User Interface - Bug #8074

Preferred UI Theme not selected in web with auto-login and forked sessions

11/23/2023 06:48 AM - Galya B

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Galya B	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD		

Description

Related issues:

Related to User Interface - Feature #4856: web client session forking	**Closed**
Related to User Interface - Feature #3931: single sign-on for virtual desktop...	**Closed**

History

#1 - 11/23/2023 06:50 AM - Galya B

The flows of auto-login and forking a new session in web bypass the login screen, thus a preferred UI theme can't be selected by the end user. On first login the theme can be stored in localStorage and retrieved directly by the client. The selected theme should still be validated against the list of declared themes in directory. When no theme is present in localStorage, the default theme should be used.

Also, you were mentioning before there are supposed to be changes related to themes. Should I plan to fix this issue soon.

#2 - 11/23/2023 09:06 AM - Greg Shah

- Description updated

> you were mentioning before there are supposed to be changes related to themes.

Can you remind me of where/when I said this? I'm not recalling the discussion.

> Should I plan to fix this issue soon.

Yes, I think it is important. The user would certainly expect that the theme in the new session would be consistent with the original session.

#3 - 11/23/2023 09:07 AM - Greg Shah

- Related to Feature #4856: web client session forking added

#5 - 11/23/2023 09:09 AM - Galya B

Greg Shah wrote:

> > you were mentioning before there are supposed to be changes related to themes.
>
> Can you remind me of where/when I said this? I'm not recalling the discussion.

If I was able to recall it, I wouldn't ask... I guess I got it wrong.

#7 - 11/23/2023 09:11 AM - Galya B

- Related to Feature #3931: single sign-on for virtual desktop mode added

#8 - 11/23/2023 09:17 AM - Greg Shah

Some customer applications do use their own subclasses of our themes. When doing so, these themes are registered in a list in the directory and are located in the application jar file(s). At that point these custom themes are just as valid as any other theme.

The other thing to note (this may be what you were thinking about) is that I was envisioning that we might have other non-theme options that we would want to expose out to the login dialog. For example, it would be very useful to be able to specify other bootstrap cfg values like the startup program. This is not too difficult to do, but we haven't done it because of the security implications. We would certainly need to check/filter these quite carefully as well as provide a security manager resource to handle the decisions. It is a big enough effort that we have not moved ahead with it.

#9 - 11/23/2023 09:22 AM - Galya B

I ponder on more trivial matters like: Should I break all current prod client storage configs and add the storageId to the path (key) to enable user Preferences to be specific to the end user and this way I'll be able to directly use ClientStorage for storing the themes, or just go with localStorage (which is user specific with sso) no matter the configured client storage...

#10 - 11/23/2023 09:27 AM - Galya B

Maybe now it's the right time to do it, since we've just launched the methods accessing client storage, up till now there should have been only enhanced web configs there... which we already broke by changing the default client storage to java preferences.

#11 - 11/23/2023 09:32 AM - Greg Shah

I agree that we don't have to worry about the existing production cfg values. Right now is the best time to make any breaking changes.

Where is the storageId coming from?

#12 - 11/23/2023 09:36 AM - Galya B

Greg Shah wrote:

> Where is the storageId coming from?

Hopefully from well written SsoAuthenticator implementation. On authenticate the result should provide unique user specific id for the storage... I hope the sample I've seen in #4571 didn't go in the same format in production. The effort to write the javadoc for the interface was substantial.

#13 - 11/23/2023 09:38 AM - Greg Shah

The solution should not be specific to SSO. Why not just use the FWD account name or some UUID associated with the FWD account?

#14 - 11/23/2023 09:47 AM - Galya B

This storageId should ensure the same client process reused by many users can store end user specific info in independent sections of the storage.

Currently it's up to the customers to decide on the key. The enhanced browse configs generate a key that includes the FWD user, but with some customers all end users are associated the same FWD user -> the same configs. It was working fairly well by having them stored in the browser (localStorage) and hoping no two end users use the same browser instance, but having them in the user Preferences changes things and I was hoping on SSO. Obviously the customers can write keys that are composite themselves, but if they migrate apps directly to FWD this is likely not going to be the case.

I'm not sure if I've laid the issue clear enough. I think having sso and storageId is pretty important and I don't see a valid alternative for customers reusing the same fwd user for all web clients.

#15 - 11/23/2023 09:59 AM - Galya B

The core issue is that in the original OE 'end user' == 'OS user' and client storage (registry, ini) rarely clashes between users. FWD web allows more freedom. Although java user Preferences are OS user specific, the customer doesn't define a separate OS user for each end user.

#16 - 11/23/2023 10:19 AM - Galya B

So I'm seeing the options as:
1. All users happily share the same configs and override each other;
2. The customer rewrites the storage methods calls to use composite unique keys and only enhanced browse is left shared;
3. The customer defines separate FWD users for each end user and it gets incorporated in ClientStorage (user Preferences and localStorage);
4. The customer shares unique id for each end user and it gets incorporated in ClientStorage (user Preferences and localStorage);
And we should take into consideration the combination of different visions of the different customers.

#17 - 11/24/2023 05:57 AM - Galya B

- Assignee set to Galya B

- Status changed from New to WIP

Since we probably don't have enough control over the customer behavior, then the solution I propose is to always incorporate the fwd user and the storageId in the key / path. I think this should also apply to registry, since I'm not happy with the place we put the keys - directly in the root registry.

#18 - 11/27/2023 04:59 AM - Galya B

If we want to support multiple users in one browser, then the theme drop-down on the login page should never have a preselected theme, because the users are anonymous before login. If they've selected a preferred theme on a previous login, it's saved in ClientStorage and should be restored after the login. So it's not correct to preselect in the drop-down at any time. An explicit choice is to be sent to the client and saved in the client storage. If none, first check the client storage, then theme/name and eventually fallback to Windows10Theme.

#19 - 11/27/2023 09:43 AM - Galya B

- % Done changed from 0 to 50

#20 - 11/28/2023 08:09 AM - Galya B

I'm working on isolating the users in the client storage, but I forgot that the enhanced browse configs can be applied to all users! This design is not standing well with allowing the customers to write & read anything to that same storage (with introduction of PUT-KEY-VALUE, etc.), where isolation is needed, especially since the storage can be user Preferences. The assumption that the enhanced configs are always stored in the browser and there is only one end user (person), is no longer valid.

Ref. EnhancedBrowseSaveTarget:

```
         case GLOBAL_DEFAULT:
            return "All Browses, All Users";
         case BROWSE_SPECIFIC_DEFAULT:
            return "This Browse, All Users";
```

Greg, what do you think?

#21 - 11/28/2023 10:12 AM - Galya B

- % Done changed from 50 to 80

#22 - 11/29/2023 03:04 PM - Greg Shah

> Since we probably don't have enough control over the customer behavior, then the solution I propose is to always incorporate the fwd user and the storageId in the key / path. I think this should also apply to registry, since I'm not happy with the place we put the keys - directly in the root registry.

OK.

> > Where is the storageId coming from?

> Hopefully from well written SsoAuthenticator implementation. On authenticate the result should provide unique user specific id for the storage... I hope the sample I've seen in #4571 didn't go in the same format in production. The effort to write the javadoc for the interface was substantial.

What changes are needed in SsoAuthenticator to provide the storageId?

#23 - 11/29/2023 03:05 PM - Greg Shah

> I'm working on isolating the users in the client storage, but I forgot that the enhanced browse configs can be applied to all users! This design is not standing well with allowing the customers to write & read anything to that same storage (with introduction of PUT-KEY-VALUE, etc.), where isolation is needed, especially since the storage can be user Preferences. The assumption that the enhanced configs are always stored in the browser and there is only one end user (person), is no longer valid.

Anything that targets more than the current user must (by definition) be stored in the directory, not in user-specific preferences.

#24 - 11/29/2023 03:32 PM - Greg Shah

The value should be added to ClientParameters at startup so that it is always available at the server (without the need for a down-call).

#25 - 11/30/2023 01:40 AM - Galya B

Greg Shah wrote:

> What changes are needed in SsoAuthenticator to provide the storageId?

It is present in the interface since 3931a, and the dummy SampleSsoAuthenticator provides unique id for each user, but this customer #8052 for example returns null in their implementation. They need to understand that using the same OS user, same FWD account and same storageId can't provide isolation for their users.

> Anything that targets more than the current user must (by definition) be stored in the directory, not in user-specific preferences.

I want to disable the two targets for enhanced browse configs - All Browses, All Users and This Browse, All Users. I can't see this working with one user deciding for all users, since we've decided to keep enhanced browse configs on the client machine in web (with user Preferences). There are only two solutions: either we "degrade" the functionality by removing both options, or the configs don't use ClientStorage, but directly localStorage for web as before (it was decided against it in #4517-26) and in this case we ignore the case where multiple app users are logged-in from the same browser instance and need different configs. I need a confirmation on what it would be.

> The value should be added to ClientParameters at startup so that it is always available at the server (without the need for a down-call).

I'm not following here. What value?

#26 - 11/30/2023 08:18 AM - Greg Shah

>> Anything that targets more than the current user must (by definition) be stored in the directory, not in user-specific preferences.

> I want to disable the two targets for enhanced browse configs - All Browses, All Users and This Browse, All Users. I can't see this working with one user deciding for all users, since we've decided to keep enhanced browse configs on the client machine in web (with user Preferences).

This already requires "admin" rights to do it. We can't remove this otherwise how does a customer save installation-wide settings in the first place?

> There are only two solutions: either we "degrade" the functionality by removing both options, or the configs don't use ClientStorage, but directly localStorage for web as before (it was decided against it in #4517-26) and in this case we ignore the case where multiple app users are logged-in from the same browser instance and need different configs. I need a confirmation on what it would be.

Or, for these "admin" cases it gets stored in the directory and for the non-admin cases it does not.

>> The value should be added to ClientParameters at startup so that it is always available at the server (without the need for a down-call).

> I'm not following here. What value?

This was meant for #4203.

#27 - 11/30/2023 08:32 AM - Galya B

Greg Shah wrote:

> > There are only two solutions: either we "degrade" the functionality by removing both options, or the configs don't use ClientStorage, but directly localStorage for web as before (it was decided against it in #4517-26) and in this case we ignore the case where multiple app users are logged-in from the same browser instance and need different configs. I need a confirmation on what it would be.
>
> Or, for these "admin" cases it gets stored in the directory and for the non-admin cases it does not.

I was confused by the fact that it worked on the browser level (with localStorage), but if it's intended to target all users, this will be considered a fix. OK, extending the task with this one too.

#28 - 12/01/2023 09:13 AM - Galya B

My bad. The enhanced configs for all users are stored in directory, so this is not in conflict with isolating the user storage in ClientStorage. I need to complete the changes in the registry and the task will be done.

#29 - 12/04/2023 06:27 AM - Galya B

I'm trying to convert an example procedure from here.

Edited: Copy-paste issue. Conversion is fine now.

#30 - 12/04/2023 07:30 AM - Galya B

Actually there is another conversion issue with the example.

It has PUT-KEY-VALUE SECTION "PSC\Clue\TEMP-PRGM" KEY "DATA" VALUE "Here, this is the value in the registry". that gets converted to EnvironmentOps.setKeyValue("PSCClueTEMP-PRGM", "DATA", "Here, this is the value in the registry");. The backslashes are missing, deforming the path.

Is this fixable?

#31 - 12/04/2023 07:45 AM - Galya B

Also there is a problem in registry where the base_name is actually not used in the native code. The char is loaded and then released, so part of the path is missing.

I wonder if we can consider replacing the native code with user preferences that defaults to registry on Windows?

#32 - 12/04/2023 08:18 AM - Greg Shah

Galya B wrote:

> Actually there is another conversion issue with the example.
>
> It has PUT-KEY-VALUE SECTION "PSC\Clue\TEMP-PRGM" KEY "DATA" VALUE "Here, this is the value in the registry". that gets converted to EnvironmentOps.setKeyValue("PSCClueTEMP-PRGM", "DATA", "Here, this is the value in the registry");. The backslashes are missing, deforming the path.
>
> Is this fixable?

On what platform are you running the conversion?

Have you configured the p2j.cfg.xml to tell the conversion that the original code is from Windows GUI:

```
      <parameter name="opsys"            value="WIN32" />
      <parameter name="winsys"           value="MS-WINDOWS" />
```

#33 - 12/04/2023 08:20 AM - Greg Shah

> Also there is a problem in registry where the base_name is actually not used in the native code. The char is loaded and then released, so part of the path is missing.

Eugenie?

> I wonder if we can consider replacing the native code with user preferences that defaults to registry on Windows?

Yes, we can probably do that.

#34 - 12/04/2023 08:26 AM - Galya B

Greg Shah wrote:

> Galya B wrote:
>
> > Actually there is another conversion issue with the example.
> >
> > It has PUT-KEY-VALUE SECTION "PSC\Clue\TEMP-PRGM" KEY "DATA" VALUE "Here, this is the value in the registry". that gets converted to EnvironmentOps.setKeyValue("PSCClueTEMP-PRGM", "DATA", "Here, this is the value in the registry");. The backslashes are missing, deforming the path.
> >
> > Is this fixable?
>
> On what platform are you running the conversion?
>
> Have you configured the p2j.cfg.xml to tell the conversion that the original code is from Windows GUI:
>
> [...]

On Windows, since I'm testing the registry. Thanks, this will probably fix it.

By the way I have second thoughts on changing the registry keys to incorporate user id and storage id, because the global registry methods are supposed to access everything in the registry and not only to create an env. This means a FWD server running on Windows can have all web users sharing the same values if the customer code doesn't include user-specific segment in the path. And I'm generally confused how it could be done safe and proper at the same time.

#35 - 12/04/2023 08:43 AM - Eugenie Lyzenko

Greg Shah wrote:

> > Also there is a problem in registry where the base_name is actually not used in the native code. The char is loaded and then released, so part of the path is missing.
>
> Eugenie?

On the time it was written it worked. I think the base_name is ignored because we always used HKEY_CURRENT_USER as the base. And it was enough. But it was 10 years ago and Windows can change internals.

#36 - 12/04/2023 08:54 AM - Galya B

Eugenie Lyzenko wrote:

> Greg Shah wrote:
>
> > > Also there is a problem in registry where the base_name is actually not used in the native code. The char is loaded and then released, so part of the path is missing.
> >
> > Eugenie?
>
> On the time it was written it worked. I think the base_name is ignored because we always used HKEY_CURRENT_USER as the base. And it was enough. But it was 10 years ago and Windows can change internals.

I think it might be good enough still.

OpenEdge does some trickery, when using different base-key and I can't actually see (+ find by search) the value anywhere in the registry. I give up on changing something registry related.

#37 - 12/04/2023 12:01 PM - Galya B

EnvironmentDaemon is almost case-insensitive (check occurrences of .toLowerCase()). I guess it started with the registry in Win. But java Preferences is case-sensitive, so we can either keep the original names (envs / keys / sections), or make everything lowercase. The first sounds more proper, so we'll have this discrepancy in behavior between systems.

Basically USE "ENV" and USE "env" will be considered different environments in Linux and the same in Windows.

#38 - 12/04/2023 01:38 PM - Greg Shah

> But java Preferences is case-sensitive, so we can either keep the original names (envs / keys / sections), or make everything lowercase.

We need to be compatible and implementing case-sensitively seems very much incompatible. If we store the paths as lowercase how does that cause problems with the converted code?

#39 - 12/05/2023 02:17 AM - Galya B

Greg Shah wrote:

> > But java Preferences is case-sensitive, so we can either keep the original names (envs / keys / sections), or make everything lowercase.
>
> We need to be compatible and implementing case-sensitively seems very much incompatible. If we store the paths as lowercase how does that cause problems with the converted code?

Everything is part of the path: base-key, env, section and key (in java preferences). Currently, as I've said, it's partially case-insensitive implemented with:

```
** 006 CA  20180523 Fixed environment name, when used in maps: registries are case-insensitive,
**                  stanza INI are case-sensitive.
```

The 'partially' part is in some inconsistencies in the code: the map with envs has the original value of the env names as keys, but on unload they are retrieved in lower-case.

I'll try to fix it and make all (but .ini) case-insensitive then.

#40 - 12/05/2023 04:38 AM - Galya B

A note: The path in FallbackEnvironmentReader when using Preferences is composed from root user node ~/.java/.userPrefs (in Linux) + base node (com/goldencode/) + FWD username (bogus) + SSO storageId (if present) + base-key (if present) + env name (or __default) + section name (if present) + key name (or null).

#41 - 12/05/2023 07:37 AM - Galya B

- Status changed from WIP to Review

- % Done changed from 80 to 100

8074a based on trunk r14859 ready for review. Includes (in order of appearance):

- Theme options include an empty option and no default one is selected. On secondary login, if no option is selected, the theme is restored from the client storage. This fixes auto-login and forked session web flows where no login form is submitted.
- The logo image in the login form no longer draggable (styles.css change).
- storageId is no longer saved in the front-end, the client uses it in flat composite keys, when accessing data in localStorage.
- FWD user incorporated in ClientStorage composite keys.
- The code in UserPreferences simplified.
- com.goldencode.p2j.ui.client.gui.theme.Theme implementation classes with public static final fields NAME, corresponding to instance getName().
- ThemeManager: The fully qualified and simple class names and getName() names for the themes now referenced instead of hard-coded. This fixes some not matching values.
- SESSION:NEW-SESSION overloading methods for String, as requested in #7813-207.
- EnvironmentDaemon: Fix for some inconsistencies in case-sensitivity. Handling several specific cases of PUT-KEY-VALUE deleting values / sections, as described here.
- FallbackEnvironmentReader: Base-key added. Handling several specific cases of PUT-KEY-VALUE deleting values / sections, as described here. Handling Win-style segment separators in sections.

#42 - 12/06/2023 12:56 PM - Greg Shah

Eugenie/Hynek: Please review.

#43 - 12/06/2023 03:33 PM - Eugenie Lyzenko

In general I have no objections.

But I have the question and the clear answer is very important. How the custom themes will be supported with this release become in trunk? One customer has two specific themes to be loaded from external jar file. Will the application behave the same way as before? Do we have to modify directory.xml to preserve usage of customer specific themes?

Have you tested these changes with big customer application?

#44 - 12/07/2023 02:19 AM - Galya B

Eugenie Lyzenko wrote:

> In general I have no objections.
>
> But I have the question and the clear answer is very important. How the custom themes will be supported with this release become in trunk? One customer has two specific themes to be loaded from external jar file. Will the application behave the same way as before? Do we have to modify directory.xml to preserve usage of customer specific themes?

Everything is the same as before in this regard. The change is that selecting the theme in the front-end is not mandatory, but the default or the previously selected theme will be picked up by the client from the server / storage respectively. When it comes to resolving a custom theme or resolving themes from directory, that part of the code is not changed.

> Have you tested these changes with big customer application?

No, because there is no change in the core theme logic.

#45 - 12/22/2023 12:09 PM - Hynek Cihlar

Code review 8074a.

WebDriverHandler.getAvailableThemes should somehow handle a theme with empty name in the directory.

const userStorageJson = userStorage ? JSON.parse(userStorage) : {}; should handle the case the value is not a valid json.

retrieveSimpleValue and saveSimpleValue should be moved to the private section in the file.

In saveByChunks when JsonProcessingException is raised and one chunk fails, should the whole operation back out?

#46 - 01/01/2024 03:57 AM - Galya B

Hynek Cihlar wrote:

> Code review 8074a.
>
> WebDriverHandler.getAvailableThemes should somehow handle a theme with empty name in the directory.

It gets added to the options in the dropdown as an empty option and it's up to the customer to fix it. Failing or removing it (hiding the problem with unexpected behavior) is not reasonable.

> const userStorageJson = userStorage ? JSON.parse(userStorage) : {}; should handle the case the value is not a valid json.

Please check which branch and revision you're reviewing. This branch introduces "flat composite keys" (ref #8074-41) and the storage value is no longer parsed as json in web_landing.html.

> retrieveSimpleValue and saveSimpleValue should be moved to the private section in the file.

Done in r14865.

> In saveByChunks when JsonProcessingException is raised and one chunk fails, should the whole operation back out?

I haven't introduced changes to the logic there, just unwrapped the method saveNextChunk, but now that you've raised the question, having an atomic operation is a good improvement. Changes in r14865.

#47 - 01/03/2024 11:25 AM - Hynek Cihlar

Galya B wrote:

> Hynek Cihlar wrote:
>
> > Code review 8074a.
> >
> > WebDriverHandler.getAvailableThemes should somehow handle a theme with empty name in the directory.
>
> It gets added to the options in the dropdown as an empty option and it's up to the customer to fix it. Failing or removing it (hiding the problem with unexpected behavior) is not reasonable.

My point was that if you use a specific value (empty string) as a placeholder for a predefined behavior (default theme), this special value should be properly handled when loading values from the directory. For example by ignoring it.

#48 - 01/04/2024 02:09 AM - Galya B

Hynek Cihlar wrote:

> Galya B wrote:
>
> > Hynek Cihlar wrote:
> >
> > > Code review 8074a.
> > >
> > > WebDriverHandler.getAvailableThemes should somehow handle a theme with empty name in the directory.
> >
> > It gets added to the options in the dropdown as an empty option and it's up to the customer to fix it. Failing or removing it (hiding the problem with unexpected behavior) is not reasonable.
>
> My point was that if you use a specific value (empty string) as a placeholder for a predefined behavior (default theme), this special value should be properly handled when loading values from the directory. For example by ignoring it.

If an empty value is entered as an option for themes, this is an issue with configurations. Hiding it means it will never be fixed.

#49 - 01/04/2024 02:20 AM - Galya B

r14866 Theme without value in directory to use class name in the dropdown

#50 - 01/04/2024 03:45 AM - Hynek Cihlar

- Status changed from Review to Internal Test

Code review 8074a revisions 14865 and 14866. The changes are good, I have no more concerns. Do you plan any more testing?

#51 - 01/04/2024 07:18 AM - Galya B

It's been tested before the review and now I've tested the newly introduced changes - UserPreferences needed some minor tweaks to catch more types of exceptions, up in r14867. I think it's good to go.

#52 - 01/04/2024 07:52 AM - Hynek Cihlar

- Status changed from Internal Test to Merge Pending

Galya B wrote:

> It's been tested before the review and now I've tested the newly introduced changes - UserPreferences needed some minor tweaks to catch more types of exceptions, up in r14867. I think it's good to go.

Code review. The changes look good.

#53 - 01/04/2024 08:30 AM - Greg Shah

It can be merged to trunk now.

#54 - 01/04/2024 08:42 AM - Galya B

- Status changed from Merge Pending to Test

8074a was merged to trunk as rev. 14911 and archived.

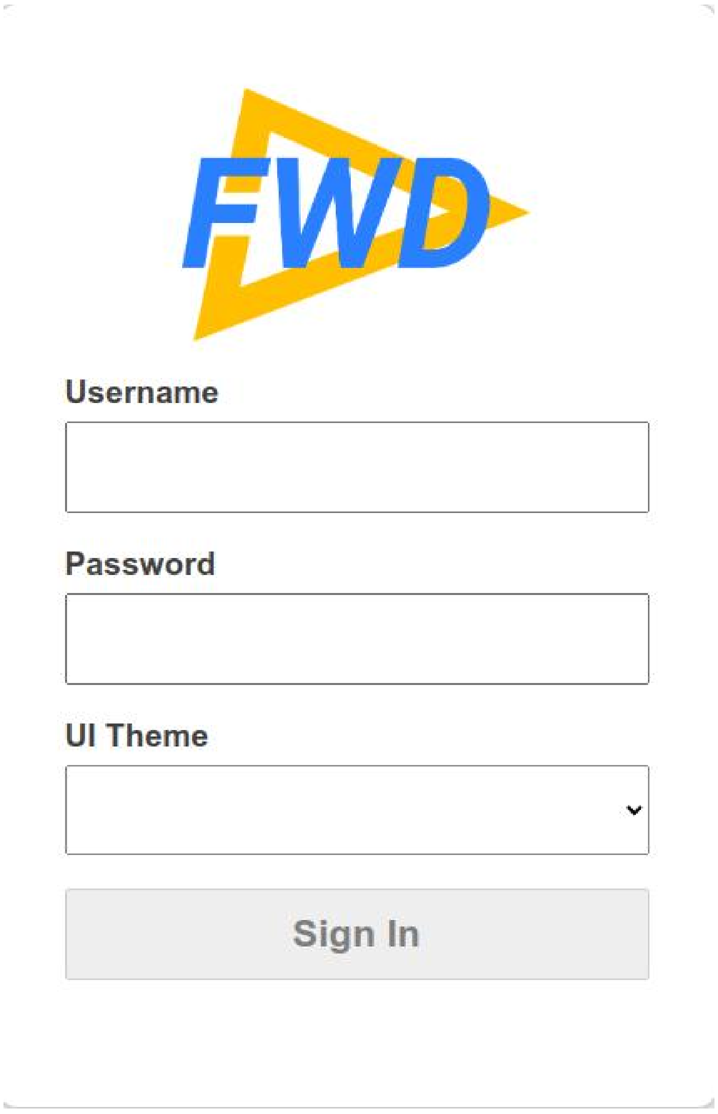
#55 - 01/04/2024 08:46 AM - Greg Shah

- Status changed from Test to Closed

#56 - 01/10/2024 08:58 AM - Roger Borrello

- File 7143_ui_theme.png added

I just wanted to clarify my post #7143-639 where I mentioned the UI Theme dropdown is appearing when it wasn't before, and make sure the applications behavior will be the same without any changes to the directory.



Prior to the merge of this branch, when the directory.xml had:

```
    <node class="container" name="server">
      <node class="container" name="default">
        <node class="container" name="available-themes">
          <node class="entry" name="theme1">
            <node-attribute name="key" value="com.goldencode.p2j.ui.client.gui.theme.Windows10Theme"/>
            <node-attribute name="value" value="Windows 10"/>
          </node>
        </node>
```

it would have a dropdown with only **Windows 10** available for selection. I removed it so that it wouldn't be available to the customer, since that was the only theme they wanted to deploy, and I thought it is the default. Is that the case now? That it is the default, and not necessary to be specified? After the merge of this branch, the dropdown came back, without any choices in it, and I just wanted to make sure there won't be any theme changes from Windows 10.

#57 - 01/10/2024 09:00 AM - Galya B

Roger Borrello wrote:

> it would have a dropdown with only **Windows 10** available for selection. I removed it so that it wouldn't be available to the customer, since that was the only theme they wanted to deploy, and I thought it is the default. Is that the case now? That it is the default, and not necessary to be specified? After the merge of this branch, the dropdown came back, without any choices in it, and I just wanted to make sure there won't be any theme changes from Windows 10.

Yes, it still works properly with the default Windows 10. It's just a UI issue of showing an empty option. Even when chosen, it still fallbacks to the default one in directory. Actually this is exactly its purpose: to allow fallback to the last selected theme, the one marked as preferred in directory or the default Windows 10 in this order.

#58 - 01/10/2024 09:03 AM - Roger Borrello

Galya B wrote:

> Yes, it still works properly with the default Windows 10. It's just a UI issue of showing an empty option. Even when chosen, it still fallbacks to the default one in directory. Actually this is exactly its purpose.

Thanks... Just for my own curiosity, where is the default theme specified? I searched for "theme" in directory.xml and didn't find anything. Is it in the code?

#59 - 01/10/2024 09:05 AM - Roger Borrello

Nevermind... easy find.

```
   public static Theme getCurrentTheme()
   {
      if (crtTheme == null)
      {
         crtTheme = new Windows10Theme();
      }

      return crtTheme;
   }
```

#60 - 01/10/2024 09:06 AM - Galya B

Roger Borrello wrote:

> Galya B wrote:
>
> > Yes, it still works properly with the default Windows 10. It's just a UI issue of showing an empty option. Even when chosen, it still fallbacks to the default one in directory. Actually this is exactly its purpose.
>
> Thanks... Just for my own curiosity, where is the default theme specified? I searched for "theme" in directory.xml and didn't find anything. Is it in the code?

There hasn't been a change in this regard. If a default is defined in directory by having theme/name that's used, otherwise the default Windows 10 is a hard-coded value, as you've found out.

#61 - 01/10/2024 11:42 AM - Greg Shah

The theme drop down was originally implemented to make it easy for GCD developers to test different themes. It was never intended to be an end-user feature. I've mentioned before that we have other settings/options which we might also want to allow a user to set (e.g. bootstrap cfg values like the startup program) but these are also special cases for testing and would not be recommended to be made available for end users.

Themes are safe to expose but we don't want to do so by default. I especially don't want an empty drop down to be shown to an end user by default. Let's change the approach to only show that drop down when explicitly enabled. Otherwise we just use the defined theme in the directory or we default to Windows 10.

#62 - 01/11/2024 01:44 AM - Galya B

Greg Shah wrote:

> The theme drop down was originally implemented to make it easy for GCD developers to test different themes. It was never intended to be an end-user feature. I've mentioned before that we have other settings/options which we might also want to allow a user to set (e.g. bootstrap cfg values like the startup program) but these are also special cases for testing and would not be recommended to be made available for end users.
>
> Themes are safe to expose but we don't want to do so by default. I especially don't want an empty drop down to be shown to an end user by default. Let's change the approach to only show that drop down when explicitly enabled. Otherwise we just use the defined theme in the directory or we default to Windows 10.

The empty dropdown is a bug. I've said in #7143-640 that it should be fixed.

#63 - 01/11/2024 03:25 AM - Galya B

8074b r14916 adding empty option to UI theme selector only when one or more themes are configured in directory.

#64 - 01/11/2024 07:33 AM - Greg Shah

Why even have the dropdown when there is only 1 choice? Shouldn't the dropdown only exist if there are more than 1 choice?

#65 - 01/11/2024 07:54 AM - Galya B

Greg Shah wrote:

> Why even have the dropdown when there is only 1 choice? Shouldn't the dropdown only exist if there are more than 1 choice?

Who can tell. You've added a theme to directory's node available-themes, but haven't selected it as the preferred one theme/name, so the server still returns Windows10Theme from ServerExports.getUiTheme. The only way to choose it, is through the web ui. That's how it's been working always. Do you want to change it somehow?

#66 - 01/11/2024 01:31 PM - Greg Shah

> That's how it's been working always. Do you want to change it somehow?

I understand this is not new. It is just bad UI design to have a dropdown with only 1 entry. If it is low risk to eliminate the dropdown in that case, that would be better.

#67 - 01/12/2024 02:40 AM - Galya B

OK, then if only one theme is added to available-themes, it will be set as a default even if it's not defined in theme/name.

What about multiple themes listed in available-themes, but none in theme/name? Do we need to set the first one by default or leave it fallback to win 10, when the user doesn't select any in the drop-down? Since there is an empty option (to be able to restore the preferred theme from client storage after login), this is a valid question.

#68 - 01/13/2024 09:06 AM - Greg Shah

> OK, then if only one theme is added to available-themes, it will be set as a default even if it's not defined in theme/name.

I'd prefer not to show the field at all in this case.

> What about multiple themes listed in available-themes, but none in theme/name? Do we need to set the first one by default or leave it fallback to win 10, when the user doesn't select any in the drop-down?

Set it to win 10 if it is in the list, otherwise the first non-empty value should be set.

#69 - 01/15/2024 06:28 AM - Galya B

r14917: Change to default theme resolvement. Login screen theme select shows up only with 2 or more configured directory themes.

Ready for final review.

#70 - 01/15/2024 07:13 AM - Greg Shah

Hynek/Sergey: Please review.

#71 - 01/15/2024 12:44 PM - Hynek Cihlar

Code review 8074b revision 14917.

The only concern I have is the init of themes in the static ctor of LogicalTerminal. It is generally not under your control how and when will be the static constructor called. A more explicit approach would be better - a dedicated static method called during server startup.

#72 - 01/16/2024 02:45 AM - Galya B

Hynek Cihlar wrote:

> Code review 8074b revision 14917.
>
> The only concern I have is the init of themes in the static ctor of LogicalTerminal. It is generally not under your control how and when will be the static constructor called. A more explicit approach would be better - a dedicated static method called during server startup.

Actually it is under your control, the first time the class is accessed the static initializer will be executed. This will be at latest when the static initializer of WebDriverHandler asks for LogicalTerminal.getThemes(), and it on its turn is executed when ServerDriver.start -> StandardServer.bootstrap -> StandardServer.registerDefaultServices, where VirtualDesktopWebHandler or EmbeddedWebHandler are accessed.

Anyways, we don't have to do the mental gymnastics. The moment the theme related methods are called, the themes will have already been loaded in the static initializer, because that's how a Java class is loaded in the JVM. If you need an official source, here it is https://docs.oracle.com/javase/specs/jls/se8/html/jls-12.html#jls-12.4 you can see:

> Initialization of a class consists of executing its static initializers and the initializers for static fields (class variables) declared in the class.

> A class or interface type T will be initialized immediately before the first occurrence of any one of the following:
> T is a class and an instance of T is created.
> A static method declared by T is invoked.

#73 - 01/16/2024 02:58 AM - Hynek Cihlar

Galya B wrote:

> Hynek Cihlar wrote:
>
> > Code review 8074b revision 14917.
> >
> > The only concern I have is the init of themes in the static ctor of LogicalTerminal. It is generally not under your control how and when will be the static constructor called. A more explicit approach would be better - a dedicated static method called during server startup.
>
> Actually it is under your control, the first time the class is accessed the static initializer will be executed. This will be at latest when the static initializer of WebDriverHandler asks for LogicalTerminal.getThemes(), and it on its turn is executed when ServerDriver.start -> StandardServer.bootstrap -> StandardServer.registerDefaultServices, where VirtualDesktopWebHandler or EmbeddedWebHandler are accessed.

You are missing the point. You don't access the class intentionally to initialize the themes. At this point it is just a coincidence the code works, the class is accessed only after the directory service is ready. This may however break in the future when LogicalTerminal is referenced before the directory service is initialized.

#74 - 01/16/2024 03:28 AM - Galya B

Hynek Cihlar wrote:

> Galya B wrote:
>
> > Hynek Cihlar wrote:
> >
> > > Code review 8074b revision 14917.
> > >
> > > The only concern I have is the init of themes in the static ctor of LogicalTerminal. It is generally not under your control how and when will be the static constructor called. A more explicit approach would be better - a dedicated static method called during server startup.
> >
> > Actually it is under your control, the first time the class is accessed the static initializer will be executed. This will be at latest when the static initializer of WebDriverHandler asks for LogicalTerminal.getThemes(), and it on its turn is executed when ServerDriver.start -> StandardServer.bootstrap -> StandardServer.registerDefaultServices, where VirtualDesktopWebHandler or EmbeddedWebHandler are accessed.
>
> You are missing the point. You don't access the class intentionally to initialize the themes. At this point it is just a coincidence the code works, the class is accessed only after the directory service is ready. This may however break in the future when LogicalTerminal is referenced before the directory service is initialized.

DirectoryService ds = DirectoryService.createInstance(config); is one of the first statements in StandardServer.bootstrap and that is on purpose, but not a coincidence. LogicalTerminal is not supposed to be used before that, because it is:

```
/**
 * This class represents the state of the logical terminal associated with the
 * current user session. There is one instance of this class per session
 * context.
 */
```

If in the future there is a misuse of LogicalTerminal it will have to be fixed.

#75 - 01/16/2024 03:45 AM - Hynek Cihlar

Galya B wrote:

> If in the future there is a misuse of LogicalTerminal it will have to be fixed.

You cannot assume today what a misuse will be in the future. Why not to future prove it today?

#76 - 01/16/2024 03:48 AM - Galya B

Hynek Cihlar wrote:

> Galya B wrote:
>
> > If in the future there is a misuse of LogicalTerminal it will have to be fixed.
>
> You cannot assume today what a misuse will be in the future. Why not to future prove it today?

You are missing the point. It's not expected to do DirectoryService.createInstance(config) before each DirectoryService.getInstance() in the whole code, that's how FWD is written.

Your argument is that we should consider cases where someone inserts line 1 before the initialization:

```
a = 2;
int a = 1;
```

But this is not how programming works.

#77 - 01/16/2024 04:01 AM - Hynek Cihlar

Galya B wrote:

> Hynek Cihlar wrote:
>
> > Galya B wrote:
> >
> > > If in the future there is a misuse of LogicalTerminal it will have to be fixed.
> >
> > You cannot assume today what a misuse will be in the future. Why not to future prove it today?
>
> You are missing the point. It's not expected to do DirectoryService.createInstance(config) before each DirectoryService.getInstance() in the whole code, that's how FWD is written.
>
> Your argument is that we should consider cases where someone inserts line 1 before the initialization:
> [...]
> But this is not how programming works.

No, my argument is that someone references LogicalTerminal even before the conditions are satisfied for the directory service to become ready. It probably can't happen today. But can you assume this for the future? You are cementing the requirement to have the directory service ready just to reference the Java class. So when somebody will want to add a static field to LogicalTerminal and use it outside of the security context he will hit this mindless requirement.

#78 - 01/16/2024 04:23 AM - Galya B

Hynek Cihlar wrote:

> No, my argument is that someone references LogicalTerminal even before the conditions are satisfied for the directory service to become ready. It probably can't happen today. But can you assume this for the future? You are cementing the requirement to have the directory service ready just to reference the Java class. So when somebody will want to add a static field to LogicalTerminal and use it outside of the security context he will hit this mindless requirement.

So you want to refactor FWD to not use DirectoryService.getInstance() anywhere, but instead DirectoryService.createInstance(config)? Who can confirm the mindless requirement is met everywhere for every static method, if someone mindlessly uses it before the conditions are satisfied?

#79 - 01/16/2024 04:44 AM - Hynek Cihlar

Galya B wrote:

> So you want to refactor FWD to not use DirectoryService.getInstance() anywhere, but instead DirectoryService.createInstance(config)? Who can confirm the mindless requirement is met everywhere for every static method, if someone mindlessly uses it before the conditions are satisfied?

No, I want explicit, readable and obvious dependencies. Dependencies that will not break with future changes. That is a dedicated method for themes initialization called from a well selected point during server startup.

#80 - 01/16/2024 04:45 AM - Galya B

Hynek Cihlar wrote:

> Galya B wrote:
>
> > So you want to refactor FWD to not use DirectoryService.getInstance() anywhere, but instead DirectoryService.createInstance(config)? Who can confirm the mindless requirement is met everywhere for every static method, if someone mindlessly uses it before the conditions are satisfied?
>
> No, I want explicit, readable and obvious dependencies. Dependencies that will not break with future changes. That is a dedicated method for themes initialization called from a well selected point during server startup.

See, you've touched on an important point, it's just 10 years late to the project. I'm happy my code stimulated your brain power and I hope my comments helped you fully understand the context. So, it's not that I oppose it, it's just not correct to be tunnel visioned when solving an issue. If the issue is an actual one, it has to be solved globally. That is the prevalent use of static methods in FWD that undermines the reasoning about dependencies. If you want to solve this issue, then open a new ticket and I'll be glad to assist.

#81 - 01/16/2024 05:11 AM - Hynek Cihlar

Galya B wrote:

> > No, I want explicit, readable and obvious dependencies. Dependencies that will not break with future changes. That is a dedicated method for themes initialization called from a well selected point during server startup.
>
> See, you've touched on an important point, it's just 10 years late to the project. I'm happy my code stimulated your brain power and I hope my comments helped you fully understand the context. So, it's not that I oppose it, it's just not correct to be tunnel visioned when solving an issue. If the issue is an actual one, it has to be solved globally. That is the prevalent use of static methods in FWD that undermines the reasoning about dependencies. If you want to solve this issue, then open a new ticket and I'll be glad to assist.

I see. You are deliberately trying to deliver a poor quality code to point on the bigger issue in FWD. :-)

#82 - 01/16/2024 05:11 AM - Galya B

Hynek Cihlar wrote:

> Galya B wrote:
>
> > > No, I want explicit, readable and obvious dependencies. Dependencies that will not break with future changes. That is a dedicated method for themes initialization called from a well selected point during server startup.
> >
> > See, you've touched on an important point, it's just 10 years late to the project. I'm happy my code stimulated your brain power and I hope my comments helped you fully understand the context. So, it's not that I oppose it, it's just not correct to be tunnel visioned when solving an issue. If the issue is an actual one, it has to be solved globally. That is the prevalent use of static methods in FWD that undermines the reasoning about dependencies. If you want to solve this issue, then open a new ticket and I'll be glad to assist.
>
> I see. You are deliberately trying to deliver a poor quality code to point on the bigger issue in FWD. :-)

No, to teach you the order of initialization.

#83 - 01/20/2024 03:40 PM - Eugenie Lyzenko

Galya,

We have serious regression with this change. The theme defined in:

```
...
        <node class="container" name="theme">
          <node class="string" name="name">
            <node-attribute name="value" value="Customer_specific_theme"/>
          </node>
        </node>
...
```

Instead the Windows10Theme is always loaded. This is a deep trouble because the customer code has specific paintings.

This caused the class cast exception in big customer application and should be fixed. Or documented how to avoid this by configuring directory.xml.

#84 - 01/20/2024 05:02 PM - Eugenie Lyzenko

Galya,

The problem is the preferred theme should be loaded from:

```
...
        <node class="container" name="theme">
          <node class="string" name="name">
            <node-attribute name="value" value="Customer_specific_theme"/>
          </node>
        </node>
...
```

Currently the specific theme can be declared in:

```
          <node class="string" name="cfgOverrides">
            <node-attribute name="value" value="... client:driver:theme=Customer_specific_theme"/>
          </node>
```

If it is not defined in cfgOverrides the Window10Theme is used. But it is not convenient. Moreover some customer code will abend if the required theme is not available. We used theme/name entry in directory.xml I think this need to be fixed to restore the functionality used before your change.

#85 - 01/22/2024 02:28 AM - Galya B

Eugenie Lyzenko wrote:

> Galya,
>
> The problem is the preferred theme should be loaded from:
> [...]
>
> Currently the specific theme can be declared in:
> [...]
>
> If it is not defined in cfgOverrides the Window10Theme is used. But it is not convenient. Moreover some customer code will abend if the required theme is not available. We used theme/name entry in directory.xml I think this need to be fixed to restore the functionality used before your change.

I'm waiting for approval to merge 8074b with the fix.

#86 - 01/22/2024 07:32 AM - Greg Shah

Eugenie: Please review the fix.

#87 - 01/22/2024 08:45 AM - Eugenie Lyzenko

Greg Shah wrote:

> Eugenie: Please review the fix.

I have no objections for changes in rev 14917.

But I think it is probably not enough to cover all cases. The ThemeManager.getCurrentTheme() can still return Windows10Theme when it is expected to use another one in customer specific code.

Galya,

Have you tested this branch with big customer application?

#88 - 01/22/2024 08:54 AM - Galya B

Eugenie Lyzenko wrote:

> Greg Shah wrote:
>
> > Eugenie: Please review the fix.
>
> I have no objections for changes in rev 14917.
>
> But I think it is probably not enough to cover all cases. The ThemeManager.getCurrentTheme() can still return Windows10Theme when it is expected to use another one in customer specific code.

ThemeManager.getCurrentTheme() is client-side method available after initialization ThemeManager.init (where ThemeManager.setTheme(String name) is called). It will never return Windows10Theme if a theme is configured in directory. ThemeManager.init first restores the theme for the current user from the client storage (where last preference has been stored). If the user hasn't selected any theme, ServerExports.getUiTheme is called, that is LogicalTerminal.getUiTheme where theme/name is returned if configured; if theme/name is not configured, the first theme from available-themes is returned or Windows10Theme when no theme is configured.

> Galya,
>
> Have you tested this branch with big customer application?

The logic is customer independent and can be easily tested with any web client. It has been well tested.

#89 - 01/22/2024 08:56 AM - Greg Shah

If the customer has configured a specific theme in the directory, that should always take precedence. An end user should not be able to override that configuration.

#90 - 01/22/2024 08:58 AM - Galya B

Greg Shah wrote:

> If the customer has configured a specific theme in the directory, that should always take precedence. An end user should not be able to override that configuration.

This is yet another new requirement that never existed in the original code. So when theme/name is available in directory, the user should not be presented the UI Theme select on the login screen?

#91 - 01/22/2024 09:10 AM - Greg Shah

> This is yet another new requirement that never existed in the original code.

OK, then that was a bug.

> So when theme/name is available in directory, the user should not be presented the UI Theme select on the login screen?

Correct.

#92 - 01/22/2024 09:26 AM - Galya B

I will need a day or so to completely implement the new requirement. It involves two server-side login endpoints and client-side restoring the preferred theme. Since theme/name overrides all other theme configs, do we need something fancy like first checking if it's a valid theme and parse the other themes based on it or just throw error and never parse available-themes when theme/name is configured?

It should not be an issue to delay the review, since the proper configuration in directory currently does the work.

#93 - 01/22/2024 09:53 AM - Eugenie Lyzenko

Eugenie Lyzenko wrote:

> Greg Shah wrote:
>
> > Eugenie: Please review the fix.
>
> I have no objections for changes in rev 14917.
>
> But I think it is probably not enough to cover all cases. The ThemeManager.getCurrentTheme() can still return Windows10Theme when it is

> expected to use another one in customer specific code.
>
> Galya,
>
> Have you tested this branch with big customer application?

I have tested rev 14917. It works fine with customer code.

So I have no more objections.

#94 - 01/22/2024 09:54 AM - Galya B

Greg, I'll be waiting for clarification to the question in #8074-92.

#95 - 01/22/2024 10:05 AM - Greg Shah

> It should not be an issue to delay the review, since the proper configuration in directory currently does the work.

Do I understand correctly, that without directory changes the large customer GUI application is broken with the current trunk?

#96 - 01/22/2024 10:14 AM - Galya B

Greg Shah wrote:

> > It should not be an issue to delay the review, since the proper configuration in directory currently does the work.
>
> Do I understand correctly, that without directory changes the large customer GUI application is broken with the current trunk?

OK, let me clarify: Previously the UI theme select appeared with only one option available that was always selected. The theme selection was driven by the UI. Forking sessions break this, so #8074 fixes themes with forked sessions, but deselects the theme in the UI. It can still be configured in directory. I'm trying to explain that we're on the road to fixing all issues in an improved manner in comparison to the simplistic original logic.

So, what's the answer of my question about the requirements?

#97 - 01/22/2024 10:16 AM - Eugenie Lyzenko

Greg Shah wrote:

> > It should not be an issue to delay the review, since the proper configuration in directory currently does the work.
>
> Do I understand correctly, that without directory changes the large customer GUI application is broken with the current trunk?

I had this issue on last Friday is some configuration. But not for today. Seems like it works fine with trunk. So I think we can drop this note for now.

#98 - 01/22/2024 10:40 AM - Galya B

Also, let me note that a customer using a custom login page is responsible for sending the selected theme the same way it is sent by the FWD page or configure it in directory as theme/name. They bypass both options and suddenly the old code doesn't work... naturally... and we have to invent a third one.

#99 - 01/22/2024 07:17 PM - Eugenie Lyzenko

Eugenie Lyzenko wrote:

> Greg Shah wrote:
>
> > > It should not be an issue to delay the review, since the proper configuration in directory currently does the work.
> >
> > Do I understand correctly, that without directory changes the large customer GUI application is broken with the current trunk?
>
> I had this issue on last Friday is some configuration. But not for today. Seems like it works fine with trunk. So I think we can drop this note for now.

Some weird addition.

If web client application explicitly loads some theme (Windows10Theme for example) the next Web client for **another** server application that attempts to load custom theme by theme/name option will load Window10Theme ignoring theme/name.

It is difficult to believe in but it is not a joke. True for both current trunk and 8074b branch.

#100 - 01/23/2024 08:30 AM - Galya B

Eugenie Lyzenko wrote:

> Some weird addition.
>
> If web client application explicitly loads some theme (Windows10Theme for example) the next Web client for **another** server application that attempts to load custom theme by theme/name option will load Window10Theme ignoring theme/name.
>
> It is difficult to believe in but it is not a joke. True for both current trunk and 8074b branch.

Thanks for reporting. It seems we have a few issues on trunk that need discussion:

1. theme/name doesn't override anything (including explicit user preference), except in the case where no option has been selected from the UI. That is how it has been working for a decade, but it's to be changed with that new requirement in #8074-91 being implemented, once its full behavior is confirmed #8074-92.
2. The selected theme is currently stored in the client storage, but the client storage is not app specific. The client storage includes Win registry, java user Preferences and localStorage. Customers with SSO in place have the control to choose unique storage id for each of their users and this has been clearly stated in the SSO documentation. If no SSO is in place, the FWD user name is the unique id for the storage. If two applications use the same FWD user name and have no SSO implemented, or don't provide storage id, currently there is no mechanism in place to differentiate between their configurations and this is something I've raised as a question / issue in the past. **Please check #4517-13!**

The theme selection got complicated by the new app flows with the custom login page, auto-login and forked sessions. Let me get into the details:
Previously the theme was chosen every time the user was asked to enter OS credentials. A select with one or more options was presented in the UI having theme/name as default and the selected theme was sent to the client. theme/name didn't override all other options and the user preference wasn't stored. This screen is now not present. It's been merged with in-app user login, which can be fully overridden by a custom page for some customers. If the customer doesn't present the select to their end-users, then the default theme is selected server-side, that is theme/name or win 10. This has been changed in 8074b (waiting for merge) to auto-select the first available-themes if theme/name is not present.
The next caveat is auto-login and forked sessions: these flows skip the login screen altogether. That's why I introduced the change to save the theme selected by the user in the client storage. Otherwise their preference will be lost on consequent login and they will not even be presented the opportunity to select on option. I tried to make client storage user specific by introducing storage id in the result returned by the customer SsoAuthenticator implementation, but the customer can decide to not provide it. The limitations of the customer storage types are presented in #4517-13. I did some improvements on top of that (yet to find the task) where I've added the FWD user and SSO selected storage ID as well.
And the last change is the empty option in the select introduced by 8074a. On the login screen the user is anonymous, so we can't preselect their previous choice. It has to be restored from the client storage. Also preselecting each time any other default theme is not correct, because it overwrites their previous choice. That's why an empty option has been added as a first option in the UI theme select.

All along the way implementing these new features I gave clear indications how things work and I've asked for feedback and requirement analysis. If we've missed something, we should discuss it and implement it next. **Please take into consideration all above and advise.**

#101 - 01/23/2024 09:38 AM - Greg Shah

As mentioned previously, exposing the theme to the end user was never intended to be an important feature. It was implemented as a convenience for Golden Code testers. Customers with production installations have (so far) all wanted to remove this drop down and hard code the theme choice. At least one of our customers, even provides their own customized theme which provides a unique look and feel. When you do things like that, you don't want your end users to have control over the theme choice.

If you don't want your end users to have control over the theme choice, then you also don't want the theme to be read from a transient storage location like offline browser storage or even Java preferences. When you've configured a fixed theme in the directory, you expect it to always be used.

As mentioned previously, I can see a future in which we don't have the theme choice as a separate drop down, but instead we have some "advanced options" dialog that can be brought up conditionally at login. This might allow bootstrap config settings to be customized (like the startup program) but it opens up serious security issues. For this reason we haven't spent time on it.

Considering how much time and arguments this is causing, I'm wondering if we should just remove the stupid theme drop down. My sense is that this unnecessary feature has been causng increasingly complicated additional implementations, which are not needed or wanted.

1. If a customer has configured a theme that should always be used, then it should always be used. There should not be any theme drop down, nor any honoring of previous selections. The user should have no control over the theme. The login page should have no impact in this case because the theme has been pre-selected.

2. If the list of configured themes has been set in the directory but there is only one valid theme in the list, then that theme should be considered pre-selected and would be treated the same way as in item 1.

3. The only time we should show a theme dialog is if there is no pre-selected theme and there are more than one valid theme in the list of available themes. These should then be shown to the user in the theme drop down. We should NOT show an empty string by default. Here is what to select by default, in order of higher to lower precedence:

a. If there is a saved user preference AND that preference matches one of the available themes, then show that as selected by default. If there is a saved user preference and it does NOT match an available theme, then that preference is ignored.
b. If Windows 10 is in the list, it should be selected in the drop down by default.
c. If Windows 10 isn't in the list then the first theme alphabetically should be selected by default.

4. In the case of auto-login or forked sessions, the selection of the theme should still follow the above rules including items 1 and 2. The only difference is that in item 3, we don't just "select" in the drop down, but we just use the selection.

As far as I understand it, by these rules we can never get an invalid theme choice or no theme choice.

#102 - 01/23/2024 09:38 AM - Greg Shah

- % Done changed from 100 to 80

- Status changed from Closed to WIP

#103 - 01/23/2024 10:12 AM - Galya B

I believe we need to invent the dev flag in our configurations (it disables all features not meant for production in one go). Some companies have already invented it. Then customers won't be inclined to configure the dev features in their configuration. Obviously available-themes was the dev feature, customers rely on instead of configuring theme/name. In the light of lack of documentation on the topic and misleading code / configs any new developer would consider this feature essential (being placed on the login screen).

The translation of "always to be used" was lost in the first implementation:

```
   /**
    * Initialize the {@code ThemeManager} using negotiation with the server. If a theme is not
    * requested in command line (passed in by {@code preferredTheme} parameter), it will request
    * the configuration from server directory, but it will use that as a hint only. The class
    * configured in server directory might no contain a proper theme/class name. If no definition
    * is found in directory, or the string obtained does not represent a name or full class of a
    * theme accessible from client path, the theme remains uninitialized.
    * It will be lazily instantiated when accessed the first time.
    *
    * @param   server
    *          the {@link ServerExports} object for communication with server.
    * @param   preferredTheme
    *          the preferred theme from command-line.
    */
   public static boolean init(ServerExports server, String preferredTheme)
   {
      if (preferredTheme == null || preferredTheme.isEmpty())
      {
         // request a configuration from server
         preferredTheme = server.getUiTheme();
      }

      // we attempt to set this theme. If call fails, the default theme will automatically be set,
      // using lazy initialization, at first call to ThemeManager.getCurrentTheme()
      return setTheme(preferredTheme);
   }
```

The configuration theme/name also doesn't present the expected semantics. Even if you repeat 100 times now that it's "always to be used" it needs something more to be true.

What arguments is this causing? I don't see anyone arguing about themes. The themes raised only the requirements analysis discussion.

The now presented requirements would mean that theme is restored from client storage only with auto-login and forked session, only if valid and theme/name is not configured. Then do we fix the multiple apps - same fwd user - one theme issue that will be present only in this particular case, that is expected to appear only for FWD devs?

#104 - 01/23/2024 01:02 PM - Greg Shah

> Even if you repeat 100 times now that it's "always to be used" it needs something more to be true.

You asked for requirements. I hope I've provided them.

> What arguments is this causing? I don't see anyone arguing about themes. The themes raised only the requirements analysis discussion.

I'm seeing effort spent implementing extra features for something that is only a mildly convenient feature. I'm seeing changes in behavior visible to the end user for a feature that is not currently useful in production environments. To the extent that those changes cause the login process to be more effort than it was previously, those changes might reasonably be reported as regressions.

We all know you didn't write all of this code. It doesn't really matter how we got the behavior that exists now. I'm confident that you can implement the requirements.

> The now presented requirements would mean that theme is restored from client storage only with auto-login and forked session, only if valid and theme/name is not configured.

In addition to a valid theme/name, the directory might only hold a single valid theme in the list of themes. This implicitly selects that theme the same way that explicitly setting theme/name would do. Because it makes no sense to implement a UI feature for selection when there is only 1 item that could ever be selected.

No, it isn't just for auto-login or forked sessions. It would be used in the case that there are multiple possible selections and we have a dropdown to display, then we can use that saved value to set the default choice for the user, so long as it is a match to a valid theme name in the drop down. We definitely don't want to set the default to empty string.

> Then do we fix the multiple apps - same fwd user - one theme issue that will be present only in this particular case, that is expected to appear only for FWD devs?

No, we don't care about this.

#105 - 01/23/2024 02:04 PM - Eugenie Lyzenko

Let me share my generic thoughts here. Just my vision.

Every application has it's commonly used life cycle. And we can divide the people using it in two(excluding developers) groups: ones who deploy the application and ones who will finally use it. And then think about requirements for both. These groups have different demands. The deploy people decides how the application should work and should look. And the process of configuration must be simple and clear. Does the final user care about UI theme to apply? I really doubt. How many times in a day you think about "what UI theme I will use for today?"

If we want every application to be configured via directory.xml then every application must be exactly configured by respective directory file. I do not think it is a good idea to store some global settings inside FWD related registry to share between multiple FWD applications. Moreover one system can be used by two or more peoples with different theme preferences...

#106 - 01/24/2024 01:47 AM - Galya B

Eugenie, I wasn't the person implementing UI themes and noone nowhere has documented how insignificant the feature is. I was trying to keep it working with all the login changes, so I keep getting surprised by all the negative feedback I get.

If I haven't made changes, forked session and auto-logged users would have never even had the chance of seeing anything but Windows 10 theme based on the original code.

So please all of you, bear that in mind.

Greg, I can't proceed to implementation until you get on the same page with Eugenie.

#107 - 01/24/2024 09:02 AM - Greg Shah

> If I haven't made changes, forked session and auto-logged users would have never even had the chance of seeing anything but Windows 10 theme based on the original code.
>
> So please all of you, bear that in mind.

We know. This isn't about blame. The point here is to ensure we get the right implementation.

> Greg, I can't proceed to implementation until you get on the same page with Eugenie.

I don't see any real conflict between my requirements and Eugenie's ideas.

Eugenie: If you see something in the requirements from #8074-101 that is not workable or which needs changing, please say so.

The only thing I can think of that is not explicitly specified there is how to treat the preferred theme returned by any SSO custom authenticator. I think we can honor that (if it is a valid theme) as if it was the configured "fixed" theme for that user.

#108 - 01/24/2024 09:22 AM - Galya B

I want to get closure, but I also want to make sure every false or troublesome statement has been addressed.

Eugenie says:

> I do not think it is a good idea to store some global settings inside FWD related registry to share between multiple FWD applications.

1. Well, this is how client storage works more or less (especially Win registry) and this behavior was described in a table and reviewed by Greg in #4517-22.
2. Nothing global has been stored in client storage, but user preferences. The issue I've described tens of times is web users without SSO or with poorly implemented SSO are treated by FWD as one user.

Eugenie says:

> Moreover one system can be used by two or more peoples with different theme preferences...

Could it? Greg says this could only happen to devs, because prod systems use only one theme that will be set by default from the server and nothing will be stored in user preferences. If not so, Greg, we need to address Eugenie's concerns of sharing storage between apps.

> The only thing I can think of that is not explicitly specified there is how to treat the preferred theme returned by any SSO custom authenticator.

The SSO authenticator doesn't return theme, but storage id (if the customer was so kind to implement it).

#109 - 01/24/2024 09:46 AM - Greg Shah

> 1. Well, this is how client storage works more or less (especially Win registry) and this behavior was described in a table and reviewed by Greg in #4517-22.
> 2. Nothing global has been stored in client storage, but user preferences. The issue I've described tens of times is web users without SSO or with poorly implemented SSO are treated by FWD as one user.

We are not trying to solve this problem here. This is a general problem caused by the original 4GL application design and as such, the solution in each case may be different.

Where we make the problem worse is that we add a new storage of the theme which the original application did not have. I largely want that additional storage to be ignored and only honored if we have more than one valid theme and the stored value matches one of them. In that case, it just doesn't matter if that preference was for some other app or user, it is a valid choice and we will select it.

> > Moreover one system can be used by two or more peoples with different theme preferences...
>
> Could it? Greg says this could only happen to devs, because prod systems use only one theme that will be set by default from the server and nothing will be stored in user preferences. If not so, Greg, we need to address Eugenie's concerns of sharing storage between apps.

We are not trying to solve the general problem here. For this specific case, I don't see a problem.

> > The only thing I can think of that is not explicitly specified there is how to treat the preferred theme returned by any SSO custom authenticator.
>
> The SSO authenticator doesn't return theme, but storage id (if the customer was so kind to implement it).

OK, then the SSO authenticator doesn't change the logic. It just may make a preference visible, which will be ignored except in very specific use cases.

#110 - 01/24/2024 09:53 AM - Galya B

Eugenie, I'll be waiting for your confirmation that you don't find an issue with the requirements in #8074-101, if so.

#111 - 01/24/2024 09:57 AM - Galya B

Greg Shah wrote:

> The only thing I can think of that is not explicitly specified there is how to treat the preferred theme returned by any SSO custom authenticator.

> > The SSO authenticator doesn't return theme, but storage id (if the customer was so kind to implement it).
>
> OK, then the SSO authenticator doesn't change the logic. It just may make a preference visible, which will be ignored except in very specific use cases.

The storage id from the SSO authenticator will make a preference invisible, because it is a piece <u>added</u> to the already existing fwd username in the key to make it even more unique.

#112 - 01/24/2024 10:25 AM - Greg Shah

Or because it is added, a stored preference will become visible. My point is that it does not matter. We don't need to change the requirements for the SSO authenticator.

#113 - 01/24/2024 10:55 AM - Galya B

Greg Shah wrote:

> Or because it is added, a stored preference will become visible. My point is that it does not matter. We don't need to change the requirements for the SSO authenticator.

The stored preferences are always visible, with or without sso. It will be less shared if a unique storage id is provided.

#114 - 01/24/2024 11:35 AM - Eugenie Lyzenko

Greg Shah wrote:

> > If I haven't made changes, forked session and auto-logged users would have never even had the chance of seeing anything but Windows 10 theme based on the original code.
> >
> > So please all of you, bear that in mind.
>
> We know. This isn't about blame. The point here is to ensure we get the right implementation.
>
> > Greg, I can't proceed to implementation until you get on the same page with Eugenie.

> I don't see any real conflict between my requirements and Eugenie's ideas.
>
> Eugenie: If you see something in the requirements from #8074-101 that is not workable or which needs changing, please say so.

I re-read this and honestly did not find something conflicting with what I noted. It is a good plan I agree.

Galya,

The customer should provide implementation requirements for specific application. My vision is a just advice, I'm not a user of the application.

#115 - 01/25/2024 08:21 AM - Galya B

Greg Shah wrote:

> Here is what to select by default, in order of higher to lower precedence:
>
> a. If there is a saved user preference AND that preference matches one of the available themes, then show that as selected by default. If there is a saved user preference and it does NOT match an available theme, then that preference is ignored.

I've missed to notice this earlier. I've said it a few times already, we can't restore the previously selected theme on the login screen, because the user is anonymous and the client storage needs to be accessed client-side from the FWD client after the login (client spawn).

#116 - 01/25/2024 08:24 AM - Galya B

The empty option was there exactly to provide a user in a hurry to have their previously selected theme restored after the login. Now, we argued that this user is a FWD dev and they need to select the theme from the drop-down every time to keep their preference.

#117 - 01/25/2024 08:26 AM - Galya B

That's why I said:

> The now presented requirements would mean that theme is restored from client storage only with auto-login and forked session, only if valid and theme/name is not configured.

#118 - 01/25/2024 08:57 AM - Galya B

Greg, are you on board that the previously selected theme will not be restored on the login screen?

#119 - 01/25/2024 10:59 AM - Greg Shah

> Greg, are you on board that the previously selected theme will not be restored on the login screen?

Yes.

> The empty option was there exactly to provide a user in a hurry to have their previously selected theme restored after the login. Now, we argued that this user is a FWD dev and they need to select the theme from the drop-down every time to keep their preference.

The empty option is harmful from a user perspective. It seems to require the user to make a choice, when we don't really want the standard user to make a choice. This is why I want the empty option to be removed.

#120 - 01/26/2024 01:38 AM - Galya B

Greg Shah wrote:

> > The empty option was there exactly to provide a user in a hurry to have their previously selected theme restored after the login. Now, we argued that this user is a FWD dev and they need to select the theme from the drop-down every time to keep their preference.
>
> The empty option is harmful from a user perspective. It seems to require the user to make a choice, when we don't really want the standard user to make a choice. This is why I want the empty option to be removed.

For UX it was bad, but it could have had some text like italic "Restore preference".

#121 - 01/26/2024 07:28 AM - Greg Shah

Galya B wrote:

> Greg Shah wrote:
>
> > > The empty option was there exactly to provide a user in a hurry to have their previously selected theme restored after the login. Now, we argued that this user is a FWD dev and they need to select the theme from the drop-down every time to keep their preference.
> >
> > The empty option is harmful from a user perspective. It seems to require the user to make a choice, when we don't really want the standard user to make a choice. This is why I want the empty option to be removed.
>
> For UX it was bad, but it could have had some text like italic "Restore preference".

I understand. Since this is not expected to be a primary use case for end users, let's avoid the added complexity of "Restore Preference". Forcing an explicit default is a perfectly good solution here.

#122 - 01/26/2024 08:22 AM - Galya B

This is the test plan I'll be executing:

Case 1
Conditions:

- theme/name is configured with a valid theme, different from Win 10;
- available-themes is configured with multiple valid themes;
- The user needs to enter credentials (no auto-login, no forked session).

Result:

- The UI displays no theme select;
- The spawned client shows up with the theme from theme/name .

Case 2
Conditions:

- theme/name is configured with an invalid theme;
- available-themes is configured with multiple valid themes including Win 10;
- In client storage there is a previously stored, different from Win 10 theme for this user;
- The user needs to enter credentials (no auto-login, no forked session).

Result:

- The UI displays a theme select with Win 10 auto-selected;
- (The user doesn't change the theme) The spawned client shows up with the auto-selected theme.

Case 3
Conditions:

- theme/name is not configured;
- available-themes is configured with multiple valid themes excluding Win 10;
- In client storage there is a previously stored theme for this user, different from the first theme in alphabetical sort in available-themes;
- The user needs to enter credentials (no auto-login, no forked session).

Result:

- The UI displays a theme select with the first theme in alphabetical sort auto-selected;
- (The user doesn't change the theme) The spawned client shows up with the auto-selected theme.

Case 4
Conditions:

- theme/name is not configured;
- available-themes is configured with multiple valid themes excluding Win 10;
- In client storage there is a previously stored, different from the first theme in alphabetical sort theme for this user;
- the user needs to enter credentials (no auto-login, no forked session).

Result:

- The UI displays a theme select with the first theme in alphabetical sort auto-selected;
- (The user changes the theme) The spawned client shows up with the manually selected theme.

Case 5
Conditions:

- theme/name is not configured;
- available-themes is not configured;
- In client storage there is a previously stored, different from Win 10 theme for this user;
- The user needs to enter credentials (no auto-login, no forked session).

Result:

- The UI displays no theme select;
- The spawned client shows up with Win 10 theme.

Case 6
Conditions:

- theme/name is not configured;
- available-themes is not configured;
- In client storage there is a previously stored, different from Win 10 theme for this user;
- The user skips the login screen (auto-login or forked session).

Result:

- The spawned client shows up with Win 10 theme.

Case 7
Conditions:

- theme/name is configured with a valid theme;
- available-themes is configured with multiple valid themes;
- In client storage there is a previously stored, different from theme/name theme for this user;
- The user skips the login screen (auto-login or forked session).

Result:

- The spawned client shows up with the theme from theme/name .

Case 8
Conditions:

- theme/name is configured with an invalid theme;
- available-themes is configured with multiple valid themes excluding Win 10;
- In client storage there is a previously stored theme for this user, different from the first theme in alphabetical sort in available-themes, but still present in available-themes.
- The user skips the login screen (auto-login or forked session).

Result:

- The spawned client shows up with the theme from client storage.

#123 - 01/26/2024 08:30 AM - Greg Shah

In case 1, the presence of a valid theme/name means there should be no drop-down present in the UI. This changes your expected results.

#124 - 01/26/2024 08:32 AM - Galya B

Greg Shah wrote:

> In case 1, the presence of a valid theme/name means there should be no drop-down present in the UI. This changes your expected results.

It says "displays **no** theme select".

#125 - 01/26/2024 08:35 AM - Greg Shah

Galya B wrote:

> Greg Shah wrote:
>
> > In case 1, the presence of a valid theme/name means there should be no drop-down present in the UI. This changes your expected results.
>
> It says "displays **no** theme select".

Good, I misread it.

#126 - 01/29/2024 05:55 AM - Galya B

- Status changed from WIP to Review

- % Done changed from 80 to 100

Test cases executed.

Fixes and file headers committed.

8074b r14920 ready for review.

#127 - 02/07/2024 01:27 PM - Greg Shah

- Status changed from Review to Internal Test

Code Review Task Branch 8074b Revisions 14916 through 14920

I'm OK with the changes.

Is any additional testing needed?

#128 - 02/08/2024 01:39 AM - Galya B

Greg Shah wrote:

> Is any additional testing needed?

I don't think so.

#129 - 02/08/2024 08:27 AM - Greg Shah

Go ahead and rebase and then merge to trunk.

#130 - 02/08/2024 08:36 AM - Greg Shah

- Status changed from Internal Test to Merge Pending

#131 - 02/08/2024 08:44 AM - Galya B

- Status changed from Merge Pending to Internal Test

8074b was merged to trunk as rev. 14976 and archived.

#132 - 02/08/2024 08:52 AM - Greg Shah

- Status changed from Internal Test to Test

#133 - 02/14/2024 03:48 PM - Vladimir Tsichevski

1. The old way to set the theme using the client configuration parameters does not work anymore, so I always run in the Material theme. What is the correct way to configure the theme?
2. Also, this issue is never referenced in changed files history entries, so was not trivial for me to trace back the changes to this task :-(

#134 - 02/15/2024 03:53 AM - Galya B

Vladimir Tsichevski wrote:

> The old way to set the theme using the client configuration parameters does not work anymore, so I always run in the Material theme. What is the correct way to configure the theme?

- I don't know what is the "old way" you've used and what is the expected behavior in your opinion. The configurations in directory are theme/name and available-themes as always, the first taking precedence.
- As stated in #7847-231, the new improved behavior is described in details in #8074-101.
- If you need further explanation how to setup your configuration, let's discuss it via email.

Also, this issue is never referenced in changed files history entries, so was not trivial for me to trace back the changes to this task :-(

- If you can't trace "an issue", use #7143.
- If you want to find a related change, use #7847 and search for theme or for the revision number.
- If you have the revision number, but the ref is missing, write an email to the committer as a last resort.

#135 - 05/02/2024 07:46 AM - Galya B

This can be closed.

#136 - 05/02/2024 07:48 AM - Greg Shah

- Status changed from Test to Closed

Files

7143_ui_theme.png	27 KB	01/10/2024	Roger Borrello