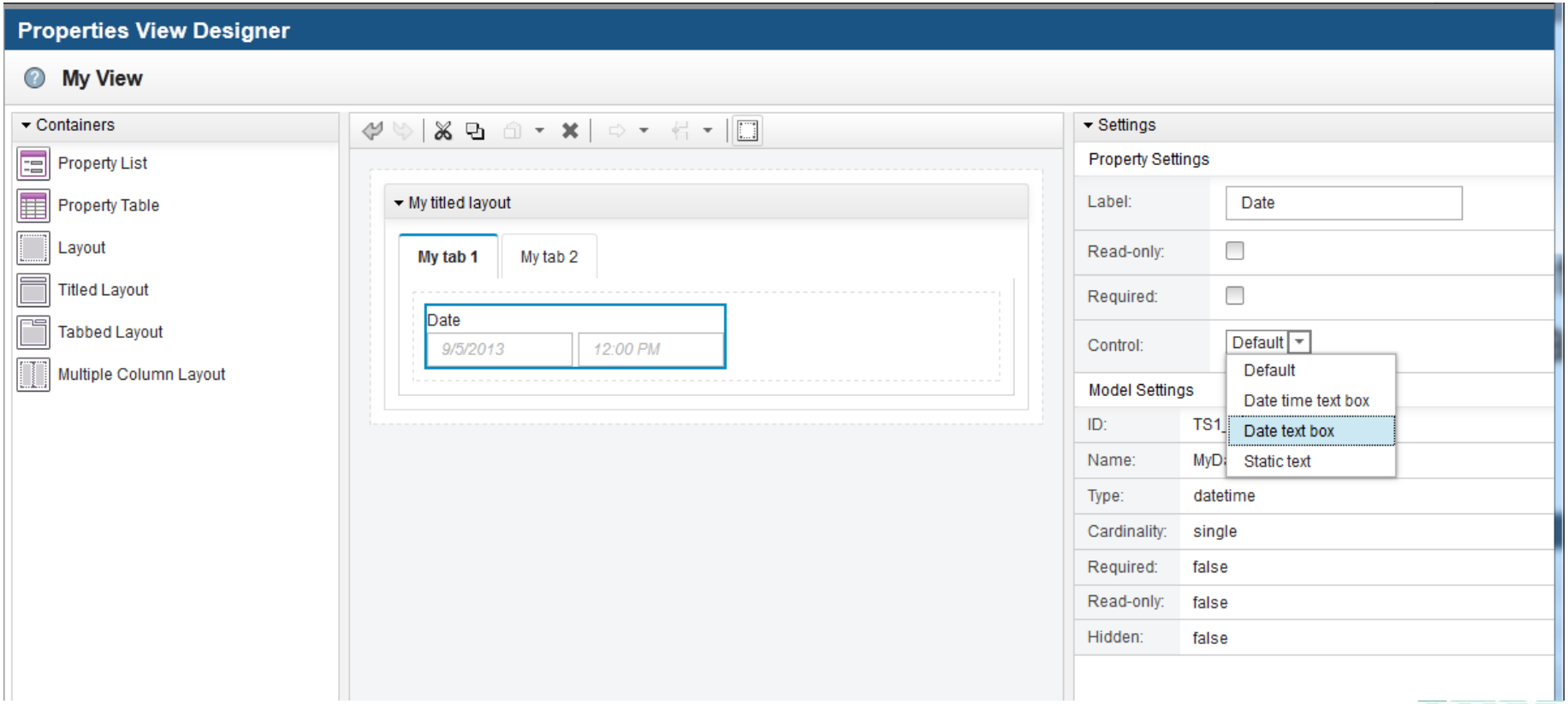


Creating a new Properties View

- The Containers are used to group properties, such as:
  - Lists
  - Tabbed views
  - Multicolumn layout
- Containers may contain other containers.
- The Property Table container is used for Multivalued Properties---the values are displayed vertically.
- Click on a container or property and drag it into the canvas.
- Control-click allows multiple properties to be selected. The order of selection determines their order on the canvas.

Property settings

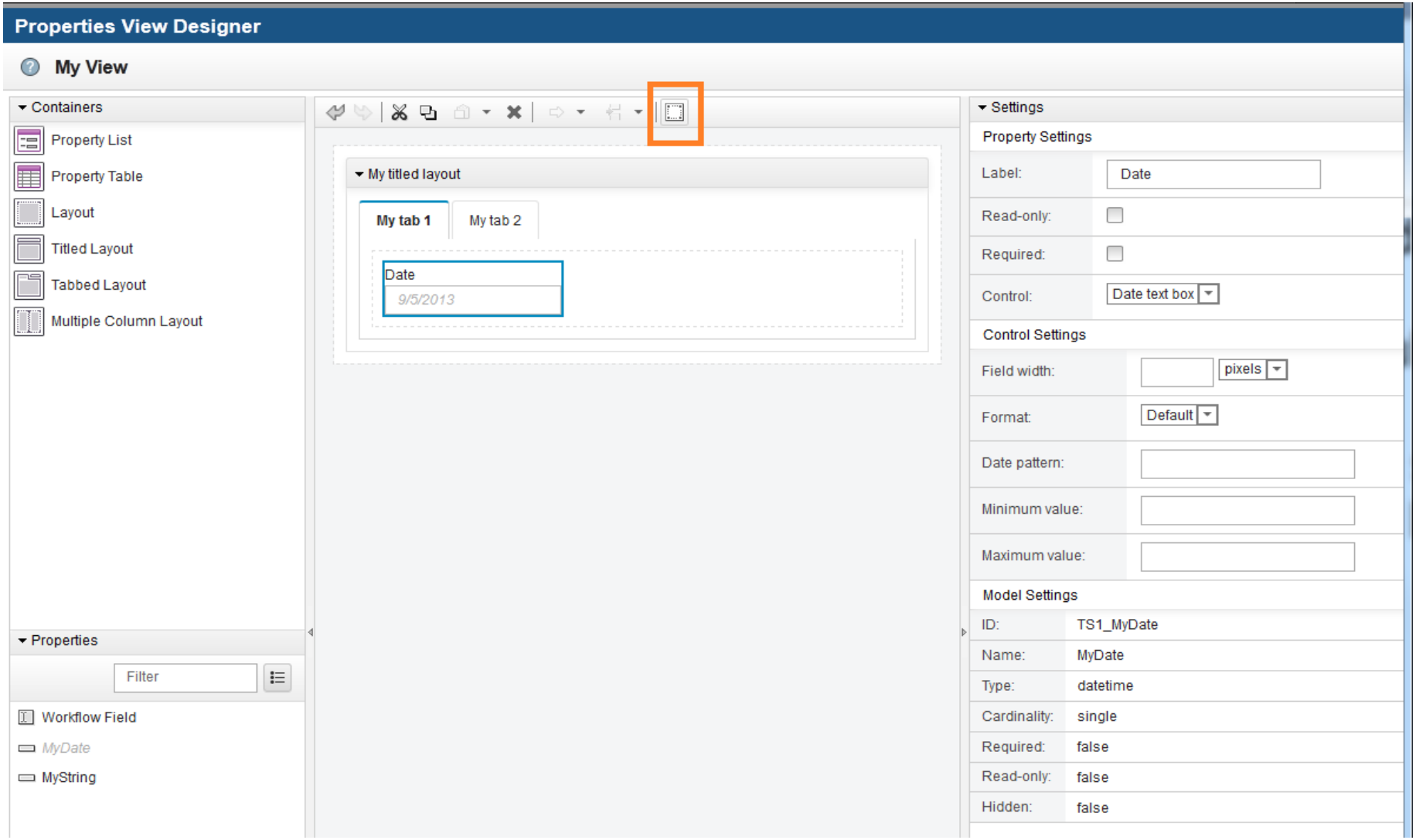
- The Settings area displays attributes of whatever item is currently selected in the canvas (a container or a property).



Property settings (con't)

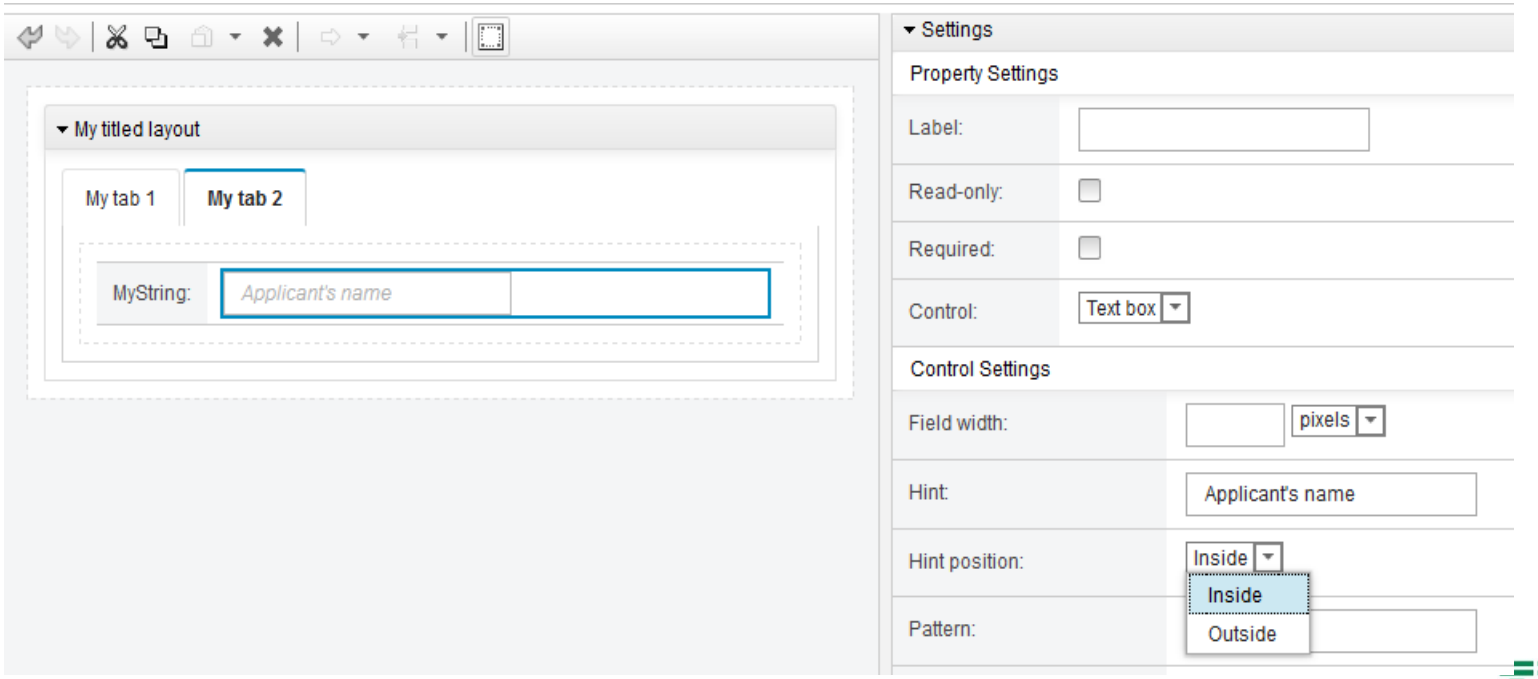
- The dotted lines on the screen indicate nested containers.
- This allows you to select a particular one to modify its settings, delete it, etc.
- There is a button at the top of the screen that toggles to show how the view will look at runtime (see next slide). Remember to turn this mode off after viewing the runtime appearance, as otherwise you can't edit the containers.
- Notice also that once the Date control is selected, the appearance changes and additional settings are displayed.

Property settings (con't)



Property settings (con't)

- “Hints” are like tooltips---they can suggest appropriate input values for fields.
- While the Date control generates its own, others allow you to input the hint string, and whether it's placed inside the field or below it.



Extending the properties view

- Extension points
  - Registry configuration – configures the available editors and the suitable mapping between data types and editors
    - Registry configuration is managed by RegistryManager. The module provides public APIs to merge customized registry configuration, enabling registering custom editors
  - Integration configuration – configures the controllers mapping and their attributes and methods
    - Integration configuration is managed by Integration.js. The module provides public APIs to merge customized integration configuration, enabling the use of custom controllers, custom data types, and customizing the attributes and methods of the controllers
- Approaches
  - Script Adapter widget
  - Extensions package

External Properties

- External properties are
  - Properties that define a collection of properties that is in its own collection and not associated with case properties or task properties
  - Meant to be defined and saved through an external source like a servlet, JSON file, or data service
  - Added to the Properties widget using script adapters
- External property binding is handled by the Properties area in the Model layer

External Properties (cont.)

- What is an external property?

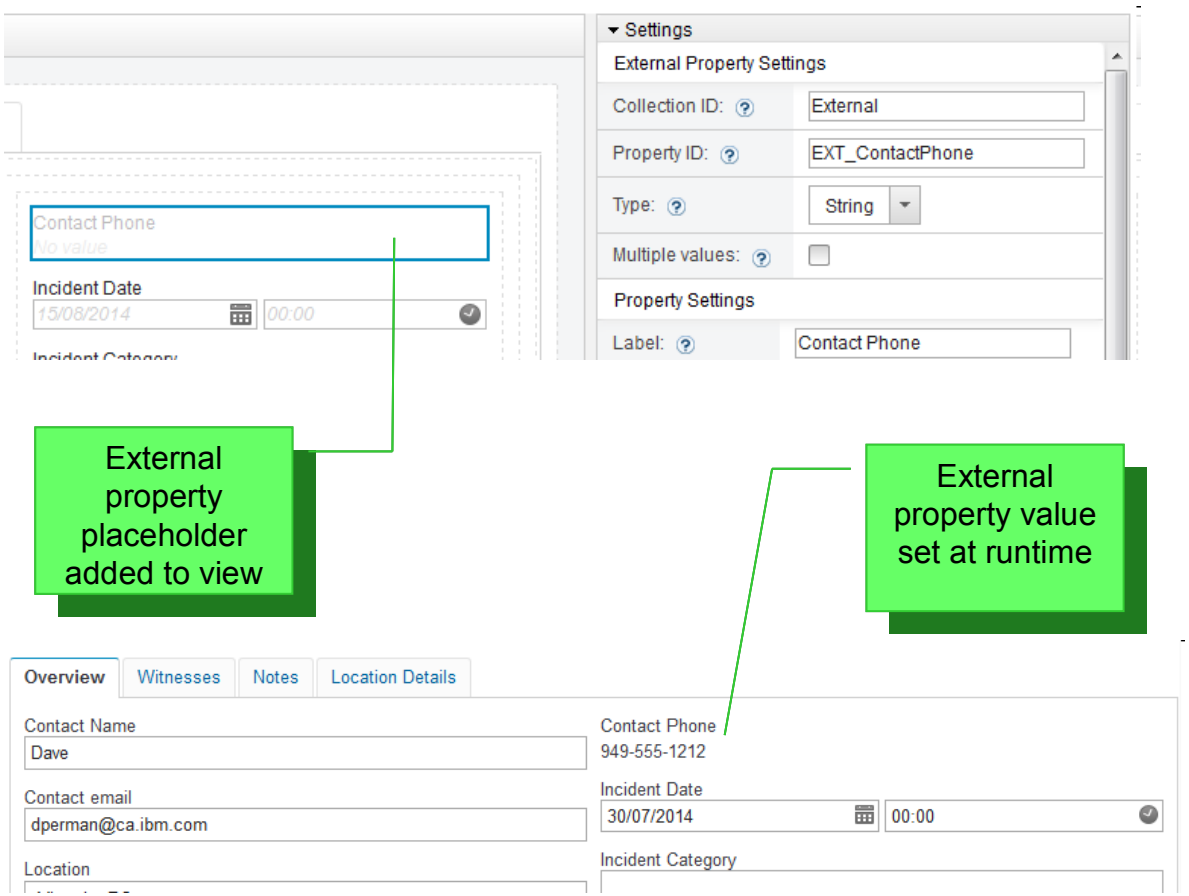
  An external property is similar to a case property or task property with the exception that an external property can be defined outside of IBM Case Manager and then bound to the Model API layer so that the data can be rendered in the Properties widget for example.
- External properties can be defined using
  - External Data Services
  - Java Servlets
  - JSON file on mid-tier
  - In-line in JavaScript

External Properties (cont.)

- General Use Case
  - *Most organizations maintain several "systems of record" data sources*
  - *These data sources often contain the most current "single source of truth"*
  - *Easy access to this information helps case workers complete their work*
  - *It is often necessary, and efficient, to access this data live and not store a copy*
- Specific Use Case
  - *While working a claim for a customer, an agent opens the case details*
  - *Along side the claim specific data, the agent sees the customer's contact information, pulled live from their client database*
  - *A list of past transactions is also conveniently displayed, saving the agent from having to access another system*

External Properties (cont.)

- Placeholder added to a property view
- A script is added to the page that gets data from another system based on one of the existing case properties
- The returned data is mixed into the view with the case properties
- The external property is never persisted with the case data itself and is always accurate
- External properties can be both read-only and read-write



Questions