Customize and configure views

This article describes how to customize the look and feel and the text content in the Nexus Service Station application and how to make general configurations that applies to all views. For configuration of the individual views, see the separate view pages.

There are three files used for configuration of Nexus Service Station:

Type of file	File name	Folder
Style file	*Style.xaml* (default values) or *Style.User.xaml* (user specific overrides)	*Program Files (x86)\Technology Nexus\Nexus Service Station*
Locale file	*Locale.xx-XX.xaml* (default values) or *Locale.xx-XX.User.xaml* (user specific overrides) "xx-XX" is the name of each locale folder	*Program Files (x86)\Technology Nexus\Nexus Service Station\locales* Each locale available will have its own sub folder, named *xx-XX* (for example, *sv-SE* for Swedish text content).
Configuration file	App.config Includes settings for the views	*Program Files (x86)\Technology Nexus\Nexus Service Station*

Expand/Collapse All

Prerequisites

- You must have administrator privileges.

Individual configuration of views

The views can be configured in different ways. For information about how to configure each individual view, see Views.

General configuration of views

The default styling of the Nexus Service Station application is included in the *Style.xaml* file, located in the installation folder *Program Files (x86)\Technology Nexus\Nexus Service Station*.

1. To override any styles in the application, copy single style keys from *Style.xaml* to *Style.User.xaml* and make the modifications there, leaving *Style.xaml* untouched.

⚠️

ⓘ

⚠ Do not make any modifications to *Style.xaml* as it will be overwritten on application upgrades.

ⓘ
- You can only have one *User Style* file.
- The *User Style* file should only contain customized styles and not be a full copy of the original, the application will fall back to the original for the missing styles.
- The *User Style* file is added by the installer if it does not exist, but never overwritten.
- The *User Style* file will be deleted by the installer when uninstalling the application.

To reset completely to the default style, delete everything between the tags `<ResourceDictionary …>` and `</ResourceDictionary>` in the *User Style* file.

The font in all views is set by overriding the **Global_Font** key in the *User Style* file.

- The banner above each view is set by overriding the **Image_Banner key** in the *User Style* file.
- The top and bottom banner gradient in all views is set by overriding the **MainWindow_GradientPrimary** and **MainWindow_GradientSecondary** keys in the *User Style* file.

- The background color of the Language view is set by overriding the **Brush_Bg_LanguageSelection** key in the *User Style* file.
- The background color in the other views is set by overriding the **MainWindow_Background** key in the *User Style* file.

These are the predefined button values that can be set for **leftButton**, **middleButton** and **rightButton**. They can be extended or overridden in the *User Style* file.

- Button_Back
- Button_Block
- Button_Cancel
- Button_Continue
- Button_Language
- Button_Login
- Button_Logout
- Button_Ok
- Button_TakePhoto
- Button_Yes
- Button_No
- Button_UpdateCard
- Button_EjectCard
- Button_Change
- Button_Redo
- Button_Confirm
- Button_Finish

⚠

The default text content of the Nexus Service Station application is included in *locale* files located in the folder *Program Files (x86)\Technology Nexus\Nexus Service Station\Vocales*. Each locale available will have its own sub folder.

1. To override any text content in the application, copy single string keys to the applicable user text content file and make the modifications there, leaving *Locale. xx-XX.xaml* untouched.

> ⚠️ Do not make any modifications to *Locale.xx-XX.xaml* files as they will be overwritten on application upgrades

> ⓘ
> - You can only have one user text content file per original locale file.
> - The user text content files should only contain customized text content and not be a full copy of the original, the application will fall back to the original for the missing text content.
> - All user text content files are added by the installer if they do not exist, but never overwritten.
> - All user text content files will be deleted by the installer when uninstalling the application.

To reset completely to the default text content, delete everything between the tags `<ResourceDictionary …>` and `</ResourceDictionary>` in the user text content file.

An app flow contains views (visible to the user to interact with). There must always exist an app flow called "default". The reason for this is if something fails in the application and it can not find the flow it is looking for, it will jump back to the flow called "default."

These are the settings made when an app flow is added.

Setting	Possible Values	Mandatory	Description
name	Any text	Yes	The name of the view flow
menuItem	A string key in the *Locale* file	No	The name of a string key in the *Locale* file which contains the name to display on the link to this view flow from the Menu view.
description	A string key in the *Locale* file	No	The name of a string key in the *Locale* file which contains the description to display on the link to this view flow from the Menu view.

These are mandatory settings that are used by all views.

Setting	Possible Values	Description
ReturnToStartTimout	Any positive integer	Time of inactivity (in seconds) before returning to splash screen (start view).
ViewProcessTimeout	Any positive integer	The maximum number of seconds to wait for a view to finish processing a request before timing out.
Camera	"Canon"	The camera to use.
PhotoCountdown	Any positive integer	The countdown, in seconds, before a photo is taken.
PhotoQuality	Any positive integer	Used to determine the quality level when converting RAW files to JPG, defaults to 80 if not specified.

ReadLayoutName	Card SDK Layout	The name of a Card SDK layout that can be retrieved from the service provider to read cards in the Update Card View.
RewriteLayoutName	Card SDK Layout	The name of a Card SDK layout that can be retrieved from the service provider to update the rewrite surface of cards in Update Card View.
clientName	Any text	Name the application will use to identify itself to the IDC web service.
username	Web service username	The username used in the IDC web service.
password	Web service password	The password used in the IDC web service.
CameraRunsBeforeRestart	Any positive integer	When the specified number of photos have been taken, the next time the splash screen is loaded, the entire application is restarted.
CameraRunsBeforeForcedRestart	Any positive integer	Same as above, but is checked when loading the Login view instead.
crlTimeToLive	Any positive integer	How often to update the CRL files, in minutes, 0 means never update.
crlsToCache	Comma-separated list of URLs	URLs to continuously download and cache CRL files from.
TrimRegisterFields	true, false	If true field values in Register views will be trimmed of whitespace before validated/saved.

When a view is added to an app flow by using the **<add...>** tag, these settings (that are shared by all views) can be made:

Setting	Possible Values	Mandatory	Description
leftButton	All Button Values	No	Text on the left side button.
middleButton	All Button Values	No	Text on the middle button.
rightButton	All Button Values	No	Text on the right side button.
returnTo	- View: <view_name> - AppFlow: <app_flow_name> - Window: <window_name> - Previous	Yes	Returns to: - The given view - The given app flow - The given window - The previous view in the flow
skipIf	A Condition Key	No	If the specified condition evaluates to true, this view is skipped and the next view in the flow is shown instead
parameters	See Views	See parameter table for each view	Custom parameters for this view.

The **<add>** tag can also define a child-tag, **<next...>**. This tag has the following settings:

Sett ing	Possible Values	Mand atory	Description
con tent	- View: <view_name> - AppFlow: <app_flow_nam e> - Window: <window_name>	Yes	Sets Next to: - The given view - The given app flow - The given window
if	A Condition Key	No	If the specified condition evaluates to true the view, app flow or window specified in **content** is loaded

An IDC condition key that evaluates to true or false.