LoanSphere Design Guide

Wednesday, August 05, 2015 10:14 AM

LoanSphere Design Guide
7/6/2015

Welcome!
The purpose of this website is to provide guidance for designing and implementing features into the LoanSphere (formerly "Fusion") platform.

Table of Contents

Inserted from <http://localhost:57964/Home/>

LoanSphere Design Principles

Wednesday, August 05, 2015 10:14 AM

LoanSphere Design Guide

7/27/2015

Design Principles

Don't make users think; let experts dig deeper.

1. Progressive Disclosure

   LoanSphere presents only the minimum data required to complete a task but makes it easy explore

   The initial presentation of minimum data to a user allows for increased learnability and decreased cognitive load while still allowing for the completion of tasks.

2. Real-World Models

   LoanSphere content matches real world mental-models

   "Ease of implementation" is secondary to making the software match a real-world model – from Database to UI.

3. Context Aware

   LoanSphere intelligently provides relevant information in context of the user's current task

   Using relationships inherently defined by the "Real-World Models" principle, LoanSphere selectively shows data that is relevant to a user's task.

4. Workflow Driven, but Not Limited

   LoanSphere guides users through workflows but also allows them to access data outside of a workflow

   The usage of workflows ensures that users are guided through the complex world of rules and regulations, but we don't limit access to data outside of a driven workflow. Every screen can also be accessed through Navigation.

5. Consistency

   LoanSphere provides consistent paradigms.

   By having consisent interactions and models, the learning curve for new users is significantly lowered.

Inserted from <http://localhost:57964/Principles/>

LoanSphere Standard Patterns

Wednesday, August 05, 2015 10:09 AM

LoanSphere Design Guide
7/27/2015

Standard Patterns
The purpose of this page is to outline the general patterns that LoanSphere uses to problems based on LoanSphere Design Principles.

1. General Layout
    1. Minimum user effort for the most vital data.
        - Prioritize field groupings by order of importance, with the most important groupings on top and left.
        - Use the BorderGroupExpander control whenever possible – making sure to take advantage of the "Collapsible Content" feature.
    2. Related fields are grouped inside of container controls, hieratically.
        - Use a BorderGroupExpander when extensive data can be hidden/collapsed
        - Use a BorderGroupExpander when in doubt (you can always hide the expansion button)
        - Use a GroupBox for small, supporting data sections that are nested within BorderGroupExpanders
        - Exception: When dealing with the smallest of subgroups, use a liberal amount of whitespace instead of a ''container'' control.
    3. Calculated totals line-up vertically below their respective equation terms.
        - All numeric values should be right-aligned.
        - Calculated total values should be visually distinguishable from the other fields.
    4. All clickable regions (buttons, links) use blue-themed coloring.

2. Labels
    1. Fields are labeled according to their respective group contexts.
        - Do not repeat the group name in the field label.
    2. When labeling a date, describe the event it signifies rather than appending ''Date''.
        - e.g. ''Approved'' > ''Approval Date''

3. Wizards
    1. Only display fields typically needed to complete the step in the task.
        - Rule of Thumb: Only show fields that are needed at least 80% of the time.

- For less frequently-used fields, "one-off" solutions should be provided such as links or pop-ups.

BorderGroupExpander - UI Library - LoanSphere Design Guide

Wednesday, August 05, 2015 10:09 AM

LoanSphere Design Guide
7/16/2015

UI Library

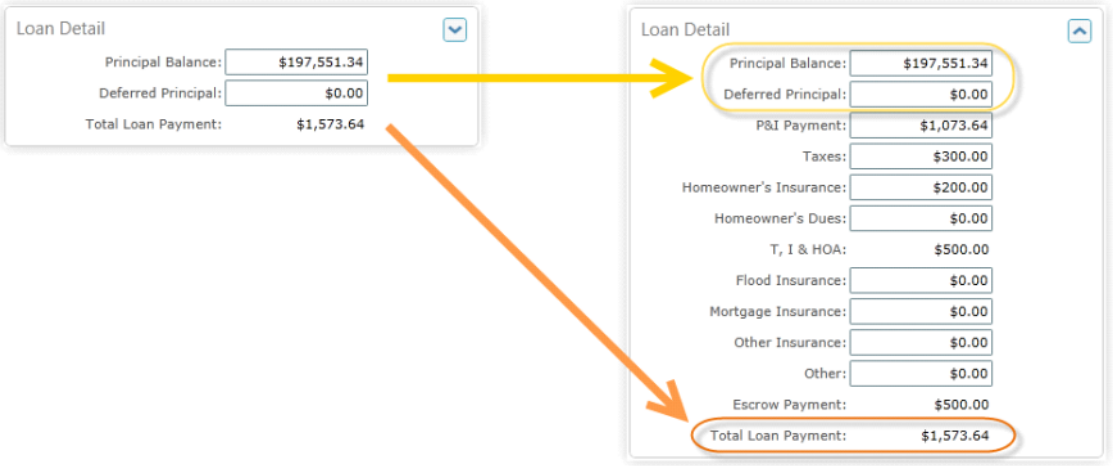
BorderGroupExpander

The BorderGroupExpander control is the most versatile "contaner" control in the platform. It is the preferred control when grouping information.
This control allows for expanding and collapsing the contents to increase/reduce detail. It is also possible to show content when in case the fields should be displayed in a different logical order when other fields are present.

Intended Use

A container that can be collapsed, used for a grouping of fields.

- Screenshot(s)
- Properties
- Sample Code



Name	Type	Description
Header	System.String	This text appears at the top-left that describes the fields in the control.
HideButton	System.Boolean	This flag determines whether the expander button is hiddened; Used for cases that the collapsing feature is not required.
BeforeCollapsibleContent	System.Windows.FrameworkElement	Use this XAML tag to show elements ***above*** the defined Content when the control is both collapsed and expanded.
AfterCollapsibleContent	System.Windows.FrameworkElement	Use this XAML tag to show elements ***below*** the defined Content when the control is both collapsed and expanded.

```
<controls:BorderGroupExpander x:Class="Dri.Tds.UI.FormParts.PaymentDetail"
                              xmlns="http://schemas.microsoft.com/client/2007"
                              xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
                              xmlns:controls="clr-namespace:Dri.Tds.UI.Controls;assembly=Dri.Tds.UI.Controls"
                              HorizontalAlignmen="Left"
                              Header="Loan Detail">
        <controls:BorderGroupExpander.BeforeCollapsibleContent>
                <controls:DataFieldPanel LabelWidth="175" FieldLabelAlignment="Right" Width="300">
                        <controls:DataField DataFieldName="PrincipalBalance" />
                        <controls:DataField DataFieldName="DeferredPrincipal" DataObjectName="LoanData" />
                </controls:DataFieldPanel>
        </controls:BorderGroupExpander.BeforeCollapsibleContent>
        <controls:BorderGroupExpander.Content>
                <controls:DataFieldPanel LabelWidth="175" FieldLabelAlignment="Right" Width="300">
                        <controls:DataField DataFieldName="MonthlyPaymentPrincipalInterest" LabelText="P&I
Payment" />
                        <controls:DataField DataFieldName="MonthlyPaymentTax" />
                        ...
                </controls:DataFieldPanel>
        </controls:BorderGroupExpander.Content>
        <controls:BorderGroupExpander.AfterCollapsibleContent>
                <controls:DataFieldPanel LabelWidth="175" FieldLabelAlignment="Right" Width="300">
                        <controls:DataField DataFieldName="MonthlyPayment" LabelText="Total Loan Payment"
IsReadOnly="True" />
                </controls:DataFieldPanel>
        </controls:BorderGroupExpander.AfterCollapsibleContent>
</controls:BorderGroupExpander>
```


Inserted from <http://localhost:57964/Library/Control?controlName=BorderGroupExpander>

MasterDetailsGrid - UI Library - LoanSphere Design Guide

Wednesday, August 05, 2015 10:09 AM

LoanSphere Design Guide
7/10/2015

UI Library

MasterDetailsGrid

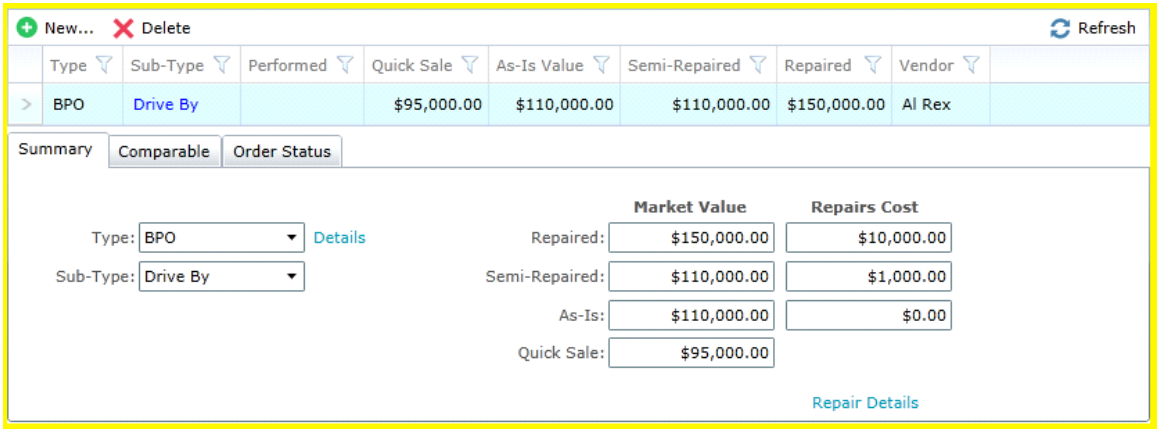
The "Master-Details Grid" control is essentially a DataGrid with many expanded features. Some examples of the features this control can express are:

- showing a detail panel below the grid that is associated to the selected row
- navigating the user to a different tab or pop-up
- handling adding/deleting records

Intended Use

This control is used to displaying multiple records and represents a one-to-many relationship.

- Screenshot(s)
- Properties
- Sample Code



Name	Type	Description
AutoSelectFirstItem	System.Boolean	Determines whether the first row of the grid is selected.
CanUserDeleteRows	System.Boolean	Determines whether the "Delete" button displays.
CanUserInsertRows	System.Boolean	Determines whether the "New" button displays.
DetailsControlName	System.String	The name of the control (usually a FormPart).
DetailsHorizontalAlignment	System.Windows.HorizontalAlignment	The alignment of the details pane within the context of the MasterDetailsGrid control.
DetailsLengthPe	System.Int32	The percentage of the entire control height that the details pane will

rcent		occupy.
DetailsScrollBarVisibility	System.Windows.Controls.ScrollBarVisibility	Determines whether or not (and if so, how) the scroll bars appear for the details panel.
FrozenColumnCount	System.Int32	The number of columns that stay "frozen" as the user scrolls horizontally.
GridMaxHeight	System.Double	The maximum height that the grid will render as and no larger. (Possible to to render shorter than this depending on a number of factors.)
GridMaxWidth	System.Double	The maximum width that the grid will render as and no larger. (Possible to to render narrower than this depending on a number of factors.)
GridMinHeight	System.Double	The minimum height that the grid will render as and no smaller. (Possible to to render taller than this depending on a number of factors.)
GridMinWidth	System.Double	The minimum width that the grid will render as and no smaller. (Possible to to render wider than this depending on a number of factors.)
IsReadOnly	System.Boolean	Determines whether grid cells are editable.
ItemTypeName	System.String	The name of the Entity type that will be displayed.
ItemsSourcePath	System.String	The path of the grid's items source.
Mode	Dri.Tds.UI.Controls.MasterDetailsGridMode	**Basic** Essentially renders the control as a DataGrid with no "fancy" features. **Detail** Displays a detail panel below the grid that represents the selected item. (most common usage) **Navigate** Same as "Basic" with the added ability to open detailed view of the selected item - either with a double-click or by clicking the "View" button on top. **NavigateWithDetail** Combines "Detail" and "Navigate" modes defined above.

```
<controls:MasterDetailsGrid x:Name="mdgValuationsList"
                            AutoSelectFirstItem="True"
                            CanUserDeleteRows="True"
                            CanUserInsertRows="True"
                            ForceDetailsReload="True"
                            IsReadOnly="True"
                            ItemTypeName="Valuation"
                            ItemsSourcePath="Valuations"
                            DetailsControlName="FormParts.ValuationDetailsTabs"
                            Mode="Detail"
                            Width="825"
                            Margin="10,10,0,0"  >
<controls:MasterDetailsGrid.SortDescriptors>
        <telerik:SortDescriptor Member="PerformedDate" SortDirection="Descending" />
                </controls:MasterDetailsGrid.SortDescriptors>
        </controls:MasterDetailsGrid>
```

Inserted from <http://localhost:57964/Library/Control?controlName=MasterDetailsGrid>

DataField - UI Library - LoanSphere Design Guide

Wednesday, August 05, 2015 10:09 AM

LoanSphere Design Guide
7/27/2015

UI Library

DataField

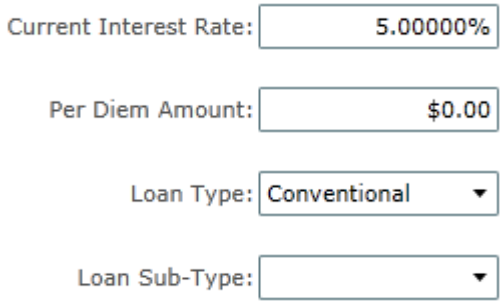
The DataField is a "smart" control that renders based on the type of data it represents from the BusinessObject's MetaData.
The screenshot below shows how the same XAML control (DataField) is rendering as different user-interaction controls.
The usage of DataFields is vital to the integrity of the framework because it provides the following features:

- Field-Level Permissions
- User-Adjustable Field Data Tool-Tips
- System-Wide Consistent Field Labels
- Common Look & Feel

Intended Use

This control frees developers from having to determine how a field should be rendered.

- Screenshot(s)
- Properties
- Sample Code



Name	Type	Description
DataFieldName	System.String	The name of the field (from the MetaData) that you want represented in the UI.
DataObjectName	System.String	The name of the BusinessObject. Used when the field's BusinessObject is not the data context of the container.
LabelText	System.String	This text overrides the label text defined in the MetaData. (This property should be used sparingly.)
LabelWidth	System.Double	The width of the label for the field. (Tip: ensure this width exceeds the width of the largest label text to avoid truncation)

```
                    <controls:DataField DataFieldName="InterestRate" LabelText="Current Interest Rate" />
<controls:DataField DataFieldName="PerDiemInterest" DataObjectName="LoanData" />
<controls:DataField DataFieldName="LoanTypeId" />
<controls:DataField DataFieldName="LoanSubTypeId" Filter="{Binding Path=LoanSubTypeFilter}" />
```