Loading Placeholder Data

Qt Design Studio supports views, models, and delegates, so that when you add a Grid View, List View, or Path View component, the ListModel and the delegate component are added automatically.

However, the missing context of the application presents a challenge. Specific models defined in C++ are the most obvious case. Often, the context is missing simple properties, which are either defined in C++, or in other component files. A typical example is a component that uses the properties of its parent, such as `parent.width`.

Using Dummy Models

If you open a file in the 2D view that references a C++ model, you see nothing in it. If the data in the model is fetched from the internet, you have no control over it. To get reliable data, *dummy data* was introduced.

For example, the following code snippet describes the file example.qml that contains a ListView that in turn specifies a C++ model:

```
ListView {
    model: dataModel
    delegate: ContactDelegate {
        name: name
    }
}
```

Create a directory named *dummydata* in the root directory of the project, so that it is not deployed to the device. In the `dummydata` directory, create a file (.qml) that has the same name as the value of `model`:

```
qml/exampleapp/example.qml
dummydata/dataModel.qml
```

Then create the dataModel.qml file that contains the dummy data:

```
import QtQuick 2.0

ListModel {
    ListElement {
```

```
        name: "Bella"
    }
    ListElement {
        name: "Corinna"
    }
}
```

Creating Dummy Context

The following example presents a common pattern:

```
Item {
    width: parent.width
    height: parent.height
}
```

This works nicely for applications but the **2D** view displays a zero-sized component. A parent for the opened file does not exist because the context is missing. To get around the missing context, the idea of a *dummy context* is introduced. If you place a file with the same name as the application (here, example.qml) in the `dummydata/context` directory, you can fake a parent context:

```
import QtQuick 2.0
import QmlDesigner 1.0

DummyContextObject {
    parent: Item {
        width: 640
        height: 300
    }
}
```

‹ Simulating Complex Experiences

Simulating Application Logic ›