

# Designer での コンポーネント指向のデザイン

## 目次

- コンポーネント指向のデザインとは何か ..... 2
- トップレベルコンポーネントを追加する ..... 3
  - トップレベルのメニューコンポーネントを作成する ..... 3
- 子コンポーネントを昇格する ..... 3
  - リンクされたインスタンスのクラスを選択する ..... 6
  - トップレベルコンポーネントの xtype とクラス名を設定する ..... 6
- トップレベルコンポーネントの再利用 ..... 7

## ☐ コンポーネント指向のデザインとは何か

Designer は UI を組み立てるのに標準の ExtJS コンポーネント構築部品を簡単に使えるようにします。単に Viewport とか Window といったコンテナをキャンバスにドラッグして、そこにコンポーネントを追加して始めます。これはスタートとしてはいい方法ですが、非常に複雑なアプリケーションの場合は、別々にデザイン、実装、保守、再利用できる小さなピースに分割したいと思うでしょう。このガイドはそれを Designer でどうやるかに焦点を当てています。ExtJS のソースコードの組織化や大きなアプリケーションの実装についての一般的な情報は、[Writing a Big Application in Ext](#) を参照してください。

Designer はコンポーネント指向デザインを手助けする次の二つのメカニズムを提供します。

- ◆ **トップレベルコンポーネント**としてプロジェクトにコンポーネントを明示的に追加できます。例えば、UI 上での各ページ用の、あるいはダイアログ用のトップレベルコンテナをプロジェクトに追加できます。
- ◆ **クラス昇格機能**を使って、子コンポーネントをトップレベルコンポーネントにすることができます。これにより、デザインが貯まってきたときにコンポーネントのリファクタリングや再利用が簡単になります。

Designer プロジェクトには好きなだけトップレベルコンポーネントを追加できます。プロジェクトをエクスポートしたら、それぞれのトップレベルコンポーネントごとに分割されたクラスファイルが生成されます。Designer は独自の用途にあらかじめ設定された Ext コンポーネントのクラスを生成し拡張するというモデルに従います。これらは生成された .ui.js ファイルに定義されたサブクラスで、Designer からエクスポートするたびに上書きされます。対応する .js ファイルでは、イベントハンドラとカスタムメソッドを実装することができるように、あらかじめ設定されたクラスを拡張しています。トップレベルコンポーネントの .js ファイルは最初にエクスポートしたときにだけ生成され、その後のエクスポートでは上書きされません。

トップレベルコンポーネントのコレクションとしての Designer プロジェクトを組織化することは、より管理しやすいコードを生成するのに加え、Designer 中での作業を簡単にします。大きく深くネストされたビューの場合、それらをキャンバスに描画するには少し時間がかかります。メインアプリケーションビューを他のトップレベルコンポーネントを使って構築すると、それらのコンポーネント上で個別に作業できます。この方法は、ある部品で作業しているだけなのに、全体が再描画されることを避けることができます。メインアプリケーションビューでリンクドインターフェースを使っている場合、必要とあらばすべての部品を簡単に表示することができます。

## ☰ トップレベルコンポーネントを追加する

トップレベルコンポーネントをプロジェクトに追加するには、次の方法があります。

- ◆ コンポーネントが選択されていない状態で Toolbox のコンポーネントをダブルクリックする。
- ◆ Toolbox からキャンバスにコンポーネントをドラッグする。
- ◆ Component メニューから New Component を選択する。

注：Window と Viewport はトップレベルコンポーネントとしてしか追加できません。

キャンバスに表示されるトップレベルコンポーネントを変更するには、Component タブで作業したいコンポーネントを選択するだけです。

## ☰ トップレベルのメニューコンポーネントを作成する

現在、Designer の Menu は Button あるいは他の Menu Item にしか割り当てられません。Menu は独立したトップレベルコンポーネントとしては生成できません。分割されたクラスとしてエクスポートできるトップレベルコンポーネントであるメニューを作成するには次のようにします。

1. Button を作成し Menu をそれに割り当てます。
2. Component タブの Menu を右クリックして **Promote to Class** を選択します。

Menu はトップレベルコンポーネントになり、Button コンポーネントはそのクラスへのリンクを保持することになります。プロジェクトをエクスポートすると、Designer は昇格した Menu コンポーネントを別の .js と .ui.js ファイルに生成します。

注：このテクニックはコンポーネントに割り当てられるようにデザインされた他のコンポーネント（Grid コラムのような）でも動作します。Component リストにあるアイテムは全部、クラスに昇格できます。

## ☰ 子コンポーネントを昇格する

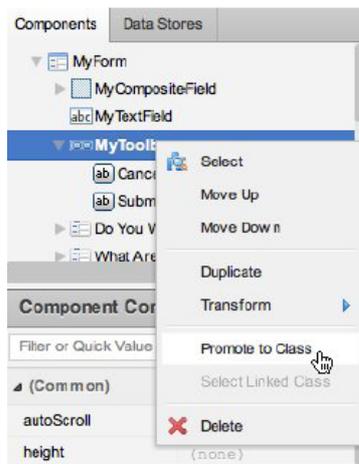
Component タブにリストされているすべてのコンポーネントは Promote to Class オプションによってトップレベルコンポーネントに昇格できます。子コンポーネントが昇格したら、その場所は昇格したクラスにリンクしたインスタンスが取って代わります。リンクされたインスタンスを直接変更したら、トップレベルコンポーネントの属性をオーバーライドします。すべてのリンクされたインスタンスに継承されるような変更はトップレベルコンポーネントで行います（トップレベルコンポーネントの追加インスタンスを生成する方法は「トップレベルコンポーネントの再利用」をご覧ください）。

注：子コンポーネントをキャンバス上の空のエリアにドラッグするとトップレベルコンポーネントを作ることができます。しかしながら、これはコンポーネントの移動であり、インスタンスを作るわけではありません。

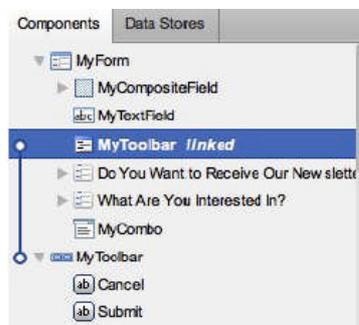
子コンポーネントを昇格させるには

- ◆ Component タブでコンポーネントを右クリックして、**Promote to Class** を選択します。

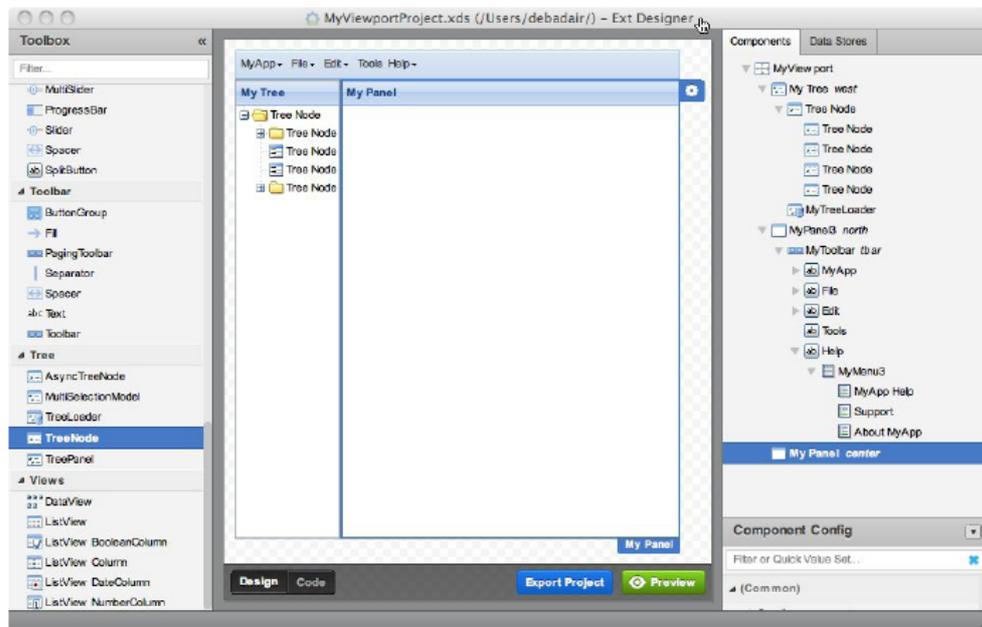
#### Designer でのコンポーネント指向のデザイン (4)



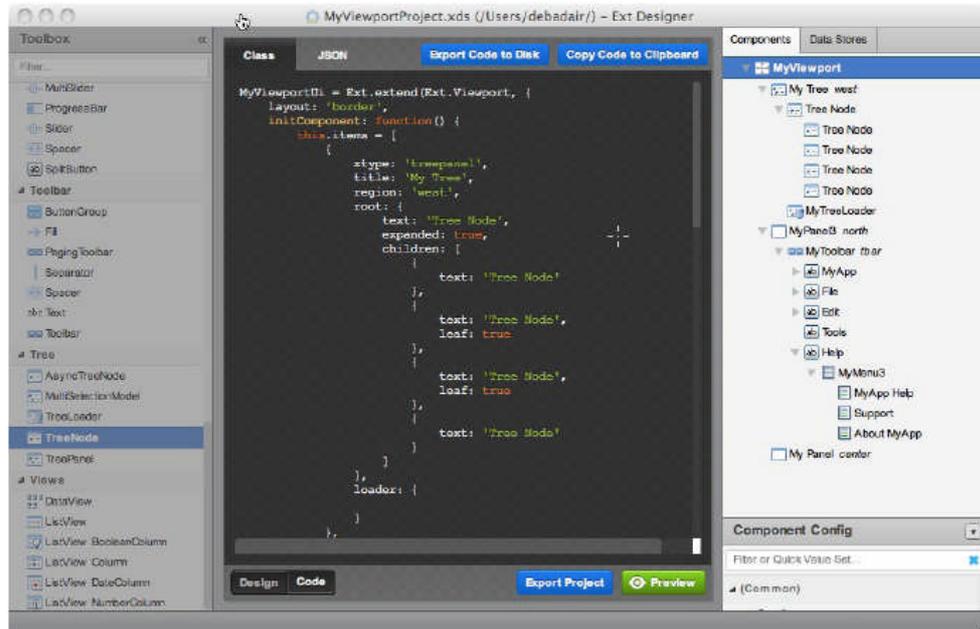
昇格したコンポーネントとリンクしたインスタンスが Component タブに表示されます。



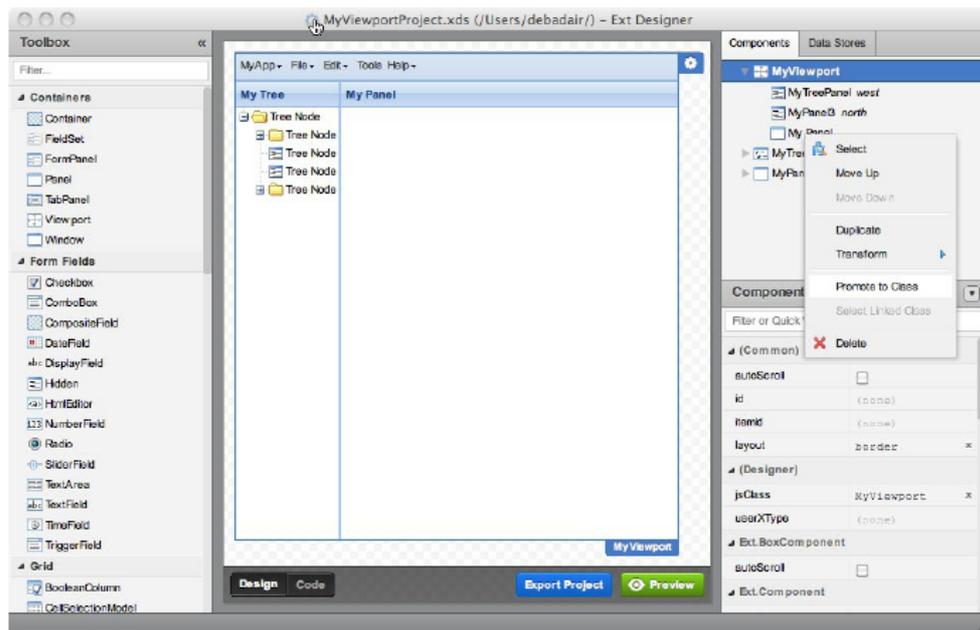
たとえば、Designer で新しい UI を構築し始めたとき、メインアプリケーションページの Viewport を最初に追加し、必要な部品を Viewport に追加します。



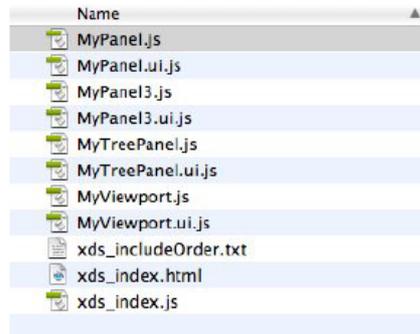
コードビューに切り替えると、Designer が Viewport とそのすべてのサブコンポーネントを 1 つのクラスで生成しようとしていることがわかります。



Designer に Viewport 上のメインコンポーネントをそれぞれ別のクラスファイルに生成させるには Promote to Class を使います。



これで Viewport のパネルを分割して作業できるようになり、プロジェクトをエクスポートするとパネルごとに別のクラスファイルが生成されます。

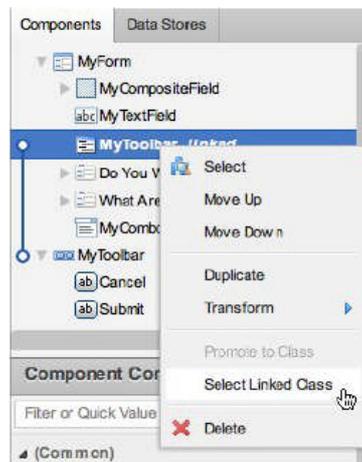


## リンクされたインスタンスのクラスを選択する

インスタンスの属性を変更するためにリンクされたインスタンスを選択することができます。トップレベルコンポーネントのすべてのインスタンスに変更を施す場合は、トップレベルコンポーネントを選択します。

リンクされたインスタンスに関連するクラスを選択するには

- ◆ リンクされたインスタンスを右クリックして **Select Linked Class** を選択します。リンクされたインスタンスをキャンバスまたは Component タブでダブルクリックしても選択できます。

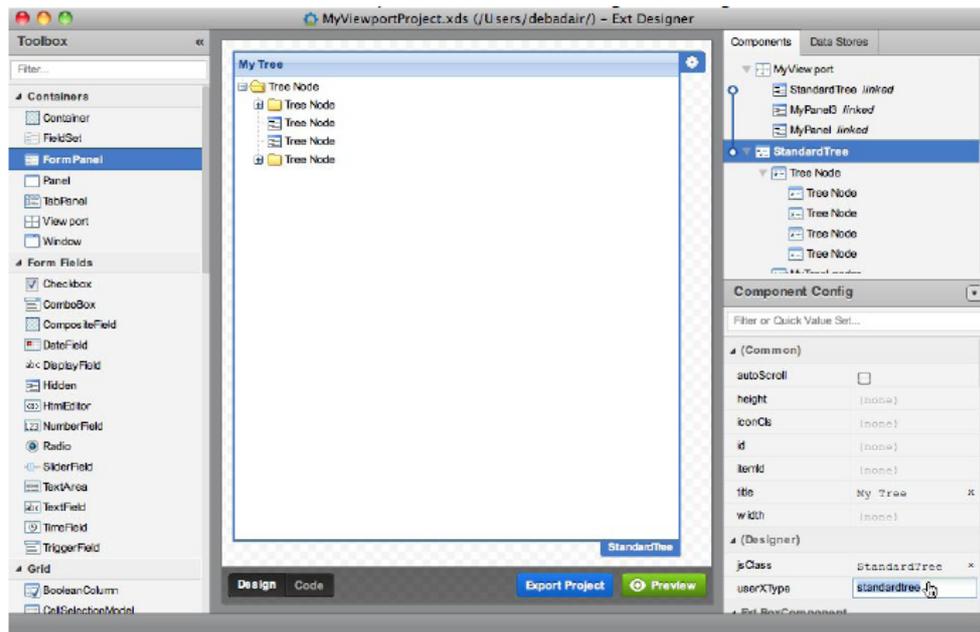


## トップレベルコンポーネントの xtype とクラス名を設定する

コンポーネントを昇格させると、新しいクラスに xtype とクラス名が自動的に生成されます。生成された値は使いたいものに変更でき、リンクされたインスタンスは新しい設定が使われるよう自動的に更新されます。

生成された値を変更するには

1. トップレベルコンポーネントを選択します。
2. Component Config インспекタで userXType と jsClass 属性を設定します。



## 🗨️ トップレベルコンポーネントの再利用

コンポーネントをクラスに昇格する時、リンクされたインスタンスはコンポーネントがあったところに自動的に作成されます。Designer ではすべてのトップレベルコンポーネントのインスタンスを明示的に作成する方法を提供しているので、コンポーネントの再利用も簡単にできます。

トップレベルのコンポーネントをコンテナにドラッグすると、次の 3 つのオプションが選択できます。

それを新しい場所に移動する。

そのコピーを新しい場所に作成する。

そのインスタンスを新しい場所に作成する。

例として、ダイアログが OK とキャンセルの標準的なツールバーを使う必要がある場合

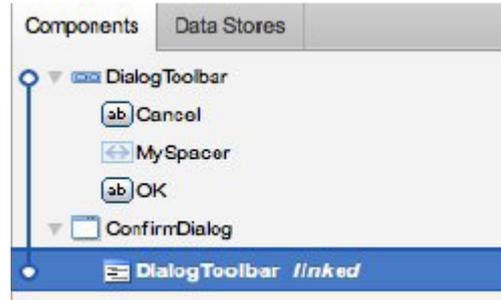
1. Components タブでツールバーをダイアログコンポーネントにドラッグします。



2. プロンプトが表示されたら、Copy Component ダイアログの Link を選択します。



Components タブに新しい子コンポーネントがインスタンスとして追加されました。



インスタンスのコピーを作成するには、同様に Duplicate を使うこともできます。コンポーネントの複製をつくる際は、単に右クリックして Duplicate を選択するだけです。