



Autodesk Vault 2023

Data Standard – Updates | Migration

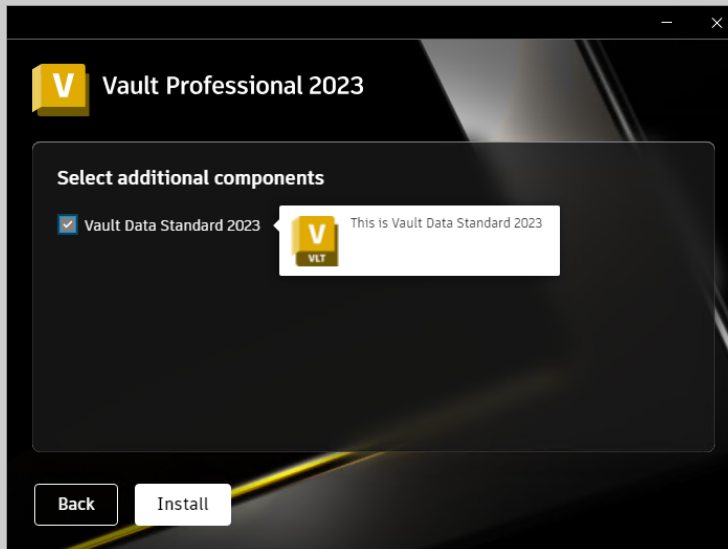
What's New | How to update customized configurations

Markus Koechl
Solutions Engineer PDM|PLM | [Community Profile](#)

Agenda

Autodesk Vault 2023 Data Standard – Updates|Migration

- What's New 2023 – Enhancements and changes
 - ***Vault UI Themes Support (2023 Update 2)***
 - General Enhancements | Changes VDS Inventor
- Editing custom configurations
 - VDS-Configuration-Editor-Basic
 - VDS-Sample-Configuration-2023
- Migrating custom configurations 2021/2022 → 2023.2
 - Minimum Updates
 - Recommended Updates
 - ***Enabling Themes***





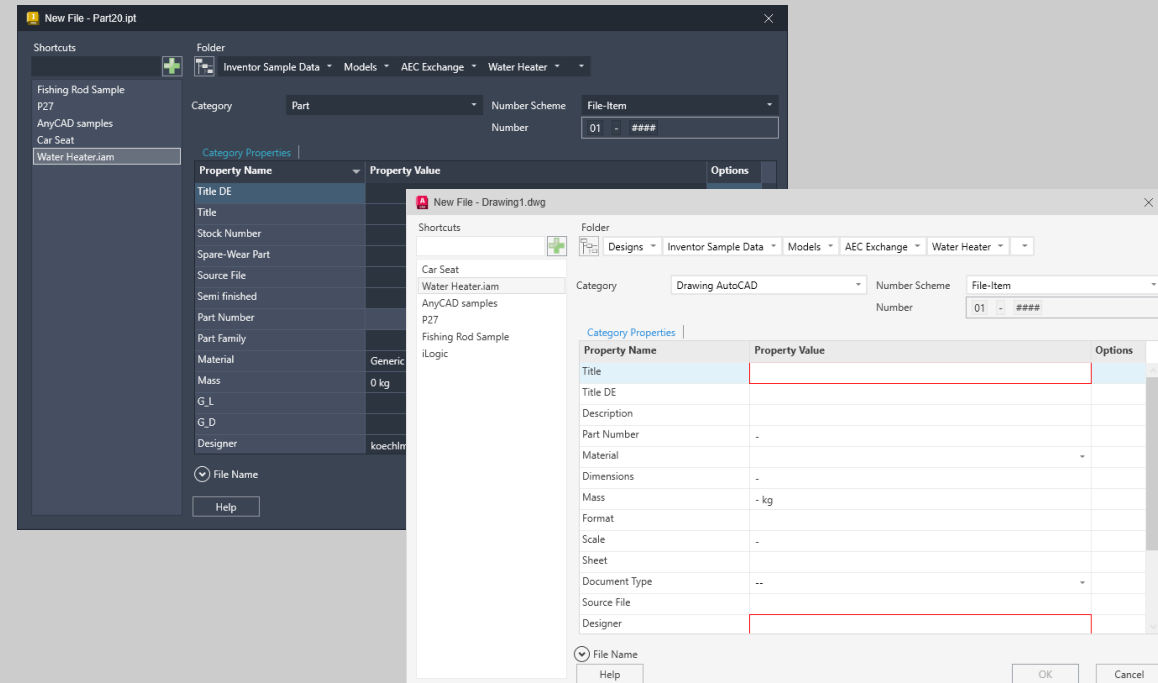
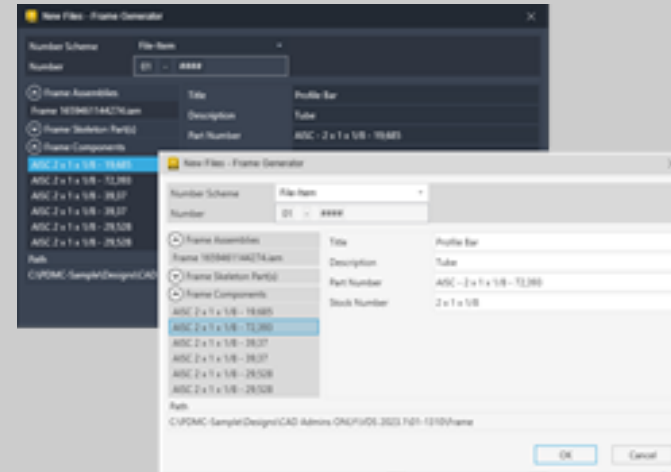
Vault Data Standard 2023 Update 2

Themes Support

Themes Support

Classic | Light | Dark

- VDS 2023.1 started supporting themes; update 2 completed and enhanced it.
 - Datasheets and dialogs adopt the hosting application's theme
 - Vault Explorer – Classic | Light | Dark
 - AutoCAD | Inventor – Light | Dark
 - Legacy customized dialogs continue using the “Classic” theme (or a mix) until the definition (XAML) incorporates the themes support updates
 - Go To the new chapter, [“Enabling Themes”](#)

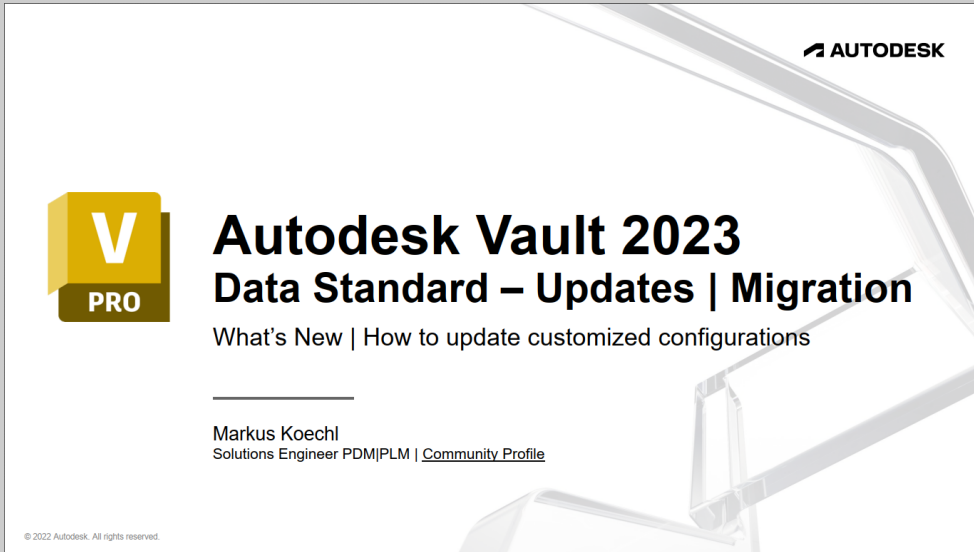


Configuration Sample PDMC-Sample – dark and light themes

Vault Data Standard

XAML Styles 2023.2

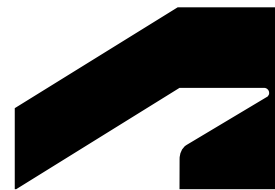
- Removed legacy style overrides from the default configuration
 - Style and individual control definitions
 - ⇒ Continue to follow the step-by-step instructions to remove legacy style overrides from customized configurations
- Shares all control type styles as a default
 - ⇒ Create new custom XAML files by copying the latest default templates or copy the style references as instructed in the step-by-step instructions from latest default templates



The slide features the Autodesk logo in the top right corner. On the left, there is a yellow square with a white 'V' and a black square with 'PRO' in white. The main title is 'Autodesk Vault 2023 Data Standard – Updates | Migration' in bold black text. Below the title is the subtitle 'What's New | How to update customized configurations'. At the bottom left, the author's name 'Markus Koechl' and title 'Solutions Engineer PDM/PLM | [Community Profile](#)' are listed. A small copyright notice '© 2022 Autodesk. All rights reserved.' is at the bottom left.

Migration with step-by-step instructions

```
<Style TargetType="{x:Type Label}" BasedOn="{StaticResource ThemedLabelStyle}">
  <Setter Property="Margin" Value="0,0,0,0" />
  <Setter Property="Height" Value="24" />
  <Setter Property="Padding" Value="5,3,3,2" />
  <Setter Property="VerticalAlignment" Value="Stretch" />
  <Setter Property="HorizontalAlignment" Value="Stretch" />
  <Setter Property="BorderBrush" Value="■" #FFAB9B3" />
  <Setter Property="BorderThickness" Value="1,1,1,0" />
  <Style.Triggers>
    <DataTrigger Binding="{Binding Content, RelativeSource={RelativeSource Self}}" Value="{x:Null}">
      <Setter Property="BorderThickness" Value="1" />
      <Setter Property="BorderBrush" Value="■" #ff0000" />
    </DataTrigger>
  </Style.Triggers>
</Style>
```



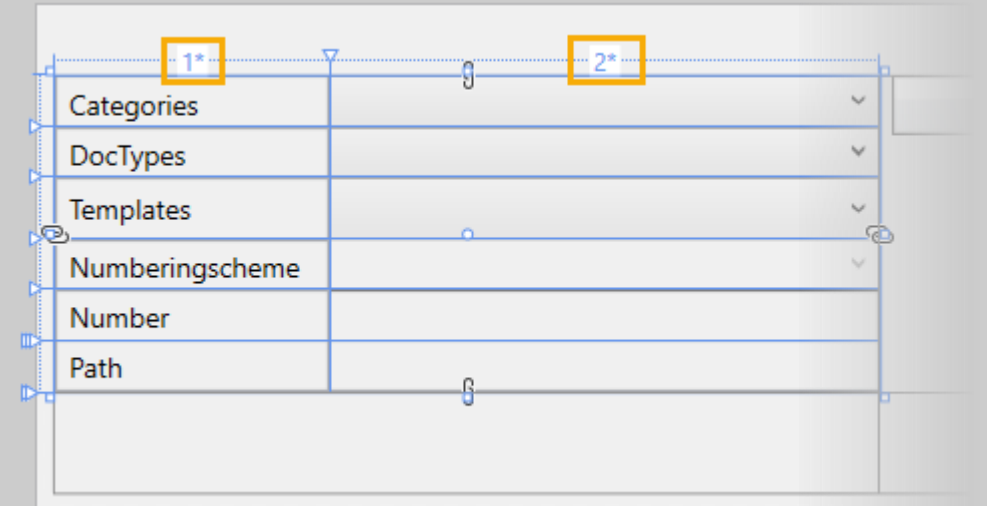
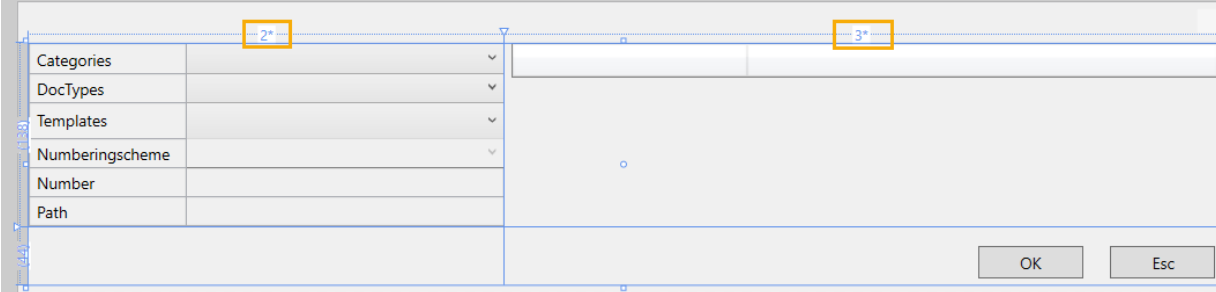
Vault Data Standard 2023

What's New – Enhancements and changes

Data Standard

General Enhancements

- Dynamic sizing behavior of all XAML dialogs* for CAD and Vault
 - Improved user experience for large label or field texts

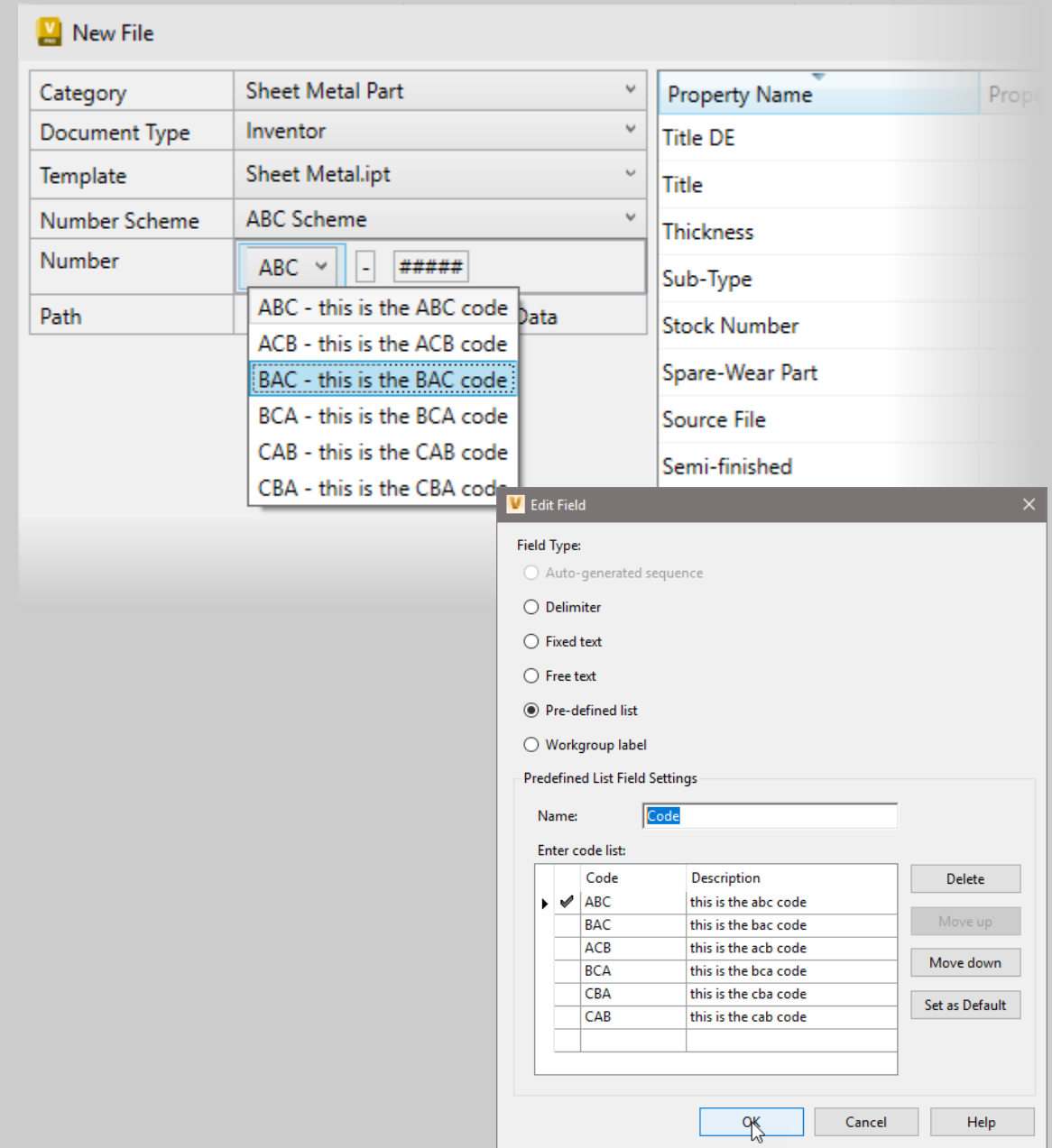


*Note – Dynamic sizing of datasheets in Vault Explorer detail tabs is nominated for future release(s)

VDS Numbering Control

List Value Descriptions

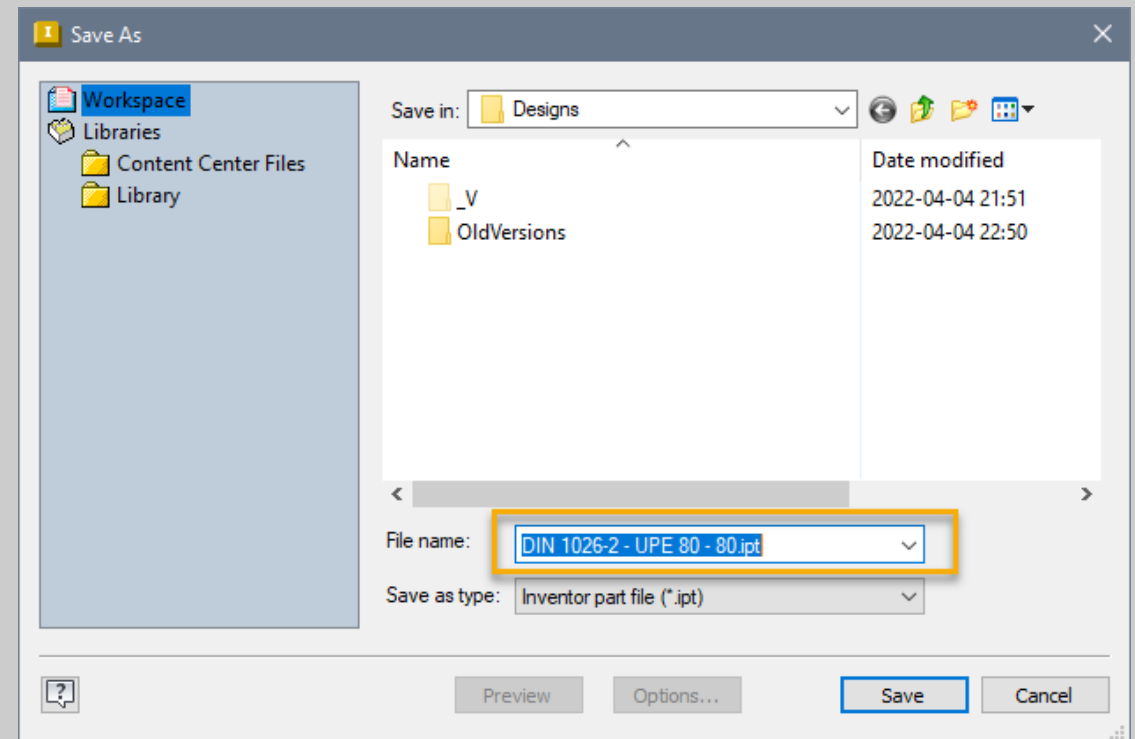
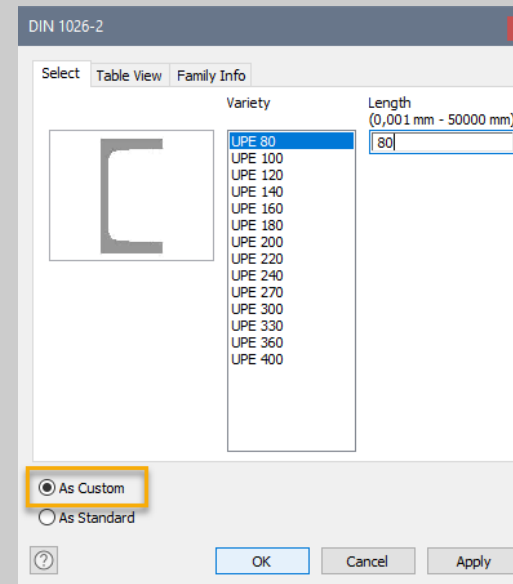
- **Change** – The VDS Numbering Control supports description display of Pre-defined list codes
 - Inventor (2022.1)
 - AutoCAD, Vault (2023)
- **Benefit** – Users can select codes by reviewing their full description as configured in the Vault numbering scheme definition



Data Standard

Inventor

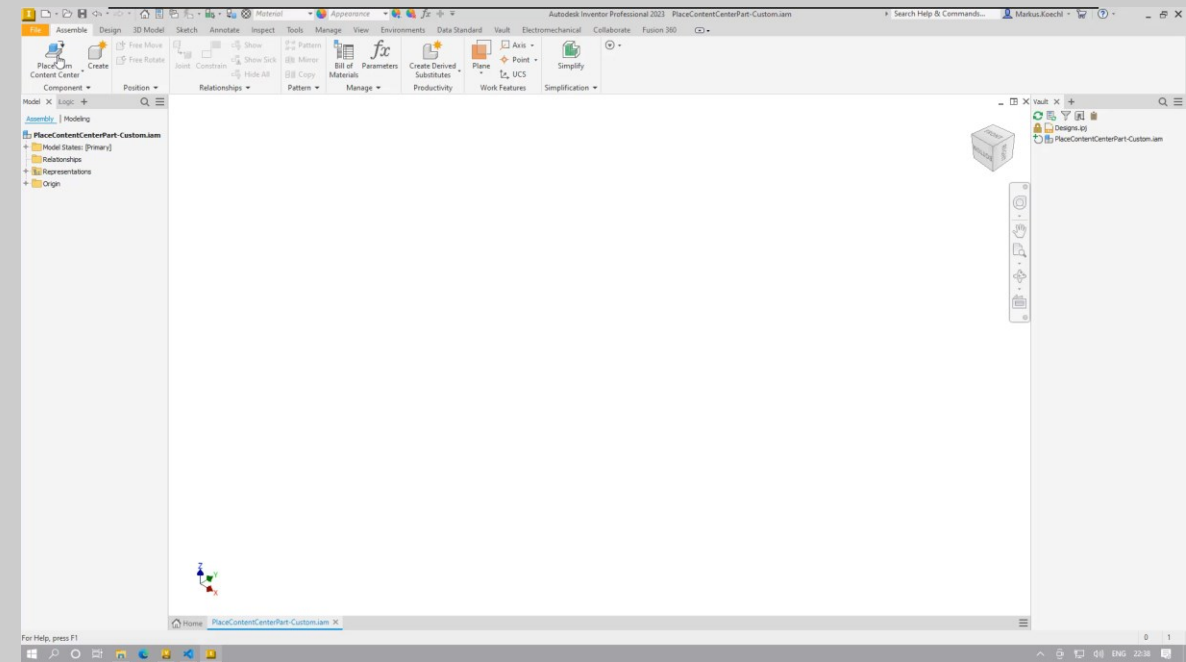
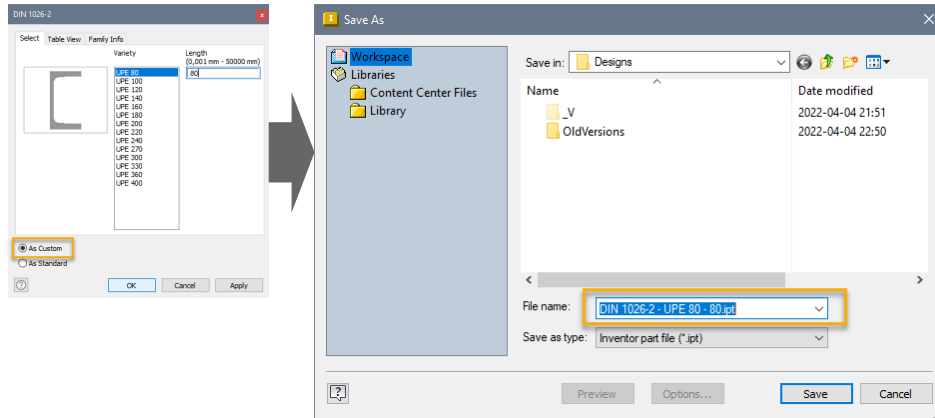
- Place from Content Center – Custom
 - VDS 2021.2.2 | VDS 2022.2 | VDS2023 changed the default behavior
 - No file numbering
 - No VDS interaction
- ⇒ Save custom CC component to project path and don't touch metadata (completed by source library)



Demo Video

New default workflow

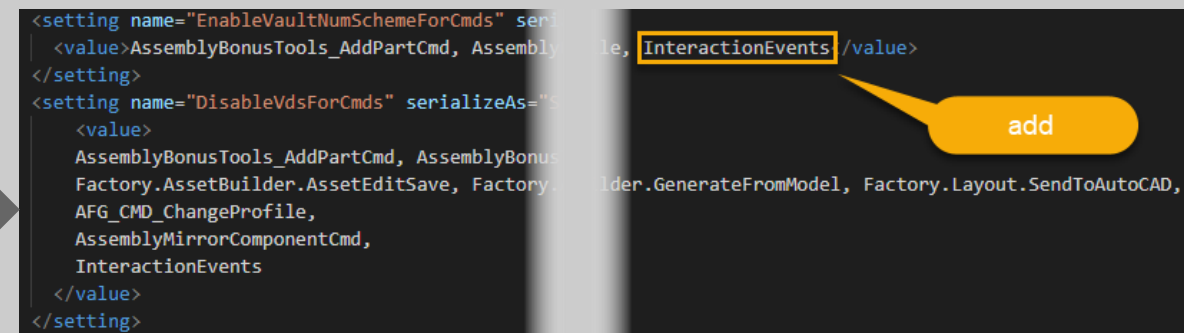
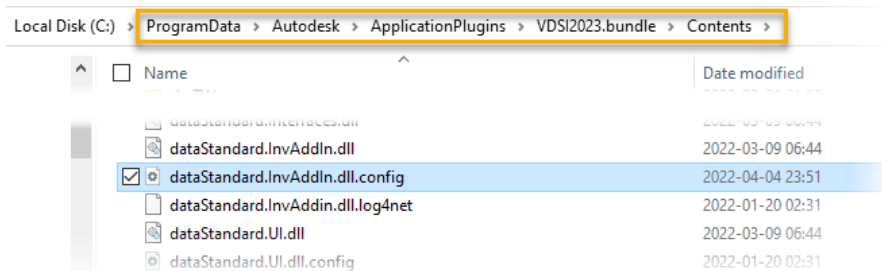
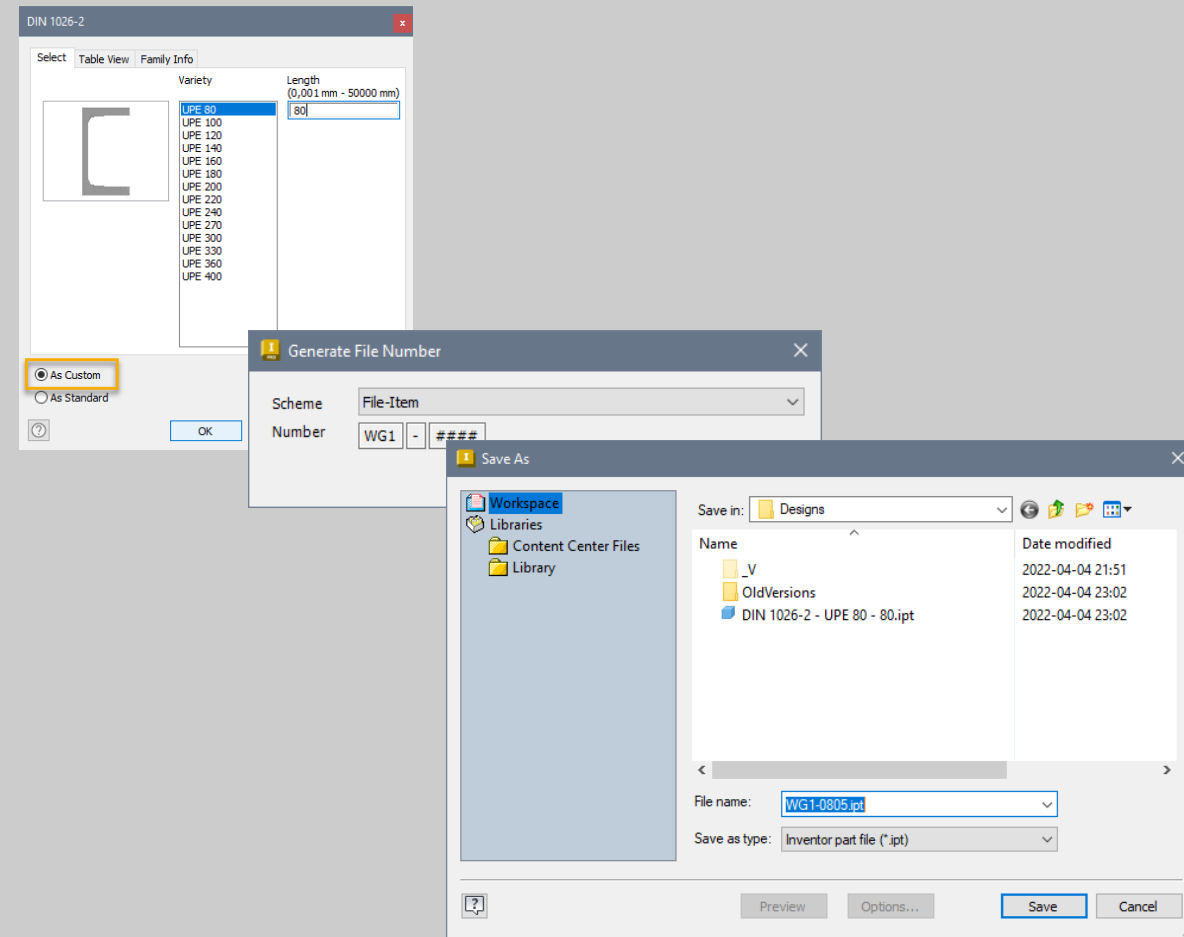
- Place from Content Center – Custom
 - **Default** workflow characteristics
 - Behaves like Inventor without Vault
 - No Vault file number proposed
 - No VDS interaction



Data Standard

Inventor

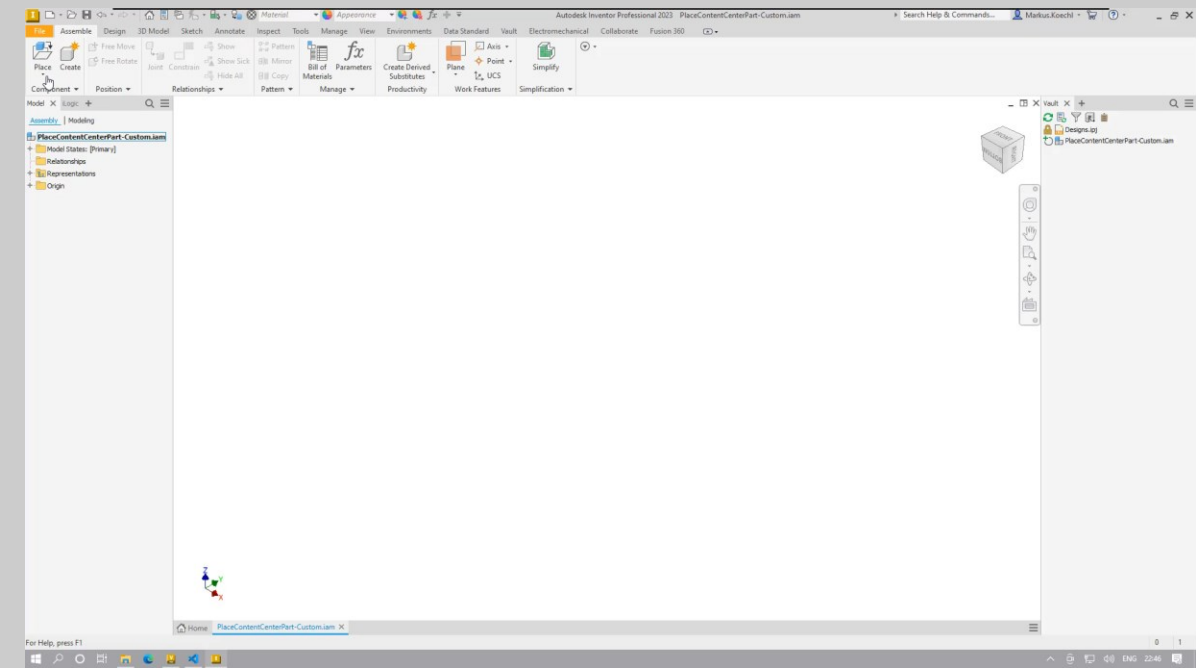
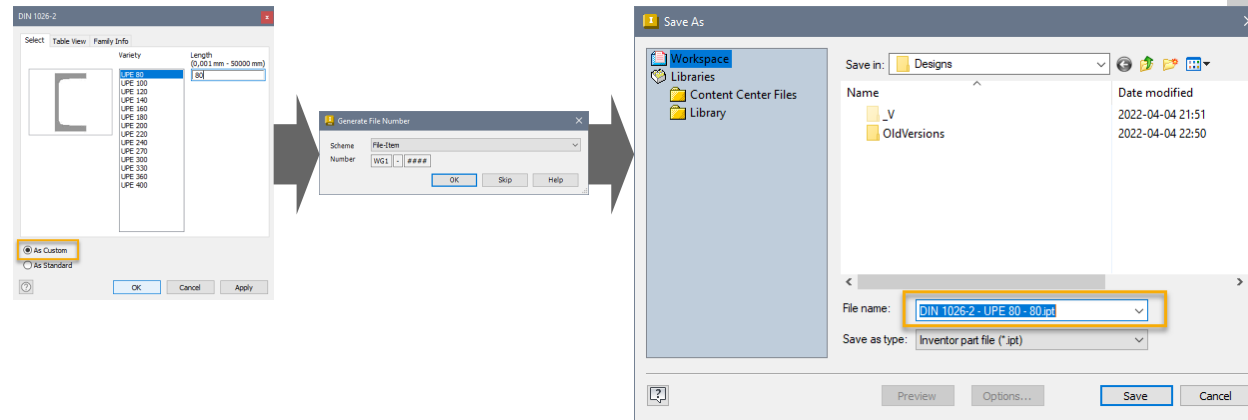
- Place from Content Center – Custom
 - Configuration options to enable
 - **Vault numbering only**
 - ⇒ Vault File Number Dialog to propose file name
 - ⇒ Save As Dialog to select file location
 - Enable by editing the configuration file
 - Add “InteractionEvents” to EnableVaultNumSchemeForCmds



Demo Video

Workflow option 1 – Vault numbering only

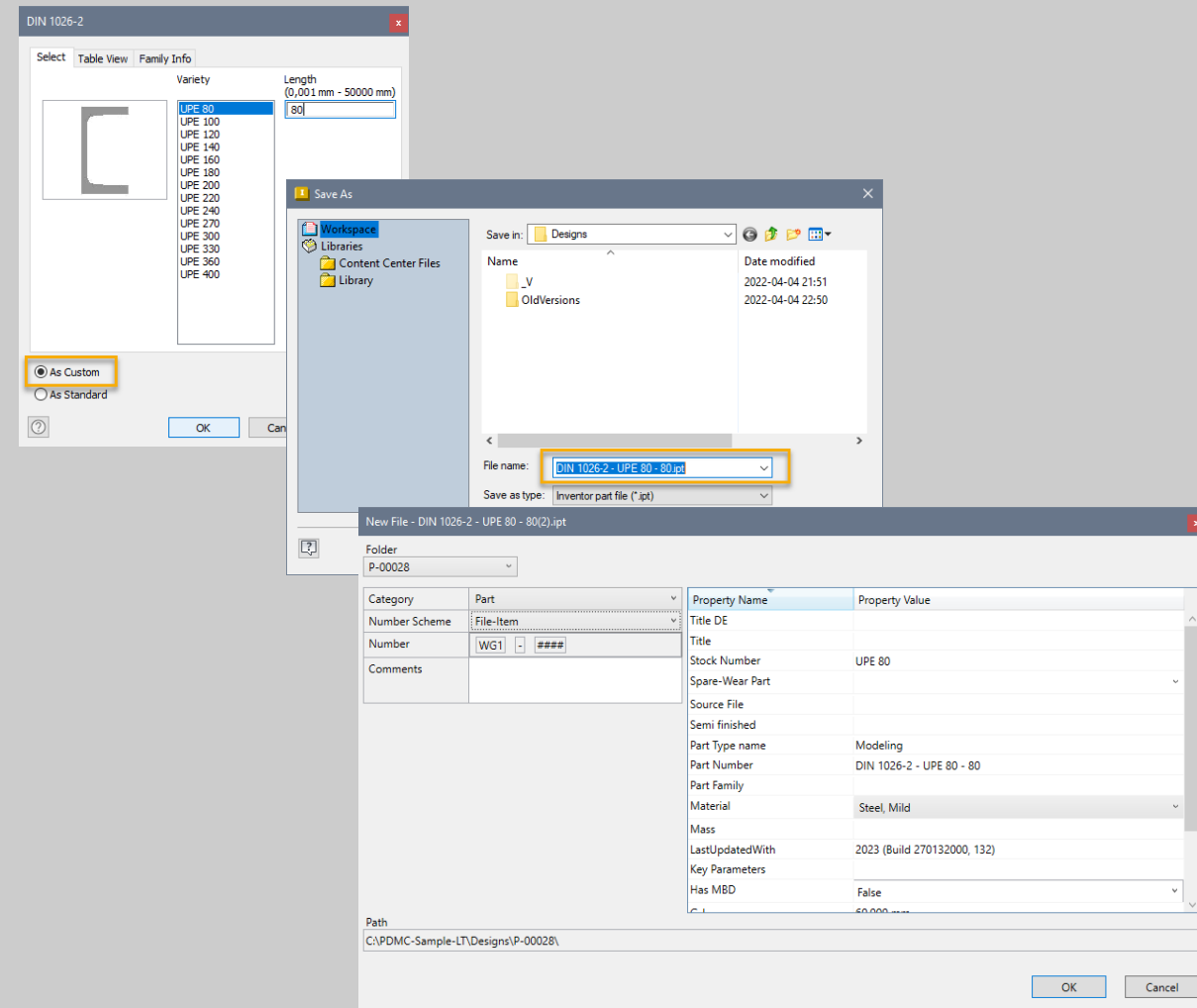
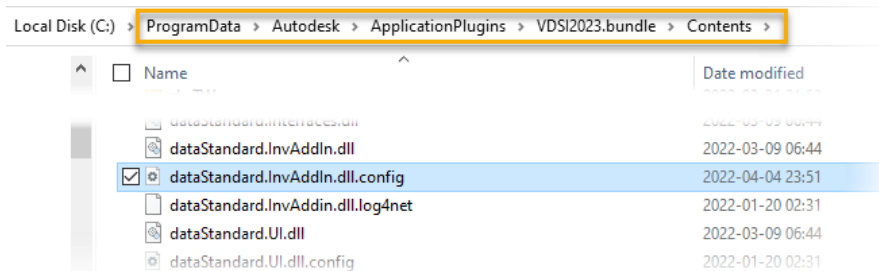
- Place from Content Center – Custom
 - Configuration option 1 characteristics
 - Behaves like Inventor **with** Vault
 - Vault file number proposed
 - No VDS interaction



Data Standard

Inventor

- Place from Content Center – Custom
 - Configuration options to enable
 - **Vault numbering & VDS Datasheet interaction**
 - ⇒ Save As Dialog is part of “Place from Content...”
 - ⇒ VDS Dialog overwrites
 - ⇒ File location (As selected in the VDS dialog)
 - ⇒ File name with file Vault number or manual file name
 - Enable by editing the configuration file
 - Remove “InteractionEvents” from DisableVdsForCmds



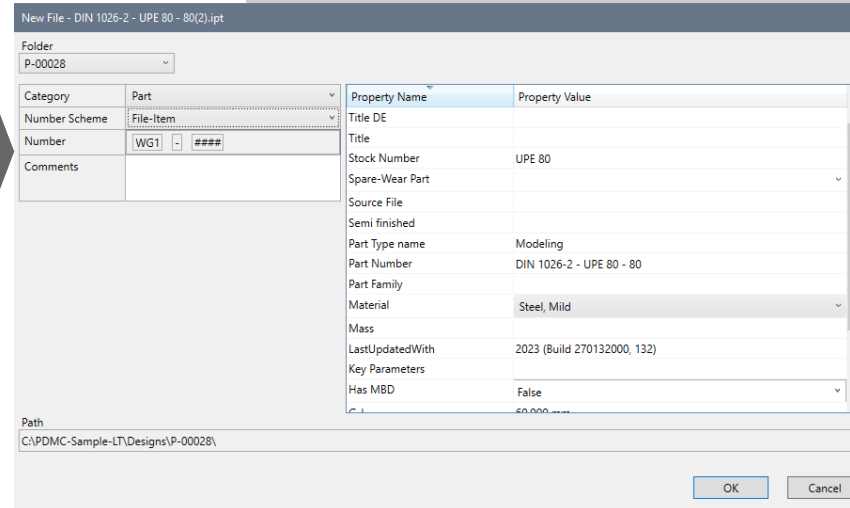
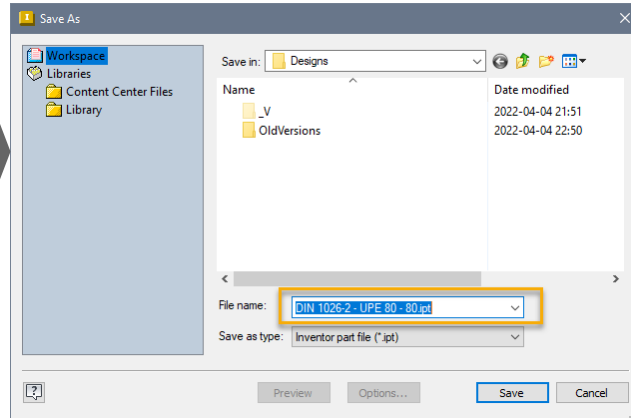
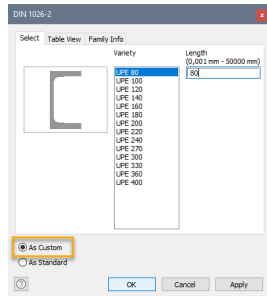
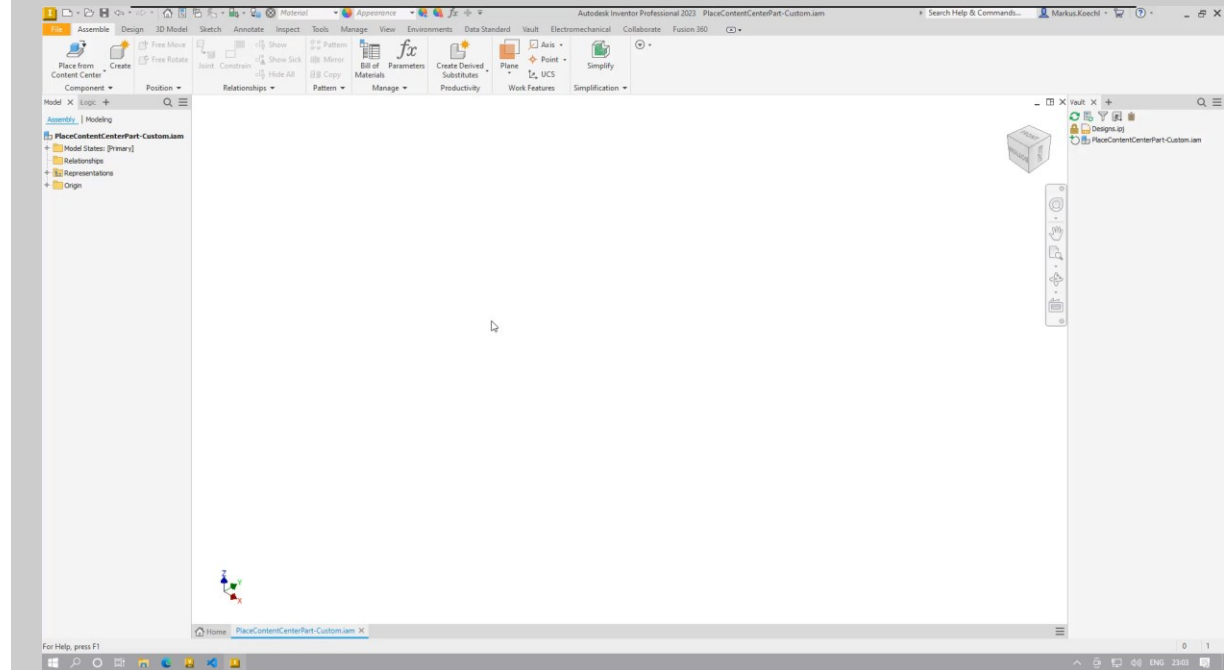
```
<setting name="DisableVdsForCmds" serializeAs="String">  
  <value>  
    AssemblyBonusTools_AddPartCmd, AssemblyBonusTools_AddAssembly,  
    Factory.AssetBuilder.AssetEditSave, Factory.AssetBuilder.Fi  
    AFG_CMD_ChangeProfile,  
    AssemblyMirrorComponentCmd,  
    InteractionEvents  
  </value>  
</setting>
```

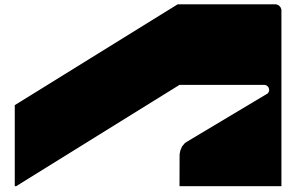
delete

Demo Video

Workflow option 2 – Vault Data Standard

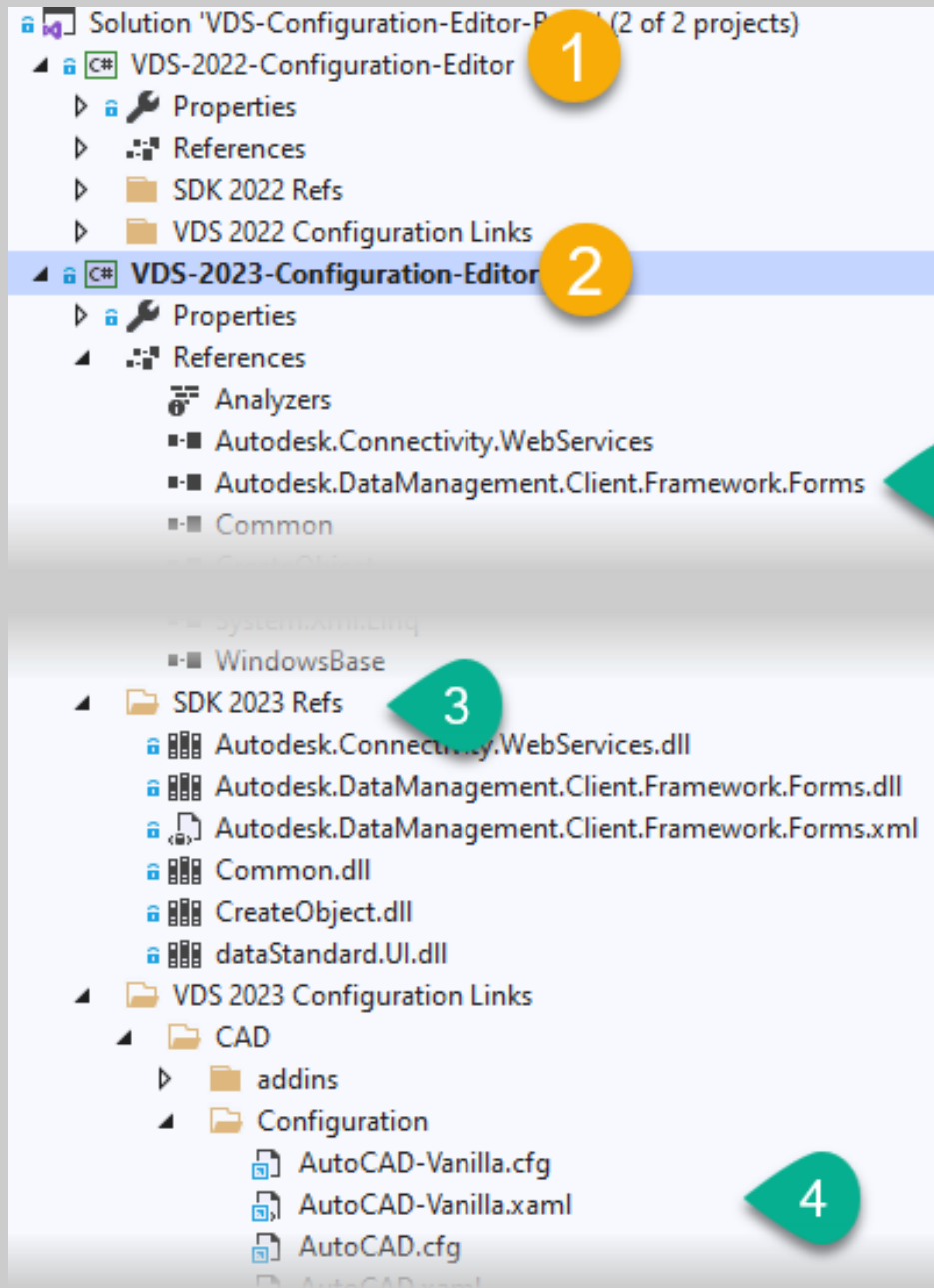
- Place from Content Center – Custom
 - Configuration option 2 characteristics
 - Inventor Save As appears
 - VDS Dialog drives (overrides Save As input)
 - Folder location
 - File name/numbering





Configuration Editor

Editor Basic | Editor Custom Configuration Samples



Editing Custom Configurations

VDS-Configuration-Editor-Basic

- Visual Studio 2019 Solution
 - Edit VDS-2022 Configurations (1)
 - Edit VDS-2023 Configurations (2)
 - Link configuration files (default) (4)
 - Embed configuration files, e.g., copies for *.Custom
- NEW
 - Embedded SDK reference sources (3)
 - ⇒ Open, Edit configurations without VDS release installed
 - ⇒ Open, Compare legacy configurations
 - Edit VDS-2023.1 XAML files using Themes (5)

Custom Configuration Samples 2023

VDS-Sample-Configuration-2023 (1)

- Visual Studio 2019 Solution

- Review/Compare VDS-2023-MFG-Sample (2)

- Custom Configuration for any Vault based on Manufacturing CFG

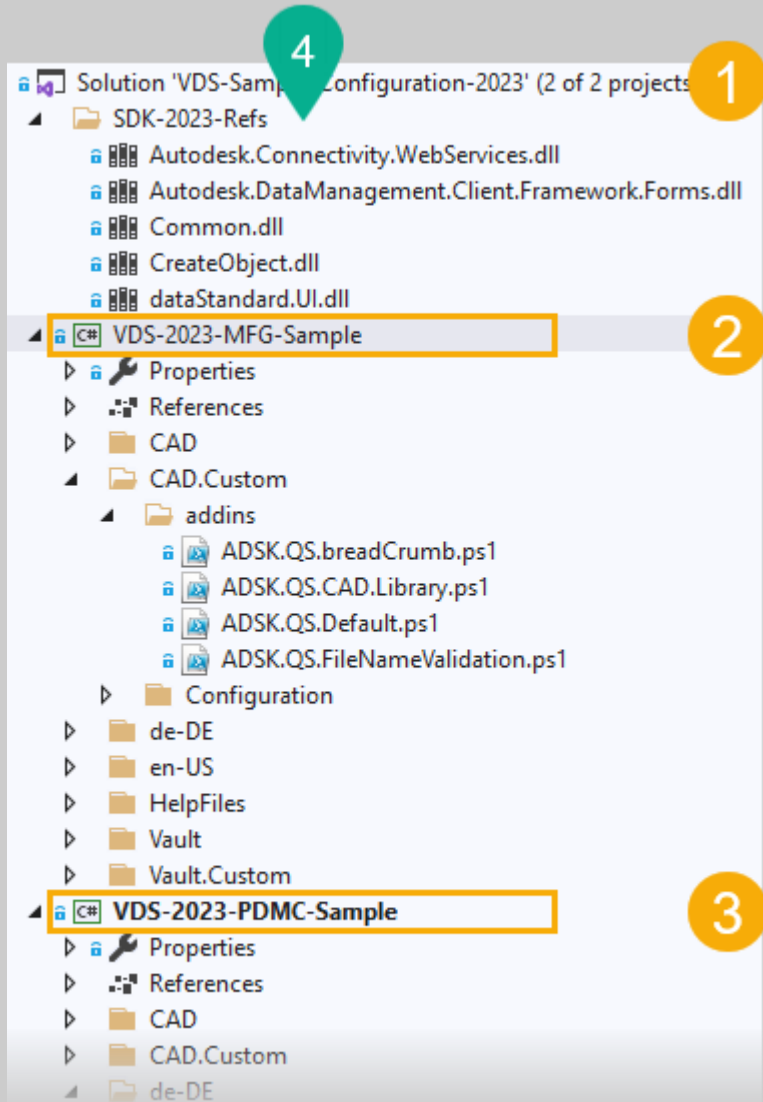
- Review/Compare VDS-2023-PDMC-Sample (3)

- Custom Configuration compatible with PDMC-Sample Vault only

- Embedded SDK reference sources on solution level (4)

⇒ Open, Edit configurations without VDS release installed

⇒ Open, Compare legacy configurations



<https://github.com/koechlm/VDS-Sample-Configuration-2023>



Migration

Custom configurations

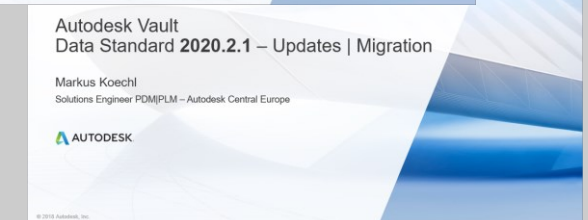
- How to update custom configurations 2021/2022 → 2023.2
 - Minimum Updates
 - Recommended Updates
- How to update custom configurations for **Themes** support → 2023.2

Minimum Updates

Note

Migration paths

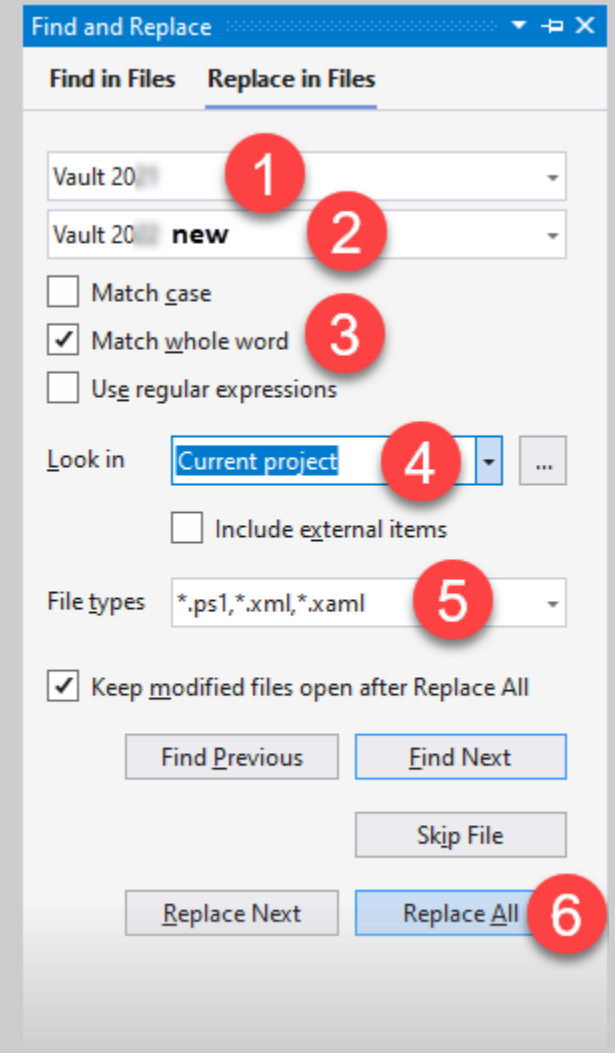
- This slide deck covers migration from VDS 2022 RTM to VDS 2023
- Available documentation for other release level migrations
 - Migration from **2020/2021 RTM** to **2022.2 (2022.3)**
 - [AKN Download](#)
 - Migration from **2020** and **2021 RTM** to **2021.1.2**
 - [AKN Download](#) – Migration 2020 – 2021.1.2
 - Configurations of **2019** require additional steps
 - [AKN Download](#) Migration 2019 – 2020.x.x



Align Application Paths

Minimum update

- Align Version
 - Find / Replace \Vault 202x\ by \Vault 2023\
 - All files of solution

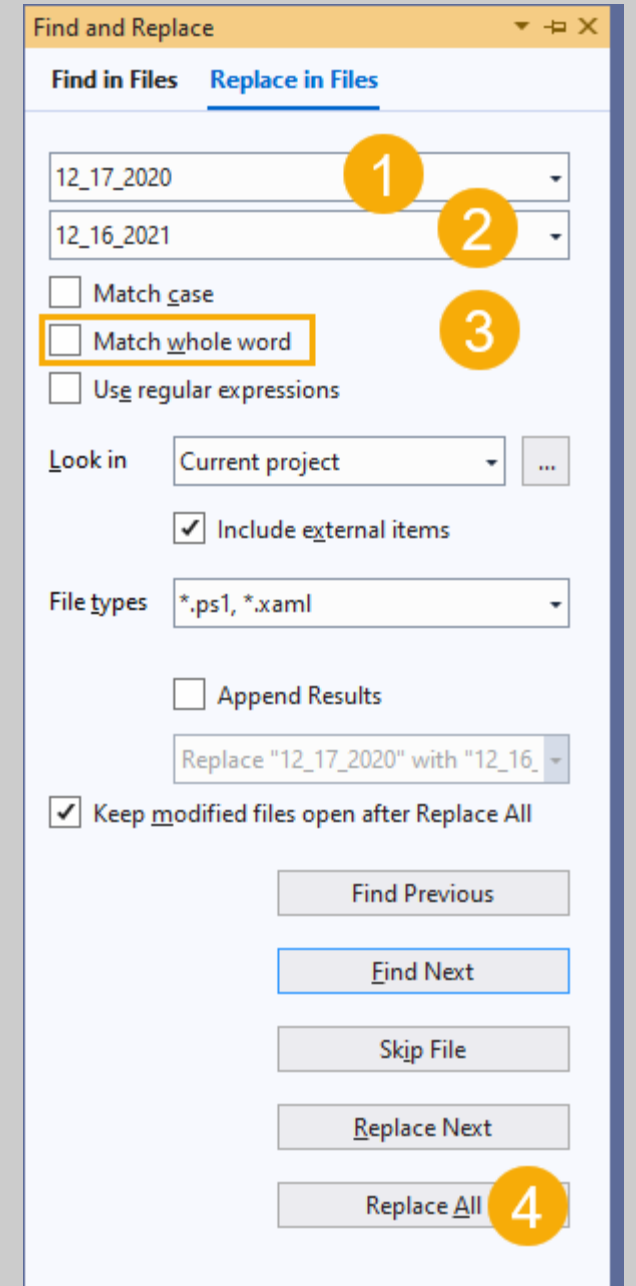


Shortcut User Profile Location

Minimum update for *VDS-Manufacturing-Sample**

- [only required if configuration implemented Shortcuts]
 - Replace path segment 12_17_2020 by 12_16_2021
 - **Note** – these folders don't represent release numbers: Vault 2023 added the folder Services_Security_12_16_2021

*: Formerly called "Quickstart" configurations

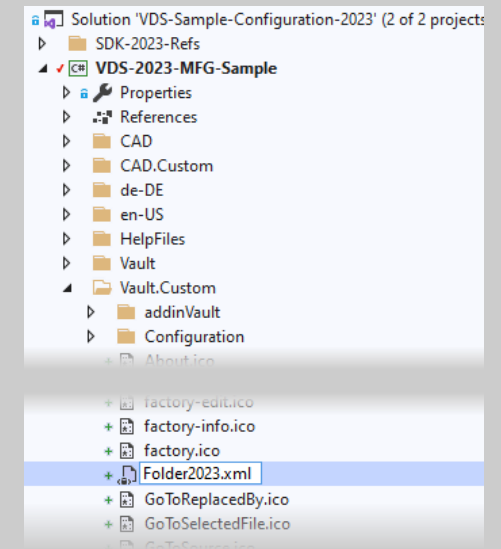
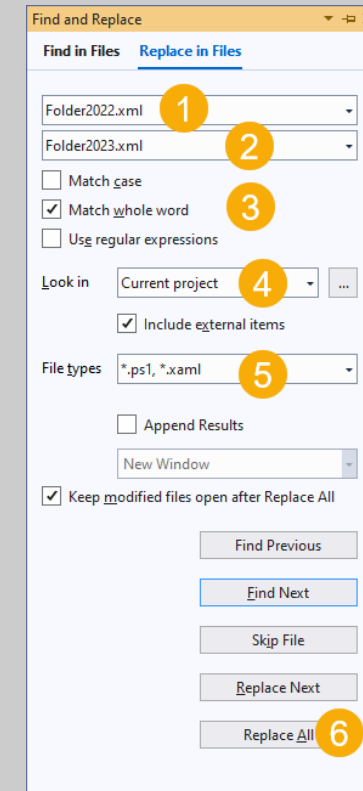


Last Used Folder

Minimum update for *VDS-Manufacturing-Sample**

- [only required if configuration implemented Shortcuts]
 - Replace substring Folder2022.xml by Folder2023.xml
 - Rename the file .\Vault.Custom\Folder2022.xml to Folder2023.xml

*: Formerly called "Quickstart" configurations



VDS-Sample-Utilities

Minimum update for *VDS-Manufacturing-Sample**

- The VdsSampleUtilities.dll 2023 added an extensible framework for
 - Event Handling
 - Vault (Server) based configuration options
- There are two options
 - Install the VDS-Sample-Utilities as a client extension (to be published later this year)
 - Use the VdsSampleUtilities.dll as shared with the VDS-MFG-Sample configuration
- Update all scripts that load the utilities by conditionally loading the preferred one (the notes share it in plain text):

```
# there are some custom functions to enhance functionality; 2023 version added webservice and explorer extensions to be installed optionally
$mVdsUtilities = "$($env:programdata)\Autodesk\Vault 2023\Extensions\Autodesk.VdsSampleUtilities\VdsSampleUtilities.dll"
if (! (Test-Path $mVdsUtilities)) {
    #the basic utility installation only
    [System.Reflection.Assembly]::LoadFrom($Env:ProgramData + "\Autodesk\Vault 2023\Extensions\DataStandard\Vault.Custom\addinVault\VdsSampleUtilities.dll")
}
Else {
    #the extended utility activation
    [System.Reflection.Assembly]::LoadFrom($Env:ProgramData + "\Autodesk\Vault 2023\Extensions\Autodesk.VdsSampleUtilities\VdsSampleUtilities.dll")
}
```

*: Formerly called "Quickstart" configurations

Dynamic Template Folder Setting

Replace static link

- Copy new function “GetTemplateFolders” from the Autodesk default to your `.\Vault.Custom\addin\Vault\default.ps1`
 - And follow next slide to resolve a known issue
- Replace ComboBox control definition in your custom `File.xaml` and `FileOffice.xaml`

```
426 function GetTemplateFolders
427 {
428     $xmlpath = "$env:programdata\Autodesk\Vault 2022\Extensions\DataStandard\Vault\Configuration\File.xml"
429 }
430 if ($_IsOfficeClient) {
431     $xmlpath = "$env:programdata\Autodesk\Vault 2022\Extensions\DataStandard\Vault\Configuration\FileOffice.xml"
432 }
433 }
434 $xmldata = [xml](Get-Content $xmlpath)
435 }
436 [string[]] $FolderPath = $xmldata.DocTypeData.DocTypeInfo | foreach { $_.Path }
437 $folders = $vault.DocumentService.FindFoldersByPaths($FolderPath)
438 }
439 return $xmldata.DocTypeData.DocTypeInfo | foreach {
440     $path = $_.Path
441     $folder = $folders | where { $_.FullName -eq $path } | Select -index 0
442     if ($folder -eq $null)
443     {
444         return
445     }
446     return $_
447 }
448 }
```

```
<!-- Don't use static xml resource, use viewmodel binding instead. -->
<ComboBox ItemsSource="{Binding PsList[GetTemplateFolders], Mode=OneTime}"
    x:Name="DocTypeCombo"
    Grid.Row="1"
    Grid.Column="1"
    IsEnabled="{Binding CreateMode}"
    DisplayMemberPath="DocName"
    SelectedValuePath="Path"
    SelectedValue="{Binding TemplatePath, Mode=OneWayToSource}"
    SelectedIndex="0"
    Style="{StaticResource DocTypeStyle}" />
```

VDS Vault Office

Known issue for 2023 RTM

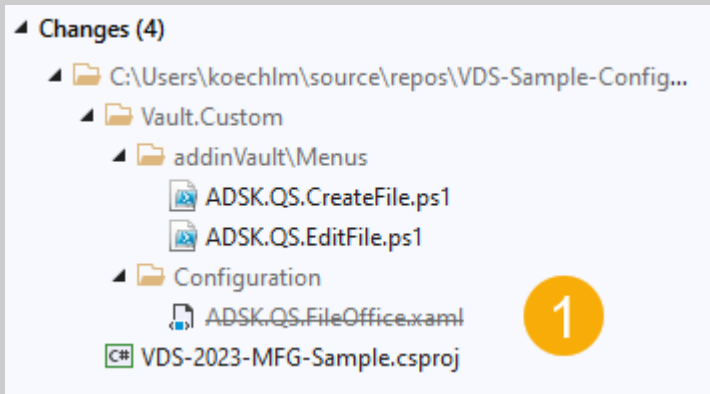
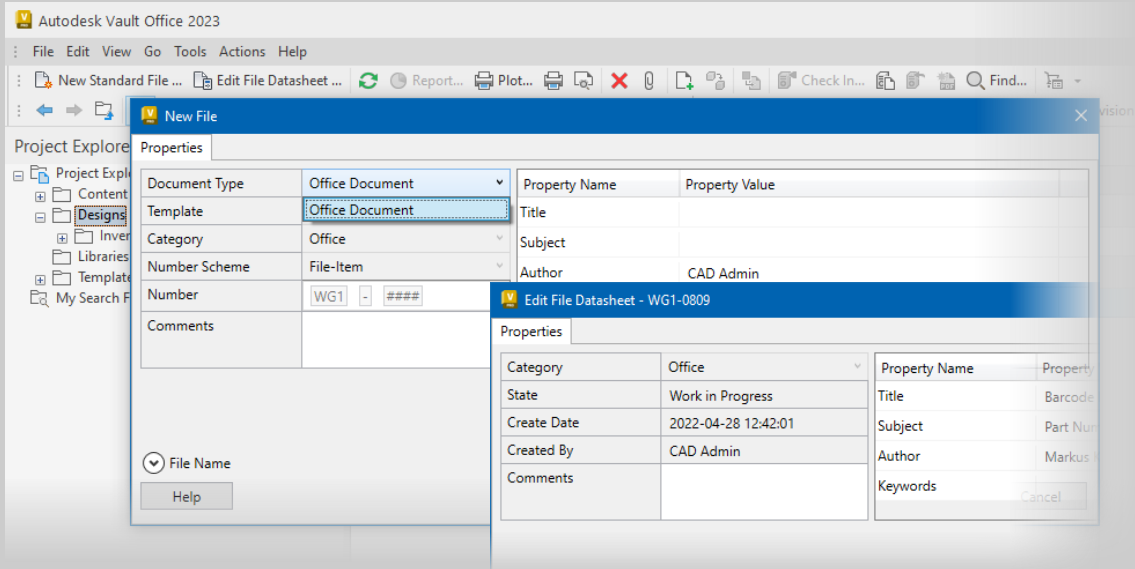
- If condition is not applicable
 - Replace
 - `$Prop["_IsOfficeClient"].Value`
by
 - `$_IsOfficeClient`

Resolved with 2023.1

```
426 function GetTemplateFolders
427 {
428     $xmlpath = "$env:programdata\Autodesk\Vault 20
429
430     if ($Prop["_IsOfficeClient"].Value) {
431         if ($_IsOfficeClient) {
432             $xmlpath = "$env:programdata\Autodesk\Vault
433         }
434     }
435     $xmldata = [xml](Get-Content $xmlpath)
```

Dynamic Template Folder Setting

Recommended Updates



```
#override the default dialog file assigned
if ($_IsOfficeClient)
{
    $xamlFile = New-Object CreateObject.WPF.XamlFile "Adsk.QS.FileOffice.xaml", "C:\ProgramData\Autodesk\Vault 2023\Extensions\
    $dialog.XamlFile = $xamlFile
}
else
{
    $xamlFile = New-Object CreateObject.WPF.XamlFile "ADSK.QS.File.xaml", "C:\ProgramData\Autodesk\Vault 2023\Extensions\DataStanc
    $dialog.XamlFile = $xamlFile
}
```

VDS Vault Office

Making FileOffice.XAML unnecessary

- File.xaml dialog is compatible for Vault Pro and Vault Office due to the new dynamic template reader (GetTemplateFolders)

⇒ Eliminate maintenance efforts of 2 dialog definitions

Steps

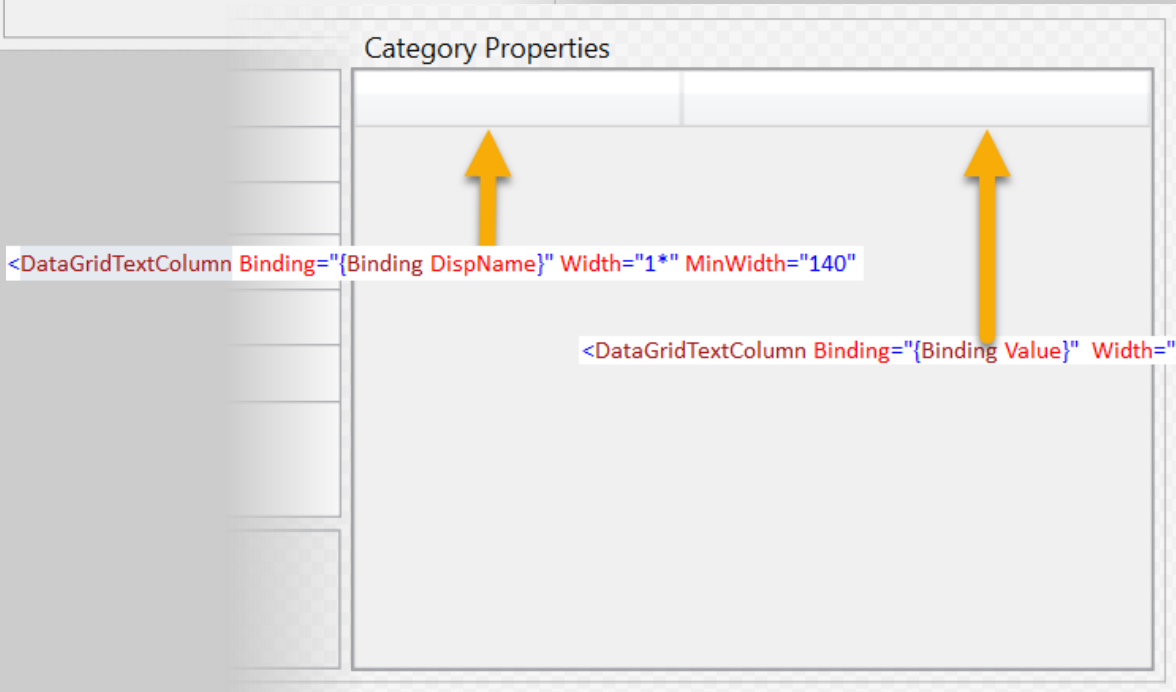
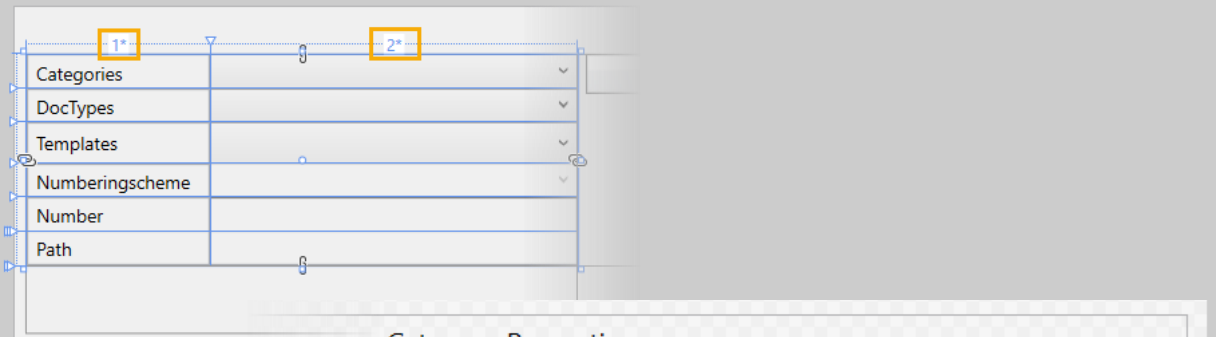
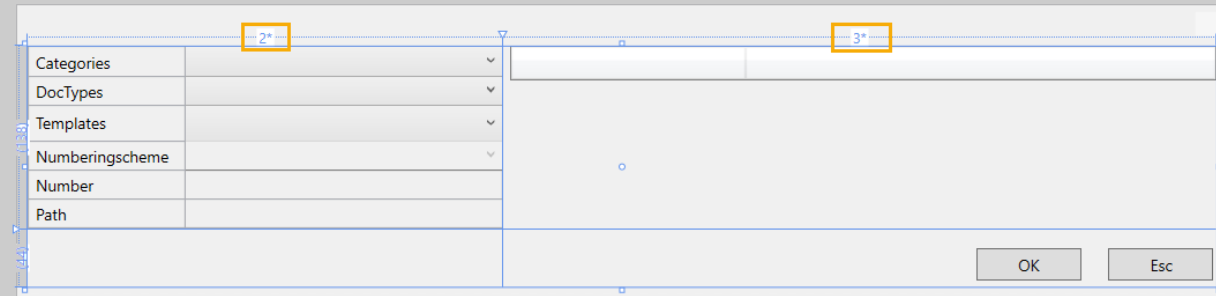
- Delete dialog file (1)
.\VaultCustom\Configuration\FileOffice.XAML
- Delete if..then differentiation in
 - CreateFile.ps1 (2)
 - EditFile.ps1 (2)

VDS Vault

Dynamic Grid Sizing

- Enable dynamic grid sizing for all
 - Dialogs
 - Detail Tab Datasheets

- Tip – Edit the values in XML instead of graphics screen



VDS Vault

VDS-Manufacturing Sample

- Fixed/Minimum Sizes (Legacy)

Reports | [Datasheet](#)

Category	Project	Property Name	Property Value
Name	P-00000	Title	FDU Mannheim Project
State	Order	Description	FDU Demo FY20
Create Date	2018-11-19 11:58:00	Date Start	2018-11-19 <input type="text" value="15"/>
Created By	CAD Admin	Date End	2018-05-30 <input type="text" value="15"/>
		Customer	Autodesk
		Project Number	

Comments

- Dynamic Grid Sizing 2023+

Reports | [Datasheet](#)

Category	Project	Property Name	Property Value
Name	P-00000	Title	FDU Mannheim Project
State	Order	Description	FDU Demo FY20
Create Date	2018-11-19 11:58:00	Date Start	2018-11-19 <input type="text" value="15"/>
Created By	CAD Admin	Date End	2018-05-30 <input type="text" value="15"/>
		Customer	Autodesk
		Project Number	

Comments

UIString.xml

Extended validation

- Add the Attribute “VAL15” to custom configurations
 - It can be used for the extended folder name and file name validations
 - Future updates also might validate user input fields of numbering schemes
 - Nominated for 2023.1

```
<UIString ID="VAL15">The field cannot contain any of the following characters &quot;/?&lt;&gt;\;*|</UIString>
```

New File

Category	Sheet Metal Part	Property
Document Type	Inventor	Title
Template	Sheet Metal.ipt	Make o
Number Scheme	FreeText Scheme	MBD
Number	!<>/ - #####	Key Para
Path	\$/Designs/FDU Tutorial	Title DE
		Part Fan

VDS Vault

Detail Tab – CAD BOM

- Note – The CAD BOM represents a file-based BOM representation and does not display
 - Instance Properties
- Recommendation
 - Several changes have been made in the default FileBOM.ps1 file over the past releases.
 - Review/Replace your custom FileBOM.ps1 file.
 - Re-use the latest FileBOM.ps1 from Update 1 or later in your customization folder
.\Vault.Custom\addinVault\
 - Comment out \$dsDiag.Trace() calls.

Latest updates with
2023.1

```
61 }
62 $bomItem Name = $cldBom.CompArray[0].Name
63 $bomItem ComponentType = $cldBom.CompArray[0].ComponentType
64 $uniqueId = $comp.UniqueId
65 #find component in current file bom only
66 $cldComp = $cldBom.CompArray | Where-Object { $_.UniqueId -eq $uniqueId } | Select -First 1
67 if (-not $cldComp) {
68     $cldComp = $cldBom.CompArray[0]
69     $dsDiag.Trace("Can not find cld comp for uniqueId: " + $uniqueId + ", use first data.")
70 }
71 $bomItem.Name = $cldComp.Name
72 $bomItem.ComponentType = $cldComp.ComponentType
73 $cldCompAttrArray = $cldBom.CompAttrArray | Where-Object { $_.CompId -eq $cldComp.Id}
74 if ($cldCompAttrArray.Count -eq 0) {
75     $cldCompAttrArray = $cldBom.CompAttrArray
76     $dsDiag.Trace("cld comp attr array is empty for compId: " + $cldComp.Id + ", use entire array.")
77 }
78 $propPartNumber = $cldBom.PropArray | Where-Object { $_.DispName -eq "Part Number"}
79 $prop = ($cldBom.CompAttrArray | Where-Object { $_.PropId -eq $propPartNumber.Id}) | Select -First 1
80 $prop = ($cldCompAttrArray | Where-Object { $_.PropId -eq $propPartNumber.Id}) | Select -First 1
81 $bomItem.PartNumber = $prop.Val
82 $bomItems += $bomItem
83 #add inventor default BOM columns
84 $thumbnailProp = $vault.PropertyService.GetProperties('FILE', @($cldIds[$cldBomCounter - 1]), @($thumbnailPropDef.Id))[0]
85 $bomItem.Thumbnail = $thumbnailProp.Val
86 $m_Prop = $cldBom.PropArray | Where-Object { $_.DispName -eq "Title"}
87 $prop = ($cldBom.CompAttrArray | Where-Object { $_.PropId -eq $m_Prop.Id}) | Select -First 1
88 $prop = ($cldCompAttrArray | Where-Object { $_.PropId -eq $m_Prop.Id}) | Select -First 1
89 $bomItem.Title = $prop.Val
90 $m_Prop = $cldBom.PropArray | Where-Object { $_.DispName -eq "Description"}
```

```
$dsDiag.Trace(">> Starting GetFileBOM($fileID)")
# $dsDiag.Trace(">> Starting GetFileBOM($fileID)")
$bom = $vault.DocumentService.GetBOMByFileId($fileID)
$cldIds = @()
$bom.InstArray | Where-Object { $_.ParId -eq 0 } | ForEach-Object {
    $cldId = $_.CldId
    $comp = $bom.CompArray | Where-Object { $_.Id -eq $cldId }
    if ($comp.XRefId -ne -1) {
        $cldIds += $comp.XRefId
    }
}
$dsDiag.Trace(" cldIds: "+$cldIds.Count)
# $dsDiag.Trace(" cldIds: "+$cldIds.Count)
$bomItems = @()
if ($cldIds.Count -gt 0) #the file contains BOM information, so continue
```


VDS CAD

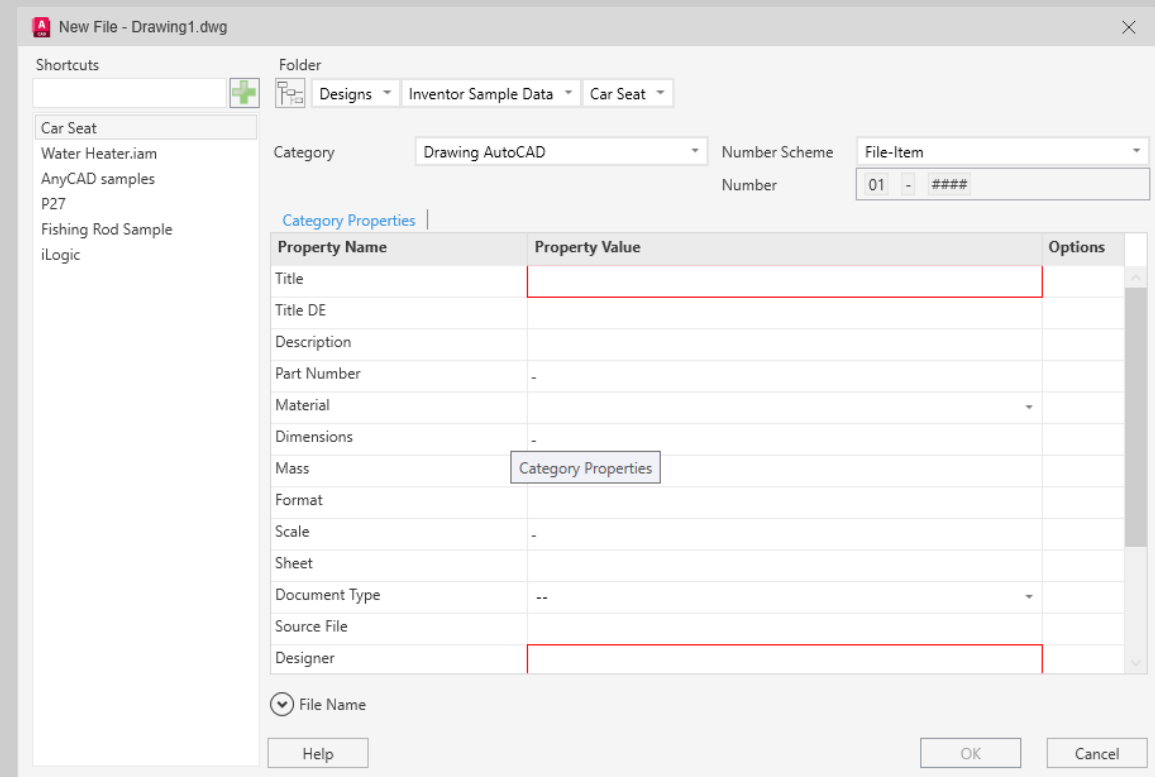
Inventor and AutoCAD

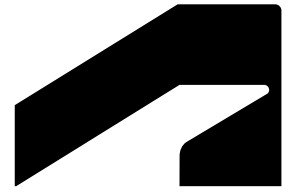
- Reset the WindowStyle for
 - ShowInTaskbar
 - WindowStyle
 - Topmost
 - Background

⇒ Benefits

- ⇒ The VDS dialogs get the parent application's icon (WindowStyle)
- ⇒ The VDS dialogs allow sub-windows and don't hide them (Topmost)
- ⇒ The VDS dialog can be activated although it is no longer topmost (ShowInTaskbar)
- ⇒ The VDS dialog is prepared for themes (Background); see also Enabling Themes section

```
x:name="InventorWindow" WindowStyle="ToolWindow" ShowInTaskbar="False" Topmost="True"
Background="#F0F0F0" ResizeMode="NoResize" SizeToContent="WidthAndHeight">
x:name="InventorWindow" ResizeMode="NoResize" SizeToContent="WidthAndHeight">
```





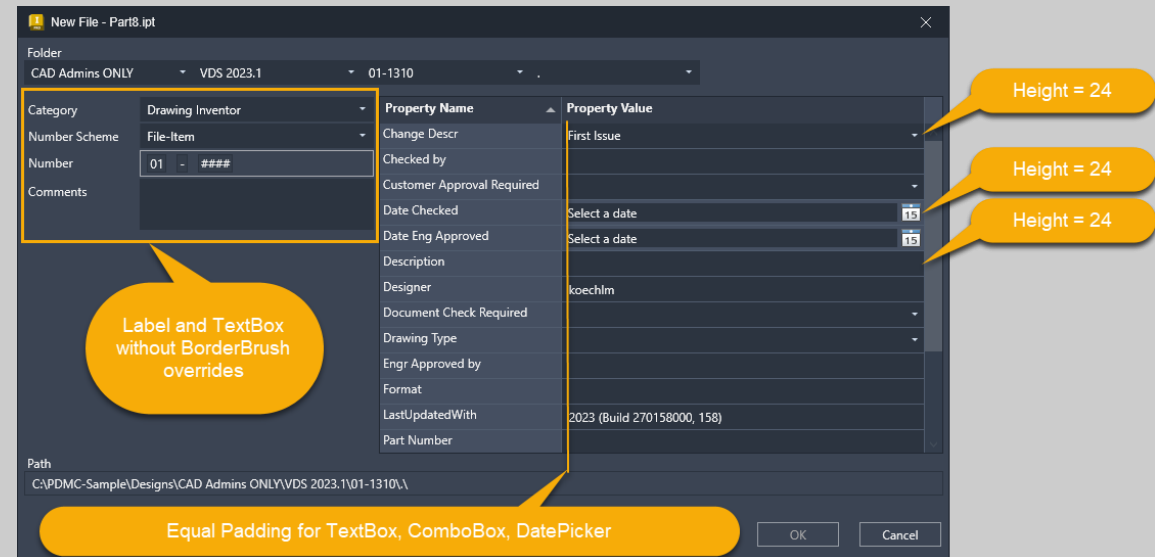
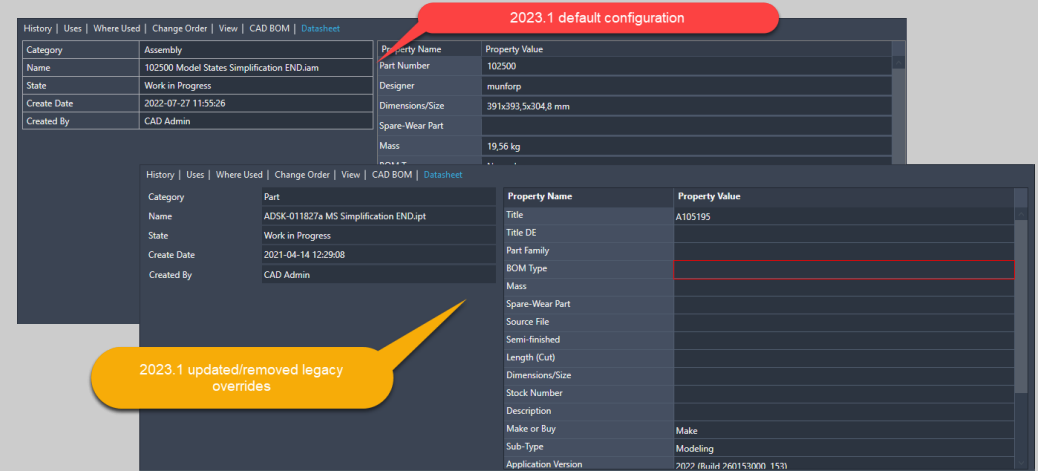
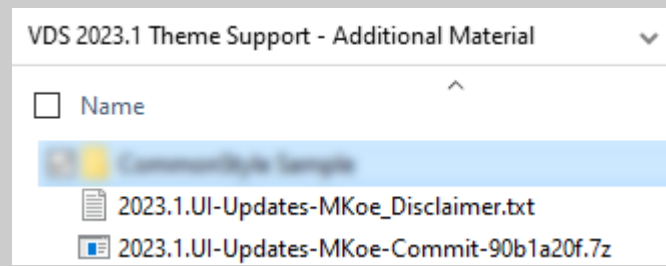
Enabling Themes

Additional Material – Useful resources to copy & paste themed styles or code snippets for PowerShell

Additional Material

Updated default configuration


- 7z Package requires a password
 - Password application = disclaimer acceptance
 - This update is a WiP version and does not reflect all enhancements of future Vault update releases
- Enhancements
 - Resolved known issues
 - Added default style references for any control that customizations might add
 - Removed all static overrides that the legacy “Classic” theme differentiated from Windows Forms UI



Additional Material

Updated default configuration

- Details – Enhancement History (Bottom-up)



update configuration template with themed error/message dialogs	2023.1.UI-Updates-MKoe	origin/2023.1.UI-Update...
line ending normalization Windows		
removed textbox style "Required": legacy style no longer required since ValidatedBinding adds formatting to the control		
adjusted TextBox Padding Value left=4		
fixed file name validation (avoid empty file names) for CAD dialogs on create.		
implemented default themed styles for all standard controls to support customizations and extensions; different style regions for CAD, VE dialogs, a...		
fixed name validation for Custom Objects (https://forums.autodesk.com/t5/vault-customization/edit-custom-object-datasheet-not-working/m-p/...)		
default UI update proposal removing all overrides interfering with themes		
added VDS 2023.1 sources and references	2023.1.0.112	origin/2023.1.0.112

VDS 2023.1 Theme Support - Additional Material

Name

Community Sample

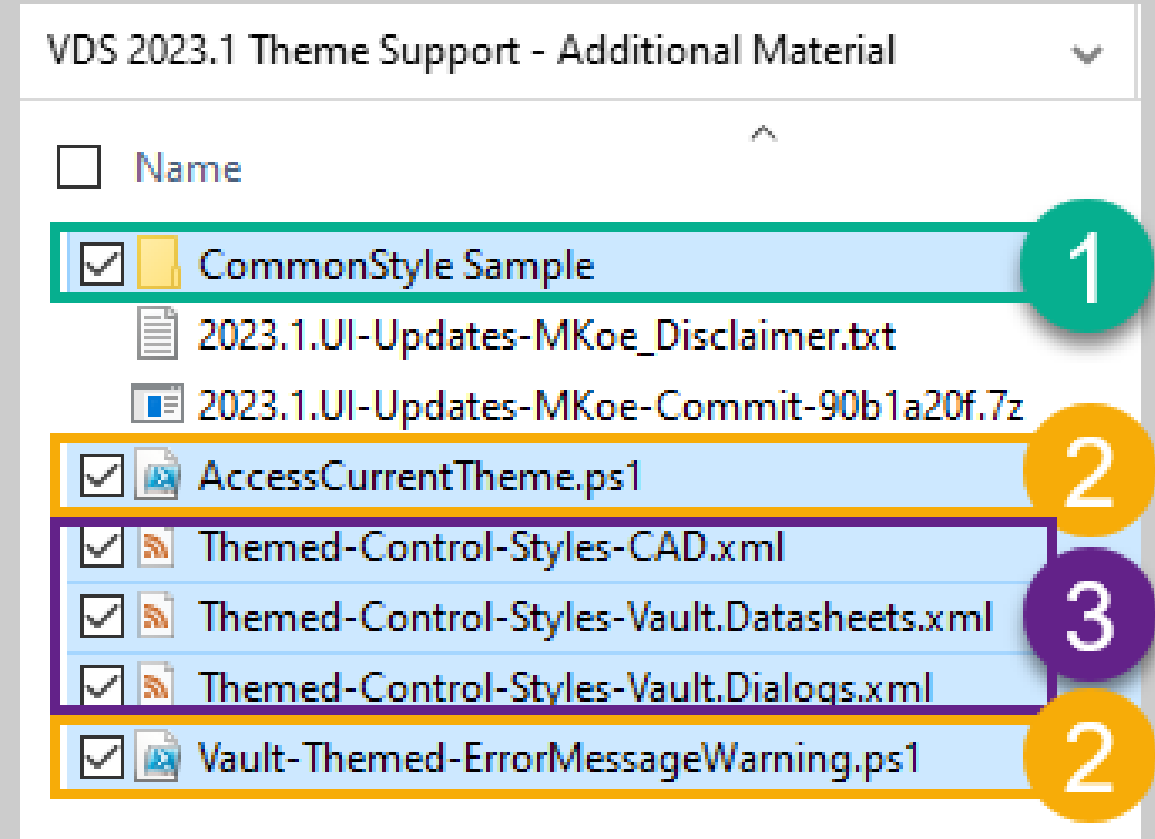
2023.1.UI-Updates-MKoe_Disclaimer.txt

2023.1.UI-Updates-MKoe-Commit-90b1a20f.7z

Additional Material

Code snippets and samples

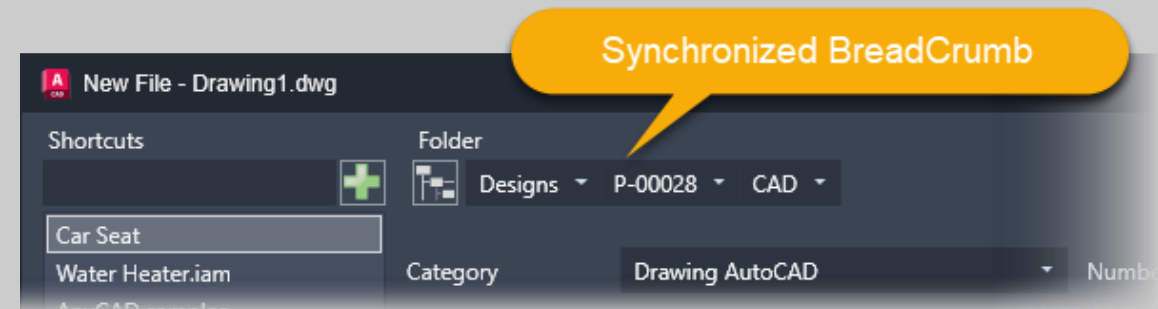
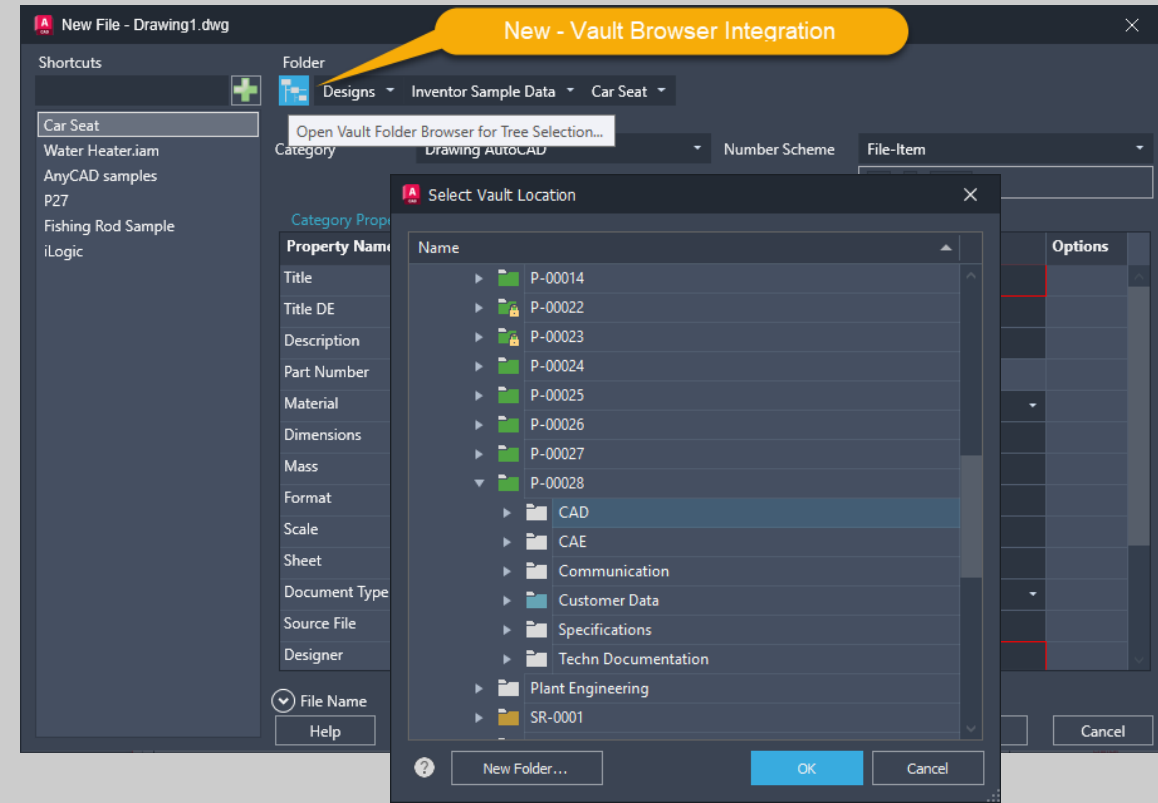
1. CommonStyle Sample
 - centralized XAML style dictionary and re-use for Vault, AutoCAD new file dialogs
2. PowerShell code snippets
3. Style resources
 - Inventor, AutoCAD Dialogs
 - Vault Datasheets | Detail Tabs
 - Vault Dialogs

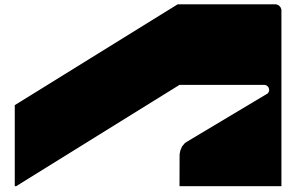


Additional Material

VDS-MFG-Sample Configuration 2023.1

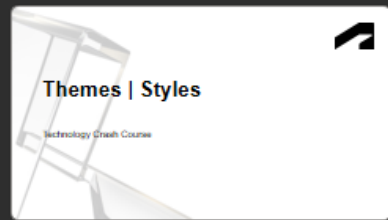
- Based on Themed Styles
 - Merged Updated default configuration changes
- What's New – Sample 2023.1
 - Vault Folder Browser
 - Themed style as a default
 - Create new folders on the fly
 - Integrates with BreadCrumb, last used folder, and shortcuts
 - BreadCrumb saves the last **applied** folder on dialog leave
 - Vault Browser saves the last **selected** folder
 - Can be aligned to last applied if preferable





Enabling Themes

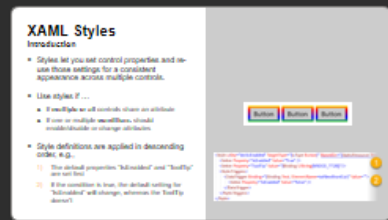
Concept | Step by Step Instructions | Tips & Tricks



60 ★



61 ★



62 ★



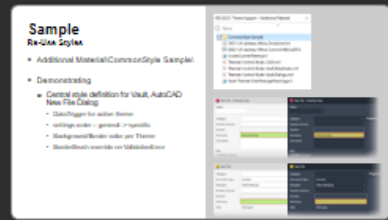
63 ★



64 ★



65 ★



66 ★



67 ★

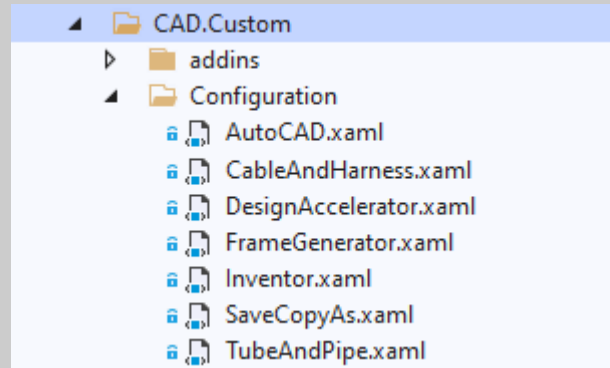
Knowledge Check

Are you familiar with “Themes”

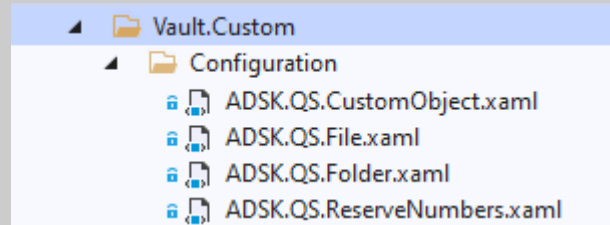
- If you are first time touching “Themes” and “Styles” for XAML, we suggest reviewing the addendum “Themes & Styles” first.

Data Standard Dialog-/Datasheet Types

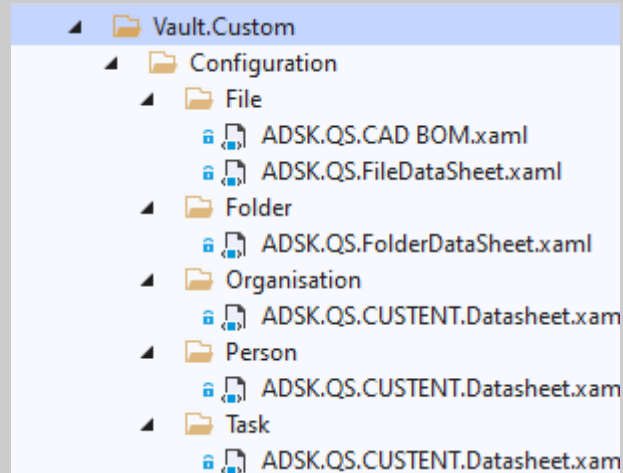
- VDS CAD Dialogs



- VDS Vault Dialogs



- VDS Vault Detail Tabs



Enabling Themes

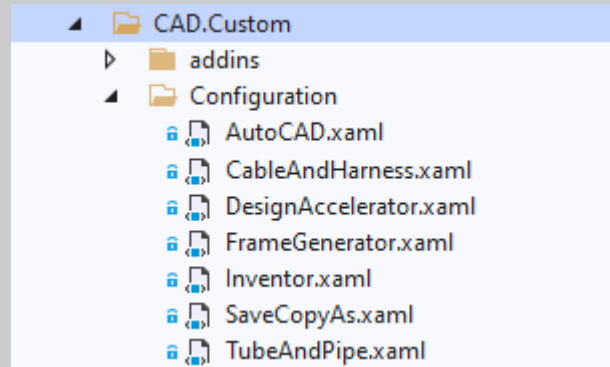
Apply 4 steps to *.Custom XAML files

1. Add references and resources
2. Update styles using “themed...” ones
3. Remove any custom style overrides
 - static background or border colors
4. Create a custom style if no themed style resource is available
 - Leveraging base style settings
 - Leveraging theme-based triggers in custom styles

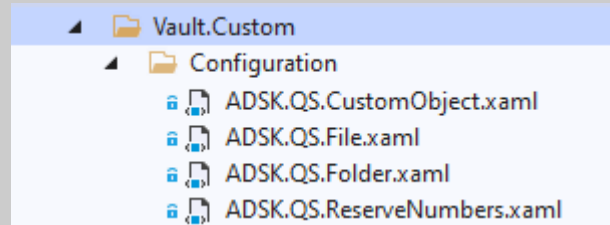
Note – Steps 1, and 2 are slightly different per VDS Datasheet/Dialog Type

Data Standard Dialog-/Datasheet Types

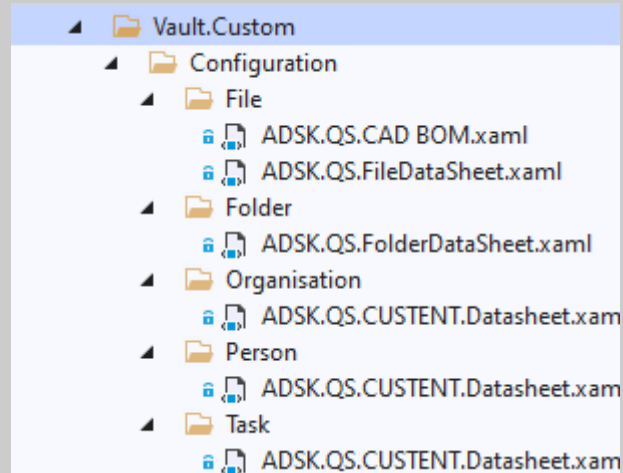
- VDS CAD Dialogs



- VDS Vault Dialogs



- VDS Vault Detail Tabs



Remark

Apply 4 steps to *.Custom XAML files

1. *The step-by-step documentation on the following slides has been captured, migrating the VDS-MFG-Sample configuration to support themes*
2. *The names of configuration files (*.ps1, *.xaml) may vary in your customized environment*

Step 1a – All XAML Types

New references

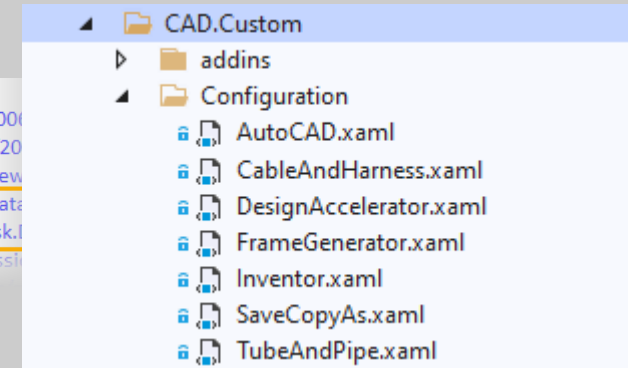
Add the reference of the Vault Forms resources supporting themes

- `xmlns:wpfUtilities="clr-namespace:Autodesk.DataManagement.Client.Framework.Controls.WPF.Utilities;assembly=Autodesk.DataManagement.Client.Framework.Forms"`
- `xmlns:winFormTheme="clr-namespace:Autodesk.DataManagement.Client.Framework.Forms.SkinUtils;assembly=Autodesk.DataManagement.Client.Framework.Forms"`

Data Standard Dialog-/Datasheet Types

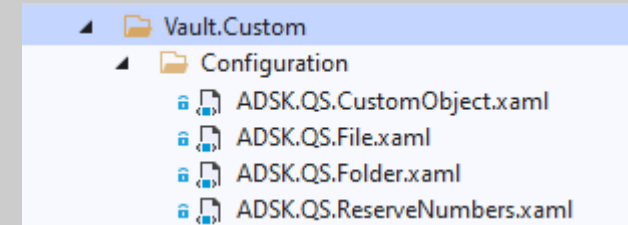
• VDS CAD Dialogs

```
<WPF:DSWindow
  xmlns="http://schemas.microsoft.com/winfx/2006/05/xaml"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/05/xaml"
  xmlns:WPF="clr-namespace:Autodesk.DataManagement.Client.Framework.Controls.WPF.Utilities;assembly=Autodesk.DataManagement.Client.Framework.Forms"
  xmlns:wpfUtilities="clr-namespace:Autodesk.DataManagement.Client.Framework.Controls.WPF.Utilities;assembly=Autodesk.DataManagement.Client.Framework.Forms"
  xmlns:winFormTheme="clr-namespace:Autodesk.DataManagement.Client.Framework.Forms.SkinUtils;assembly=Autodesk.DataManagement.Client.Framework.Forms"
  xmlns:d="http://schemas.microsoft.com/expression/2010/declarative"
  ...
>
```



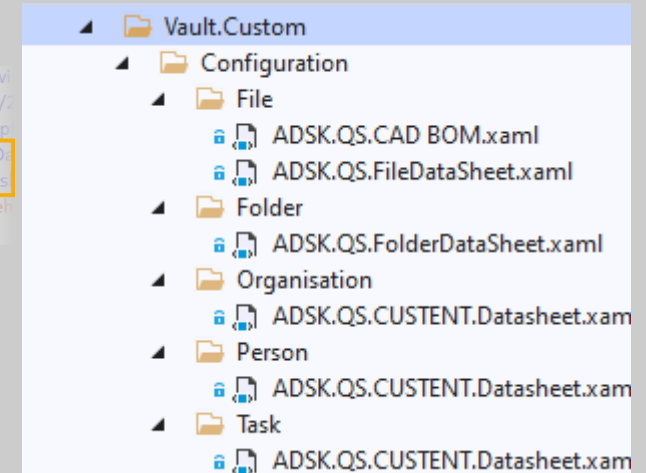
• VDS Vault Dialogs

```
<WPF:MainWindow xmlns="http://schemas.microsoft.com/winfx/2006/05/xaml"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/05/xaml"
  x:Name="FileWindow"
  xmlns:WPF="clr-namespace:Autodesk.DataManagement.Client.Framework.Controls.WPF.Utilities;assembly=Autodesk.DataManagement.Client.Framework.Forms"
  xmlns:wpfUtilities="clr-namespace:Autodesk.DataManagement.Client.Framework.Controls.WPF.Utilities;assembly=Autodesk.DataManagement.Client.Framework.Forms"
  xmlns:winFormTheme="clr-namespace:Autodesk.DataManagement.Client.Framework.Forms.SkinUtils;assembly=Autodesk.DataManagement.Client.Framework.Forms"
  AllowsTransparency="False" ResizeMode="NoResize"
  <Window.Resources>
```



• VDS Vault Detail Tabs

```
<UserControl xmlns="http://schemas.microsoft.com/winfx/2006/05/xaml"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/05/xaml"
  xmlns:behaviours="clr-namespace:Autodesk.DataManagement.Client.Framework.Controls.WPF.Behaviours;assembly=Autodesk.DataManagement.Client.Framework.Forms"
  xmlns:wpfUtilities="clr-namespace:Autodesk.DataManagement.Client.Framework.Controls.WPF.Utilities;assembly=Autodesk.DataManagement.Client.Framework.Forms"
  xmlns:winFormTheme="clr-namespace:Autodesk.DataManagement.Client.Framework.Forms.SkinUtils;assembly=Autodesk.DataManagement.Client.Framework.Forms"
  x:Name="MainWindow" behaviours:TabTitleBehavior="TabTitleBehavior"
  <UserControl.Resources>
```



Step 1b – CAD/Vault Dialogs

New Style Resource

Add the Themed Window Style

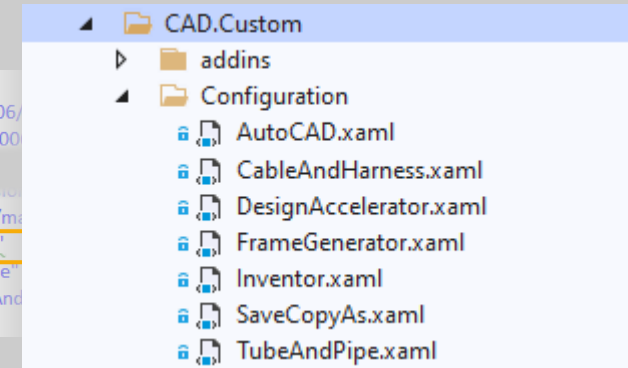
- `Style="{DynamicResource ThemedWindowStyle}"`

Note – the path to this resource won't validate in your Visual Studio environment

Data Standard Dialog-/Datasheet Types

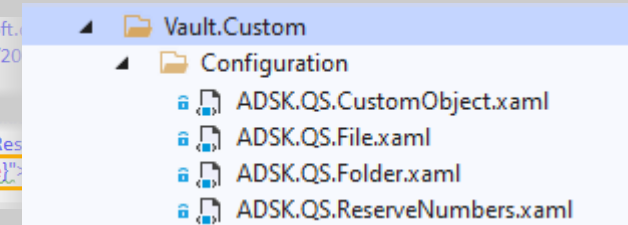
- VDS CAD Dialogs

```
<WPF:DSWindow  
  xmlns="http://schemas.microsoft.com/winfx/2006/05/xaml" xmlns:x="http://schemas.microsoft.com/winfx/2006/05/xaml" xmlns:d="http://schemas.microsoft.com/expression/2010/dynamicmetadata-types" xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006" Style="{DynamicResource ThemedWindowStyle}" x:Name="AutoCADWindow" ShowInTaskbar="True" ResizeMode="NoResize" SizeToContent="WidthAndHeight" />  
<WPF:DSWindow.Resources>
```



- VDS Vault Dialogs

```
<WPF:MainWindow xmlns="http://schemas.microsoft.com/winfx/2006/05/xaml" xmlns:x="http://schemas.microsoft.com/winfx/2006/05/xaml" x:Name="FileWindow" />  
  AllowsTransparency="False" ResizeMode="NoResize" Style="{DynamicResource ThemedWindowStyle}" />  
<Window.Resources>
```



Step 1c – All XAML Types

New Resource Dictionary

Embed all style definitions within a new node

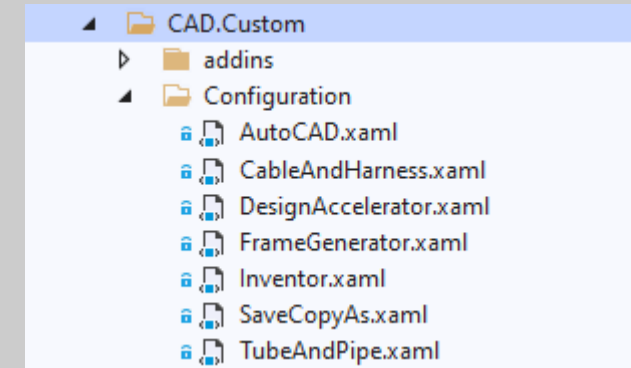
```
<ResourceDictionary>
```

```
</ResourceDictionary>
```

Data Standard Dialog-/Datasheet Types

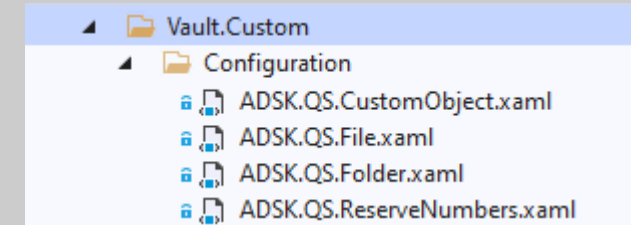
- VDS CAD Dialogs

```
<WPF:DSWindow.Resources>  
<ResourceDictionary>  
  <ResourceDictionary.MergedDictionaries>  
    <WPF:IsMoreThanZeroConverter x:Key="IsMoreThanZero" />  
    <WPF:IsLessThanZeroConverter x:Key="IsLessThanZero" />  
  </MergedDictionaries>  
</ResourceDictionary>  
</WPF:DSWindow.Resources>
```



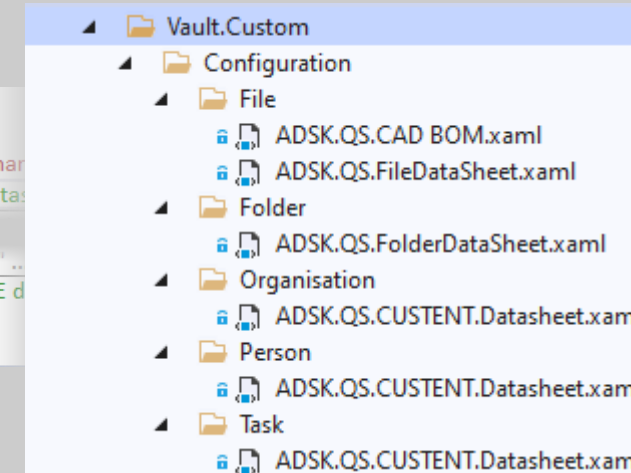
- VDS Vault Dialogs

```
<Window.Resources>  
<ResourceDictionary>  
  <ResourceDictionary.MergedDictionaries>  
    <WPF:IsMoreThanZeroConverter x:Key="IsMoreThanZero" />  
    <WPF:IsLessThanZeroConverter x:Key="IsLessThanZero" />  
  </MergedDictionaries>  
<!--#endregion themed styles for VE data-->  
<BooleanToVisibilityConverter x:Key="BooleanToVisibility" />  
</ResourceDictionary>  
</Window.Resources>
```



- VDS Vault Detail Tabs

```
<UserControl.Resources>  
<ResourceDictionary>  
  <ResourceDictionary.MergedDictionaries>  
    <!--#region themed styles for VE data-->  
    <Style TargetType="{x:Type Border}" />  
    <!--#endregion themed styles for VE data-->  
  </MergedDictionaries>  
</ResourceDictionary>  
</UserControl.Resources>
```



Step 1d – All XAML Types

New Dictionary Source

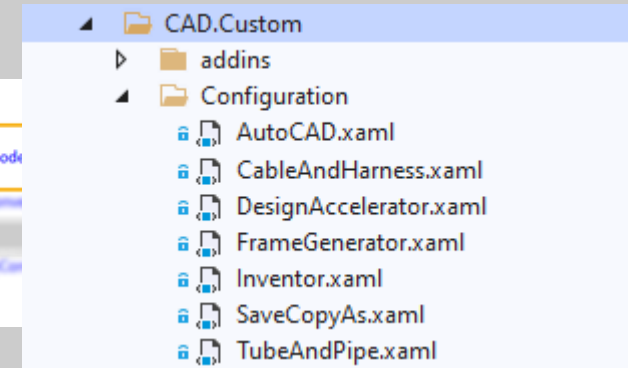
Add the dictionary providing all themed style definitions

- `<ResourceDictionary.MergedDictionaries>`
`<ResourceDictionary`
`Source="pack://application:,,,/Autodesk.`
`DataManagement.Client.Framework.Forms;co`
`mponent/Controls/WPF/ControlTemplates/Me`
`rgedResources.xaml" />`
`</ResourceDictionary.MergedDictionaries>`

Data Standard Dialog-/Datasheet Types

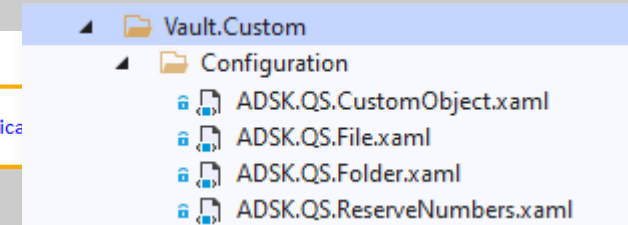
- VDS CAD Dialogs

```
<WPF:DSWindow.Resources>  
<ResourceDictionary>  
  <ResourceDictionary.MergedDictionaries>  
    <ResourceDictionary Source="pack://application:,,,/Autodesk.  
    DataManagement.Client.Framework.Forms;component/Controls/WPF/ControlTemplates/MergedResources.xaml" />  
  </ResourceDictionary.MergedDictionaries>  
</ResourceDictionary>  
</WPF:DSWindow.Resources>
```



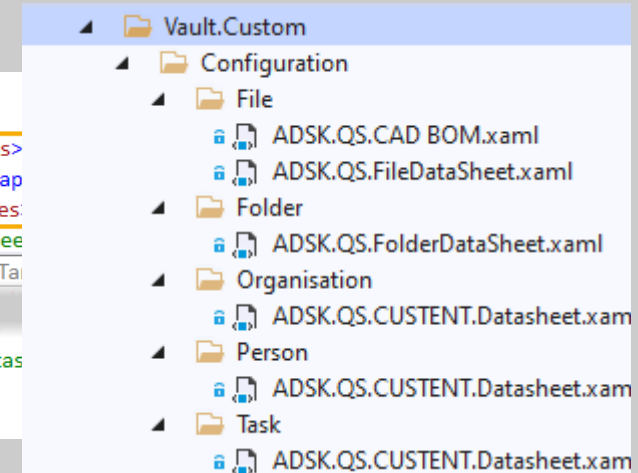
- VDS Vault Dialogs

```
<Window.Resources>  
<ResourceDictionary>  
  <ResourceDictionary.MergedDictionaries>  
    <ResourceDictionary Source="pack://application:,,,/Autodesk.  
    DataManagement.Client.Framework.Forms;component/Controls/WPF/ControlTemplates/MergedResources.xaml" />  
  </ResourceDictionary.MergedDictionaries>  
</ResourceDictionary>
```

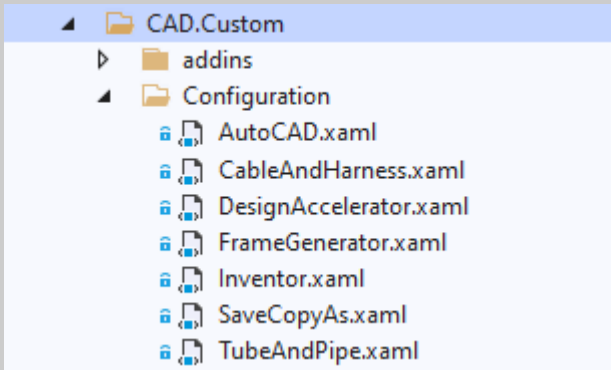


- VDS Vault Detail Tabs

```
<UserControl.Resources>  
<ResourceDictionary>  
  <ResourceDictionary.MergedDictionaries>  
    <ResourceDictionary Source="pack://application:,,,/Autodesk.  
    DataManagement.Client.Framework.Forms;component/Controls/WPF/ControlTemplates/MergedResources.xaml" />  
  </ResourceDictionary.MergedDictionaries>  
  <!--#region themed styles for VE datasheet-->  
  <Style x:Key="DataStandardLabelStyle" TargetType="TextBlock" .../>  
  <!--#endregion themed styles for VE datasheet-->  
</ResourceDictionary>  
</UserControl.Resources>
```



Data Standard Dialog-/Datasheet Types



- VDS CAD Dialogs

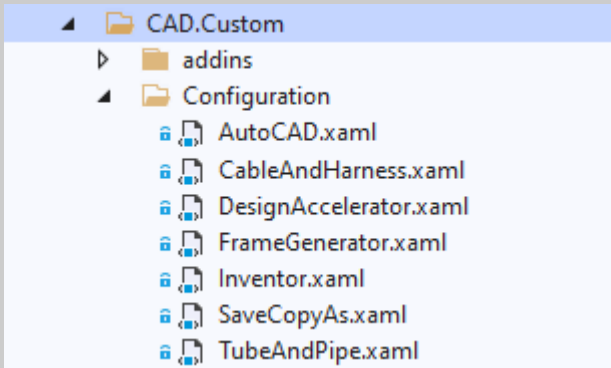
Step 2a – Control Styles

Update DynamicDataGridCellStyles

Re-base the cell style for label and value on themed cell styles

- Copy and Paste both styles from a matching CAD template
 - DynamicDataGridCellStyle
 - DynamicDataGridLabelStyle

Data Standard Dialog-/Datasheet Types



- VDS CAD Dialogs

```
<!--#region themed styles for CAD dialog standard controls-->
<Style TargetType="{x:Type Label}" BasedOn="{StaticResource}>...</Style>
<Style TargetType="{x:Type TextBox}" BasedOn="{StaticResource}>...</Style>
<Style x:Key="Required" TargetType="{x:Type TextBox}" BasedOn="{StaticResource}>...</Style>
<Style TargetType="{x:Type ListBoxItem}" BasedOn="{StaticResource ThemedListBoxItemStyle}"/>
<Style TargetType="{x:Type TreeView}" BasedOn="{StaticResource ThemedBaseControlStyle}"/>
<Style TargetType="{x:Type Border}">...</Style>
<!--#endregion themed styles for CAD dialog standard controls-->
```

Step 2b – Control Styles

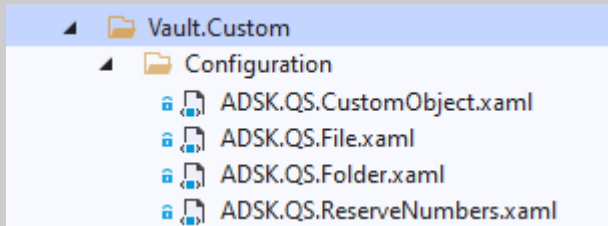
Re-base control styles for CAD

Re-base all control styles on themed control styles

- Add themed default control styles
 - Copy & Paste from: .\CAD\Configuration*.xaml files, installed with 2023 Update 2

- Derive named control styles from default ones if needed (see also cascading styles)

Data Standard Dialog-/Datasheet Types



- VDS **Vault** Dialogs

```
<!--#region themed styles for VE dialog standard controls-->
<Style x:Key="DataStandardLab" TargetType="{x:Type Label}" BasedOn="{StaticResource}>...</Style>
<Style TargetType="{x:Type Label}" BasedOn="{StaticResource DataStandardLabelStyle}"/>
<Style TargetType="{x:Type ListBoxItem}" BasedOn="{StaticResource ThemedListBoxItemStyle}"/>
<Style TargetType="{x:Type TreeView}" BasedOn="{StaticResource ThemedBaseControlStyle}"/>
<Style TargetType="{x:Type Border}">...</Style>
<!--#endregion themed styles for VE dialog standard controls-->
```

Step 2c – Control Styles

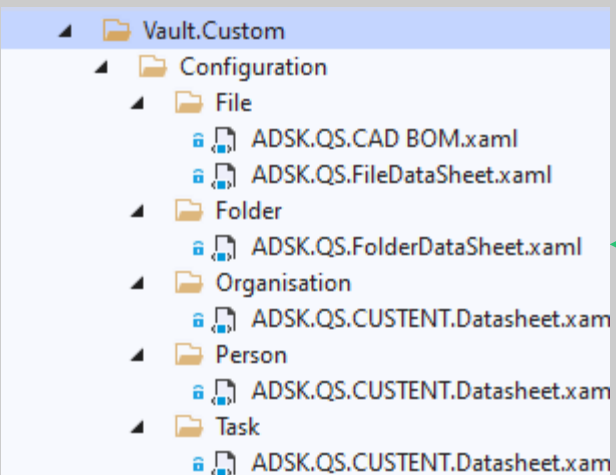
Re-base control styles for Vault Dialogs

Re-base all control styles on themed control styles

- Add themed default control styles
 - Copy & Paste from: .\Vault\Configuration*.xaml files, installed with 2023 Update 2
- Derive named control styles from default ones if needed (see also cascading styles)

Data Standard Dialog-/Datasheet Types

```
<!--#region themed styles for VE datasheet standard controls-->  
<Style x:Key="DataStandardLab" TargetType="{x:Type Label}" BasedOn="{StaticResource}>...</Style>  
<Style TargetType="{x:Type Label}" BasedOn="{StaticResource DataStandardLabelStyle}"/>  
<Style x:Key="DataStandardTex" TargetType="{x:Type TextBox}" BasedOn="{StaticResource}>...</Style>  
<Style TargetType="{x:Type TextBox}" BasedOn="{StaticResource DataStandardTextBoxStyle}"/>  
  
<Style TargetType="{x:Type ListBoxItem}" BasedOn="{StaticResource ThemedListBoxItemStyle}"/>  
<Style TargetType="{x:Type TreeView}" BasedOn="{StaticResource ThemedBaseControlStyle}"/>  
<Style TargetType="{x:Type Border}">...</Style>  
<!--#endregion themed styles for VE datasheet standard controls-->
```



- VDS Vault Detail Tabs

Step 2d – Control Styles

Re-base control styles for Vault Detail Tabs

Re-base all control styles on themed control styles

- Add themed default control styles
 - Copy & Paste from: .\Vault\Configurations\File, Folder, or Task - *.Datasheet.xaml

- Derive named control styles from default ones if needed (see also cascading styles)

Step 3 – Static Overrides

Remove static style overrides

Reset/remove static values for

- Background, (Foreground) colors
- BorderBrush, BorderThickness values

⇒ You will need to review each control individually, e.g., a TextBox displaying a status message may keep its static yellow color

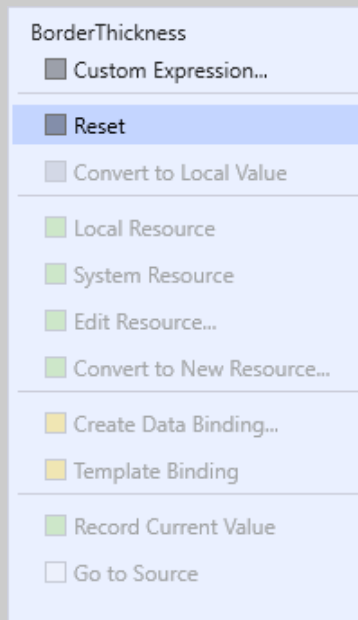
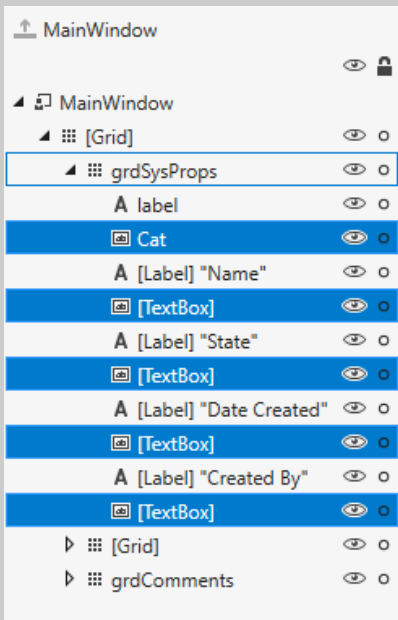
⇒ If you are sure that no individual overrides are required, Visual Studio XAML Designer allows you to multi-select control types and apply the “Reset” option in the Properties pane for all.

```
<WPF:MainWindow xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:WPF="clr-namespace:CreateObject.WPF;assembly=CreateObject"
  xmlns:wpfUtilities="clr-namespace:Autodesk.DataManagement.Client.Framework.Forms.Controls.WPF.Utilities;assembly=Autodesk.DataManagement.Client.Framework.Forms.Controls.WPF.Utilities"
  xmlns:winFormTheme="clr-namespace:Autodesk.DataManagement.Client.Framework.Forms.SkinUtils;assembly=Autodesk.DataManagement.Client.Framework.Forms.SkinUtils"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  xmlns:glob="clr-namespace:System.Globalization;assembly=mscorlib"
  Style="{DynamicResource ThemedWindowStyle}"
  x:Name="CustomObjectWindow" Background="#FF0F0F" SizeToContent="WidthAndHeight" ResizeMode="NoResize">
</Window.Resources>
```

Delete to enable theme color

```
<ComboBox x:Name="Categories"
  DisplayMemberPath="Name"
  SelectedItem="{Binding DynamicPropertiesCategory}"
  SelectedValuePath="Name"
  SelectedValue="{Binding Prop[_Category].Value}"
  ItemsSource="{Binding PsList[GetCategories]}"
  IsTabStop="False"
  BorderBrush="#FFF0F0" BorderThickness="0,1,1,0" />
```

Delete to enable theme color



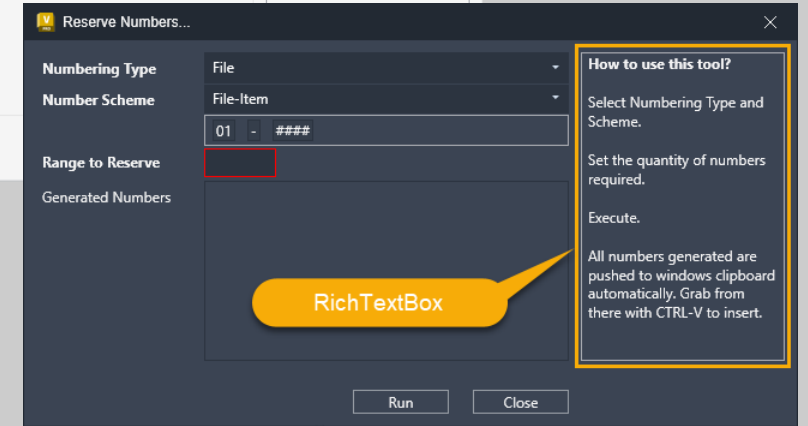
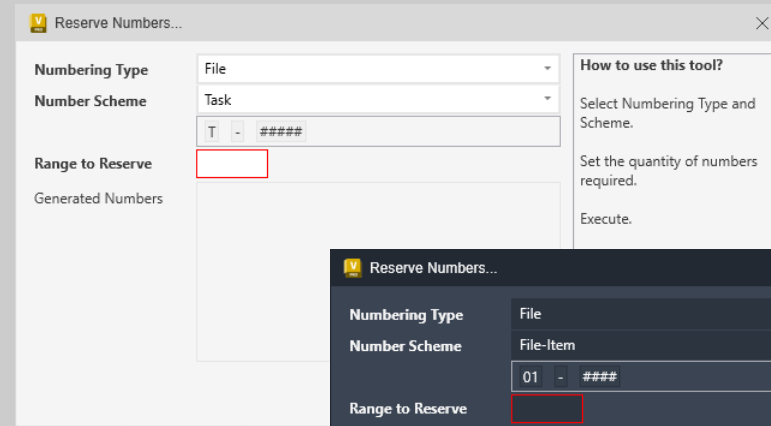
Step 4 – Custom Style

Themed base style

Create a themed style for controls that cannot derive from Vault

Themed<ControlType>Style definitions

- Example – The VDS sample tool **Reserve Numbers** uses a RichTextBox control
- **Solution Option 1**
 - Assign a base control style to reflect matching Back- and Foreground colors



```
<RichTextBox Grid.Column="2" Margin="10,0,0,0" Grid.RowSpan="5" Width="170">  
  <RichTextBox.Resources>  
    <Style TargetType="RichTextBox" BasedOn="{StaticResource ThemedBaseControlStyle}" />  
  </RichTextBox.Resources>  
</RichTextBox>
```

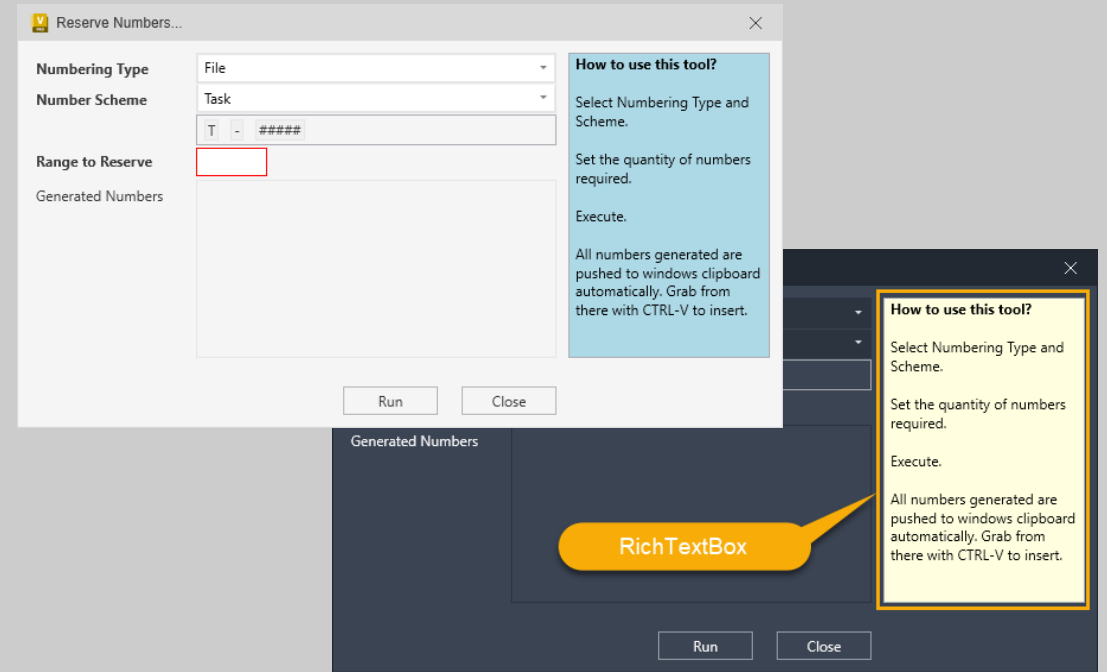
Step 4 – Custom Style

Themed custom style

Create a themed style for controls that cannot derive from Vault

Themed<ControlType>Style definitions

- Example – The VDS sample tool **Reserve Numbers** uses a RichTextBox control
- **Solution Option 2**
 - Create triggered custom style, if alternate colors are expected for the Dark and Light theme



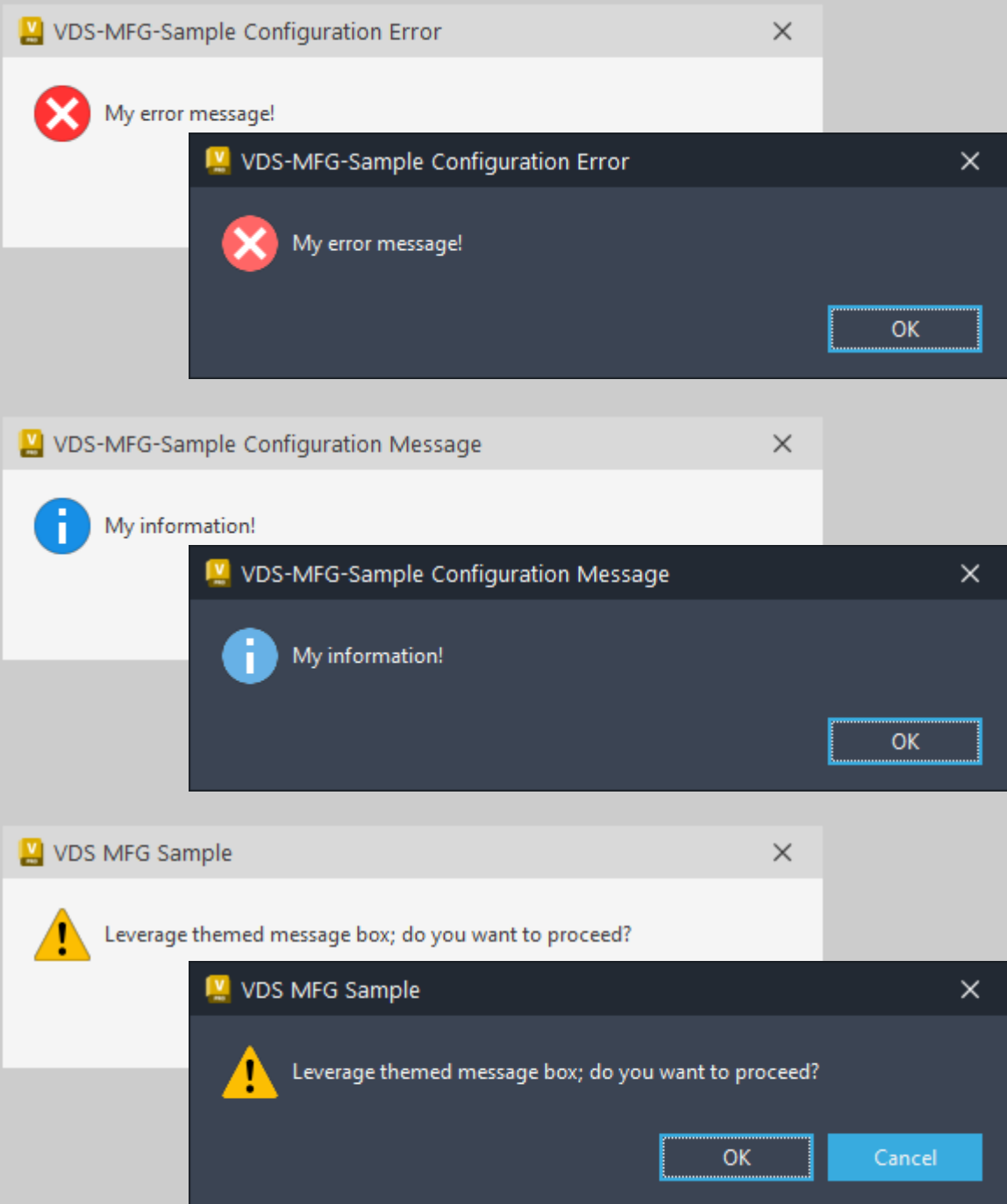
```
<RichTextBox Grid.Column="2" Margin="10,0,0,0" Grid.RowSpan="5" Width="170">
  <RichTextBox.Resources>
    <Style TargetType="RichTextBox">
      <Style.Triggers>
        <DataTrigger Binding="{Binding Path=CurrentTheme,
          Source={x:Static wpfUtilities:ActiveThemeProvider.Instance}}" Value="{x:Static winFormTheme:Theme.Default}">
          <Setter Property="Background" Value="□"LightGray"/>
        </DataTrigger>
        <DataTrigger Binding="{Binding Path=CurrentTheme,
          Source={x:Static wpfUtilities:ActiveThemeProvider.Instance}}" Value="{x:Static winFormTheme:Theme.Dark}">
          <Setter Property="Background" Value="□"LightYellow"/>
        </DataTrigger>
        <DataTrigger Binding="{Binding Path=CurrentTheme,
          Source={x:Static wpfUtilities:ActiveThemeProvider.Instance}}" Value="{x:Static winFormTheme:Theme.Light}">
          <Setter Property="Background" Value="□"LightBlue"/>
        </DataTrigger>
      </Style.Triggers>
    </Style>
  </RichTextBox.Resources>
</RichTextBox>
```

System Forms

Information & Error Messages

Leverage the pre-defined and theme-enabled ShowError, ShowMessage, and ShowWarning objects

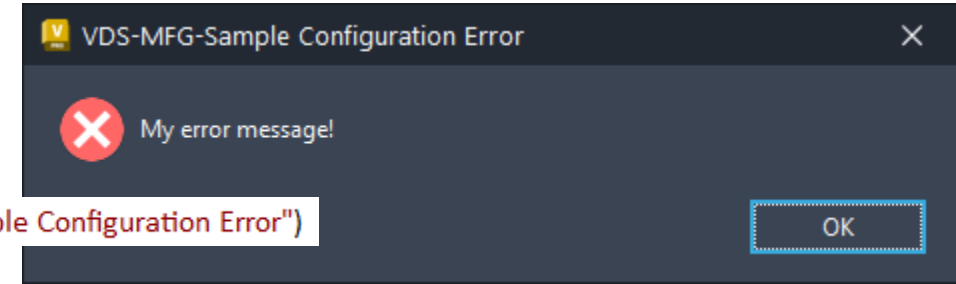
- Replace `[System.Windows.MessageBox]::Show()`
- With matching message type forms
 - `ShowError("Text", "Title")`
 - `ShowMessage("Text", "Title", "OK")`
 - `ShowWarning("Text", "Title", "OKCancel")`(see next slide for full syntax samples)



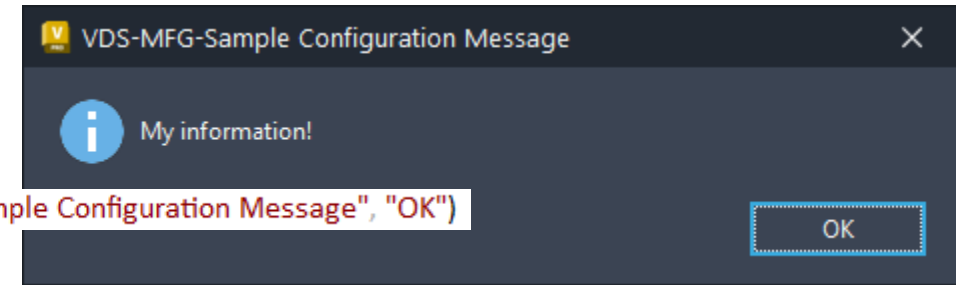
System Forms

Information & Error Messages

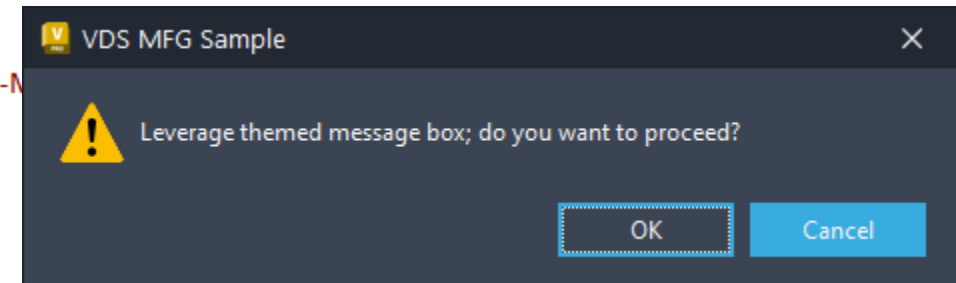
```
[Autodesk.DataManagement.Client.Framework.Forms.Library]::ShowError("My error message!", "VDS-MFG-Sample Configuration Error")
```



```
[Autodesk.DataManagement.Client.Framework.Forms.Library]::ShowMessage("My information!", "VDS-MFG-Sample Configuration Message", "OK")
```



```
$result = [Autodesk.DataManagement.Client.Framework.Forms.Library]::ShowWarning("Leverage themed message box; do you want to proceed?", "VDS MFG Sample", "OKCancel")  
if ($result -eq "OK") {  
    [Autodesk.DataManagement.Client.Framework.Forms.Library]::ShowMessage("Thank you for hitting 'OK'", "VDS-MFG-Sample Configuration Message", "OK")  
}  
else {  
    [Autodesk.DataManagement.Client.Framework.Forms.Library]::ShowMessage("Canceling information!", "VDS-MFG-Sample Configuration Message", "OKCancel")  
}
```



Current Theme

XAML | PowerShell

- How to access the current (active) scheme?

- XAML

- Theme.Default (Available in Vault Client only. Display name = "Classic")
- Theme.Dark
- Theme.Light

```
<DataTrigger Binding="{Binding Path=CurrentTheme, Source={x:Static wpfUtilities:ActiveThemeProvider.Instance}}"  
    Value="{x:Static winFormTheme:Theme.Dark}">
```

- PowerShell

- "Default" | 0 (Available in Vault Client only. Display name = "Classic")
- "Dark" | 1
- "Light" | 2

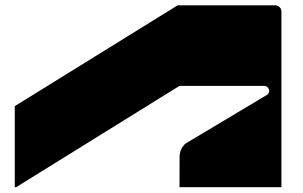
```
#how to access the current scheme information?
```

```
$ActiveScheme = [Autodesk.DataManagement.Client.Framework.Forms.SkinUtils.WinFormsTheme]::Instance.CurrentTheme
```

Troubleshooting

What if ...

- ... the window background does not adopt the dark or light scheme
 - ... a control's background does not adopt the dark or light scheme
 - ... a border is unexpectedly colored or incomplete
 - ... static non-conformant property fields don't highlight red borders
- ... the window background does not adopt the dark or light scheme
 - Static override in Window attributes
 - Missing Style reference of ThemedWindowStyle
 - ... a control's background does not adopt the dark or light scheme
 - Static Background color attribute at the control level
 - Style reference at control level to a style not based on a Themed<ControlType>Style
 - An explicit styles is applied to the individual control definition
 - ... a border is unexpectedly colored or incomplete
 - Static BorderBrush attribute at the control level
 - Style definition or reference at control level not based on Themed<ControlType>Style
 - ... static non-conformant property fields don't highlight red borders
 - Missing validated Binding WPF:ValidatedBinding
 - Missing style trigger in referenced style (e.g., default XAML templates delivered with 2023.1 missed this style extension)



Themes | Styles

Technology Crash Course

Themes | Styles

Introduction

- Theme = commonly applied set of styles
 - Windows UI Themes
 - Light | Dark | High Contrast
 - Autodesk Vault Client Themes
 - Classic | Dark | Light
 - Autodesk Inventor/AutoCAD Themes
 - Dark | Light
- Style = set of object attributes for visual appearance and layout
 - Colors, BorderBrush/-Thickness
 - Icons/Images
 - Margins
 - ...

The screenshot shows the Microsoft Docs page for 'Style for Windows apps'. The page title is 'Style for Windows apps' and it includes a breadcrumb trail: 'Docs / Windows / Apps / Desktop / Design /'. The page content is organized into four main sections: Color, Typography, Icons, and Acrylic. Each section has a brief description and a corresponding visual example. The 'Color' section shows a gradient of blue shades. The 'Typography' section shows the letters 'Aa' in a white serif font on a black background. The 'Icons' section shows three cloud icons in different colors (blue, orange, white) on a dark blue background. The 'Acrylic' section shows three overlapping rectangular shapes in red, orange, and white on a dark red background. The page also features a left-hand navigation menu with a search bar and a list of topics including 'Build desktop apps', 'Design', 'Motion', 'Shell', 'Input and interactions', 'Devices', 'Usability', 'Develop', 'Deploy', and 'API reference'.

XAML Styles

Introduction

- Styles let you set control properties and re-use those settings for a consistent appearance across multiple controls.
- Use styles if ...
 - If **multiple or all** controls share an attribute
 - If one or multiple **conditions** should enable/disable or change attributes
- Style definitions are applied in descending order, e.g.,
 - 1) The default properties “IsEnabled” and “ToolTip” are set first
 - 2) If the condition is true, the default setting for “IsEnabled” will change, whereas the ToolTip doesn’t



```
<Style x:Key="btnScEnabled" TargetType="{x:Type Button}" BasedOn="{StaticResource ThemeDefaultStyle}">
  <Setter Property="IsEnabled" Value="True" />
  <Setter Property="ToolTip" Value="{Binding UIString[MSDCE_TT20]}" />
  <Style.Triggers>
    <DataTrigger Binding="{Binding Text, ElementName=txtNewShortCut}" Value="">
      <Setter Property="IsEnabled" Value="False" />
    </DataTrigger>
  </Style.Triggers>
</Style>
```

1

2

Implicit | Explicit Style

Scope of styles

- An implicit style defines a target control type only
 - Any target control type will adapt the style
- An explicit style defines a target control type and a unique key attribute **x:Key**
 - A target control type needs to bind to the style by referencing the unique style key

```
<Style TargetType="{x:Type Label}" BasedOn="{StaticResource ThemedLabelStyle}">
```

```
<Style x:Key="DataStandardLabelStyle" TargetType="{x:Type Label}" BasedOn="{StaticResource ThemedLabelStyle}">  
  <Style.Triggers>  
    <DataTrigger Binding="{Binding Content, RelativeSource={RelativeSource Self}}" Value="{x:Null}">  
      <Setter Property="BorderThickness" Value="1" />  
      <Setter Property="BorderBrush" Value="■ #ff0000" />  
    </DataTrigger>  
  </Style.Triggers>  
</Style>
```

```
<Label Content="Path" Style="{StaticResource DataStandardLabelStyle}" Grid.
```

Cascading Styles

Inherit styles

- Create an explicit style to inherit base settings to other styles
- Example
 - 1) The style “*DataStandardLabelStyle*” derives from the global themed control type style “*ThemedLabelStyle*” and adds a trigger/attributes
 - 2) The implicit style derives from 1) and targets all labels of this dialog/datasheet
 - 3) The explicit style “*IblRequired*” derives from 1) and will be applied by controls that reference this style explicitly
- Note – Inheritance works top-down; put the general style before the deriving one.

```
<Style x:Key="DataStandardLabelStyle" TargetType="{x:Type Label}" BasedOn="{StaticResource ThemedLabelStyle}"> 1
  <Style.Triggers>
    <DataTrigger Binding="{Binding Content, RelativeSource={RelativeSource Self}}" Value="{x:Null}">
      <Setter Property="BorderThickness" Value="1" />
      <Setter Property="BorderBrush" Value="■" #ff0000" />
    </DataTrigger>
  </Style.Triggers>
</Style>
<Style TargetType="{x:Type Label}" BasedOn="{StaticResource DataStandardLabelStyle}" /> 2
```

```
<Style x:Key="IblRequired" TargetType="{x:Type Label}" BasedOn="{StaticResource DataStandardLabelStyle}"> 3
  <Setter Property="TextBlock.FontWeight" Value="Bold" />
  <Setter Property="ToolTip" Value="An input/selection is required to generate numbers" />
</Style>
```

Re-use Styles

Window Scope | Configuration Scope

- The default scope of any style is the containing XAML file, the individual Window or UserControl definition
- Styles can be shared across multiple windows or control definitions
 - Create a style resource file as a shareable definition, e.g., “C:/.../CommonStyle.xaml”
 - Reference the style resource for re-use as a merged resource dictionary
 - Use absolute path references for ‘Source’ (xaml files are loaded at runtime, and the loader can’t share its root path as a default)

CommonStyle.xaml

```
1 <ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
2 xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
3 xmlns:WPF="clr-namespace:CreateObject.WPF;assembly=CreateObject"
4 xmlns:wpfUtilities="clr-namespace:Autodesk.DataManagement.Client.Framework.Controls.WPF.Utilities;assembly=Autodesk.D
5 xmlns:winFormTheme="clr-namespace:Autodesk.DataManagement.Client.Framework.Forms.SkinUtils;assembly=Autodesk.DataManag
6 <Style x:Key="MyThemedTextBoxStyle" TargetType="{x:Type TextBox}" >
7
8 <Style.Triggers>
9 <DataTrigger Binding="{Binding Path=CurrentTheme,
10 Source={x:Static wpfUtilities:ActiveThemeProvider.Instance}}" Value="{x:Static winFormTheme:Theme.Default}">
11 <Setter Property="BorderBrush" Value="#FFABADB3"/>
12 <Setter Property="Background" Value="#FFF7F98" />
13 </DataTrigger>
14 <DataTrigger Binding="{Binding Path=CurrentTheme,
15 Source={x:Static wpfUtilities:ActiveThemeProvider.Instance}}" Value="{x:Static winFormTheme:Theme.Dark}">
16 <Setter Property="BorderBrush" Value="#FF323A48"/>
17 <Setter Property="Background" Value="#FFC3BF69" />
18 </DataTrigger>
19 <DataTrigger Binding="{Binding Path=CurrentTheme,
20 Source={x:Static wpfUtilities:ActiveThemeProvider.Instance}}" Value="{x:Static winFormTheme:Theme.Light}">
21 <Setter Property="BorderBrush" Value="#FFEBEBEB"/>
22 <Setter Property="Background" Value="#FFBAEC89" />
23 </DataTrigger>
24 </Style.Triggers>
25 </Style>
</ResourceDictionary>
```



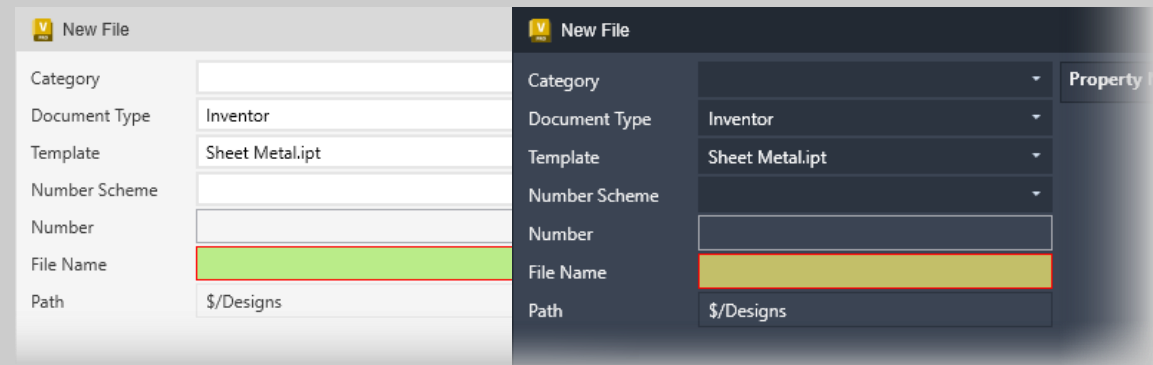
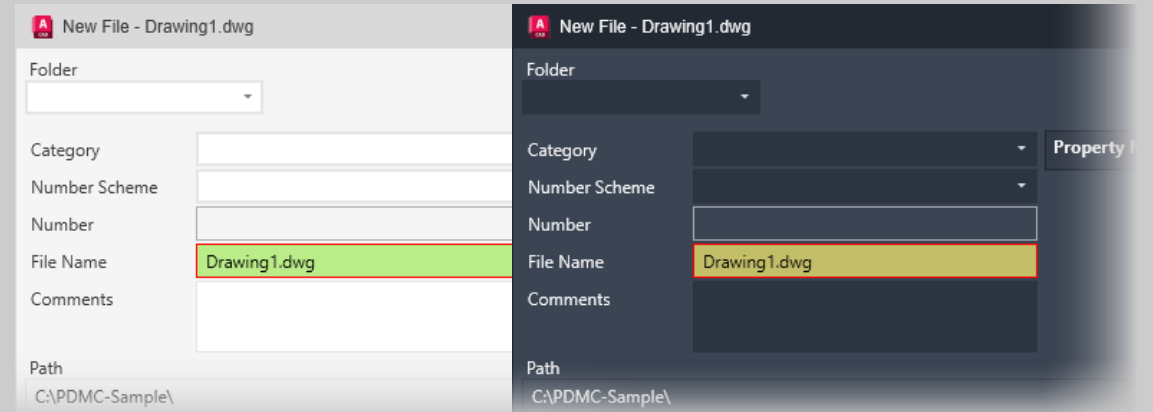
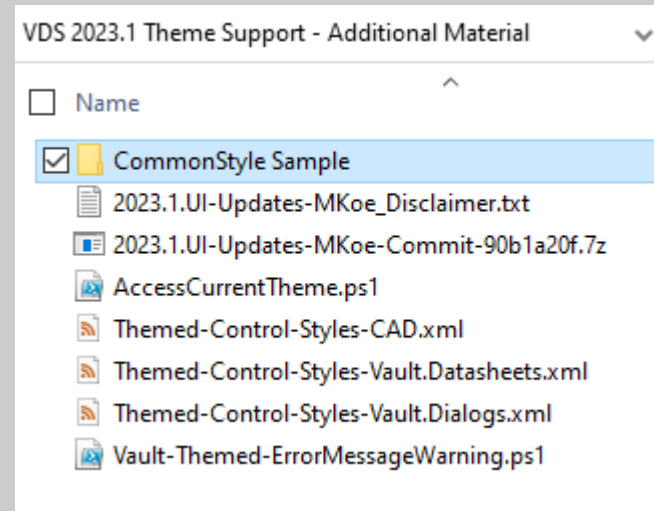
```
<ResourceDictionary>
  <ResourceDictionary.MergedDictionaries>
    <ResourceDictionary Source="C:\ProgramData\Autodesk\Vault.Custom\Configuration\CommonStyle.xaml" />
  </ResourceDictionary.MergedDictionaries>
</ResourceDictionary>
```



Sample

Re-Use Styles

- Additional Material\CommonStyle Sample\
 - Demonstrating
 - Central style definition for Vault, AutoCAD New File Dialog
 - DataTrigger for active theme
 - settings order – general -> specific
 - Background/Border color per Theme
 - BorderBrush override on ValidationError



Useful Links

XAML Styles

- <https://docs.microsoft.com/en-us/windows/apps/design/style/xaml-styles>
- https://www.tutorialspoint.com/xaml/xaml_styles.htm



Autodesk and the Autodesk logo are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and/or other countries. All other brand names, product names, or trademarks belong to their respective holders. Autodesk reserves the right to alter product and services offerings, and specifications and pricing at any time without notice, and is not responsible for typographical or graphical errors that may appear in this document.

© 2023 Autodesk. All rights reserved.