User Interface - Feature #7814

Supporting translated texts for web login / logout screens

09/19/2023 02:25 AM - Galya B

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**vendor_id:**	GCD

Description

Related issues:

Related to User Interface - Feature #3931: single sign-on for virtual desktop...	**Closed**

History

#1 - 09/19/2023 02:43 AM - Galya B

Web login screens at /gui, /chui, /embedded and the respective logout screens introduced with SSO in #3931 need to support texts in the enabled current languages. The texts include the labels, buttons and error messages.

Current translation mechanics 1:

- The web driver shows certain messages and menus that reflect the CURRENT-LANGUAGE set by 4GL.
- CURRENT-LANGUAGE is by OpenEdge definition unknown (?) or a list of one or more languages (in full text), separated by comma, e.g. "English,German". FWD somehow uses locales instead of full text languages.
- A default value for CURRENT-LANGUAGE is set by directory config /server/runtime/default/currentLanguage used in EnvironmentOps.
- The web driver texts are present in p2j.js, p2j.screen.js and WebClientConstants.java.
- Using xgettext (part of the GNU gettext toolset) the translatable texts are extracted from the three files into a .po catalog. The step is manual on the dev workstation.
- The texts are translated in new .po files, preserving the English texts as keys.
- Node.js and gettext.js are installed on the dev workstation. gettext/bin/po2json.js is then used to convert the translated .po file to json file: po2json -p -F --fallback-to-msgid WebClient_de_DE.po WebClient_de_DE.json.
- On new web client being spawned the WebClientBuilderOptions adds "lang" for the locale corresponding to the CURRENT-LANGUAGE, e.g. de_DE. Then WebClientBuilder sends the value as client:web:lang to the client, where it's used by EmbeddedWebServerImpl as bundle identifier.
- The index.html config template built by WebPageHandler adds as "translations" the json with translated messages.
- The client subscribes to listen to CURRENT-LANGUAGE changes with thinClient.getEnvironmentDaemon().subscribe("client", "lang", this::setCurrentLanguage);.
- The front-end loads gettext.js and sets the locale and "translations" json, then uses i18n.gettext to translate the hardcoded English texts, if present in the loaded translations.

P.S. Although CURRENT-LANGUAGE is full text (e.g. "English") in 4GL, FWD adopted another approach and uses locale ("nl_NL").

P.S.S. There seems to be a bug where dynamically setting the CURRENT-LANGUAGE in 4GL loads the new translations in js, but doesn't change the texts in the ui.

Current translation mechanics 2:

- Texts to be translated in converted 4GL apps use TranslationManager on FWD server.
- TranslationManager loads java classes extending java.util.ResourceBundle where the translated texts reside.
- Using xgettext (part of the GNU gettext toolset) the translatable texts are extracted from the 4GL app files into a .po catalog. The step is manual on the dev workstation.
- The texts are translated in new .po files, preserving the original texts as keys.
- The .po catalogues are converted to Java ResourceBundle classes by msgfmt (part of the GNU gettext toolset). The step is manual on the dev workstation.

New requirements:

- Translate web login and logout pages and all errors that can show up, keeping the approach consistent to what's already in place.

Analysis:

- Login / logout pages and the related errors live outside the client session and can't directly use the current TranslationManager methods.
- The default value for CURRENT-LANGUAGE is currently server-side driven (defined in directory).
- Web client spawn / launch errors are critical to debugging end-user issues. It's preferable to be able to easily find the log corresponding to the end-user issue. It should be considered if it's appropriate the error to be translated only for the web ui and not in the logs. An id approach was proposed in #3931.
- Translating html labels in the front-end will likely cause flickering in the ui. Server-side rendering is only possible though if the locale is based on directory configs and not browser properties.

#3 - 09/19/2023 03:28 AM - Galya B

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

#4 - 09/19/2023 03:57 AM - Sergey Ivanovskiy

Galya B wrote:

> Current translation mechanics 1:
>
> - The web driver shows certain messages and menus that reflect the CURRENT-LANGUAGE set by 4GL.
> - CURRENT-LANGUAGE is by OpenEdge definition unknown (?) or a list of one or more languages (in full text), separated by comma, e.g. "English,German". FWD somehow uses locales instead of full text languages.

Are you asking a question?

#5 - 09/19/2023 03:58 AM - Galya B

Sergey Ivanovskiy wrote:

> Galya B wrote:
>
> > Current translation mechanics 1:
> >
> > - The web driver shows certain messages and menus that reflect the CURRENT-LANGUAGE set by 4GL.
> > - CURRENT-LANGUAGE is by OpenEdge definition unknown (?) or a list of one or more languages (in full text), separated by comma, e.g. "English,German". FWD somehow uses locales instead of full text languages.
>
> Are you asking a question?

Thanks for spotting this one. I was going to ask later. If you know the answer, it will be helpful to share it.

#6 - 09/19/2023 04:14 AM - Galya B

Greg, you want to preserve the current approach. If that's the case, then the login page should use currentLanguage locale from directory instead of the browser locale.

The cons of using the browser locale:

- it's not consistent with the current approach;
- one language has multiple locales, for example English has 13 and French 6, but translations are likely to provide only one, so locales will have to be mapped to languages instead of adding 6 times the same texts or loading the default English for a fr_BE user. On the other hand when the locale is defined as a config in directory, it's supposed to be provided in the resources, so there is no mapping;
- loading label and button texts in the ui with javascript will cause flickering, if the page is well written and layout shifts, if badly. Server-side rendering for such high traffic low functionality pages is in my opinion better, but the browser locale can't be determine with GET request;
- the high traffic low functionality pages will need to load a js library and all translations;
- the mere fact the login page can't change the language, but will load all languages;
- with auto-login the browser locale can't be fetched and fed to the client, except if the login page / script / flow is modified to show a loading screen on auto-login (currently it doesn't show a page, but directly redirects to the client url).

#7 - 09/19/2023 04:44 AM - Galya B

Another important aspect is to keep a correlation between the error messages displayed in the ui and the logs on the server-side for traceability. The end-user can cause only one type of error and that's invalid credentials. All other errors are due to misconfiguration on the admin side. Even so, it was decided in #3931 all errors to be translated and displayed on the ui. Then there are two approaches:

- translate all texts only client-side (related to the previous question in #7814#note-6) and add error id keeping a link to the non-translated errors in the logs. This approach introduces one additional manual step in converting the .po files to json;
- translate all texts only server-side (related to the previous question in #7814#note-6). This approach introduces one additional manual step in converting the .po files to java.util.ResourceBundle;
- translate errors server-side and log them translated and translate labels client-side. This approach introduces two additional manual steps in converting the .po files to both json and java.util.ResourceBundle;

In any case the requirement for serving translated errors to the ui would mean to identify and move all nested (for example in BrokerManager) client spawn errors to one java class and the generate .po file from it. This will be a required manual step newly introduced for the FWD runtime. I say manual, because it depends on external tools, although it can be added as dependency to ant-jar presuming it's always executed only on Linux with gettext present (which I wouldn't be completely confident in). Also generating .po files without translating them wouldn't bring much of a benefit. In any case it's still a new step to be taken care of.

The approach with bundles in .properties files doesn't require any additional steps of extracting or converting .po files and it scales better with more texts, because the keys are stable, while full texts used with .po files can be modified (on fixing wrong wording or spelling mistake) and this breaks the translations.

So I'll be waiting for final decision on locale source (server-side / client-side rendering) and approval of the additional manual steps that will have to be maintained in the FWD runtime.

#8 - 09/19/2023 04:47 AM - Sergey Ivanovskiy

Galya B wrote:

> Sergey Ivanovskiy wrote:
>
> > Galya B wrote:
> >
> > > Current translation mechanics 1:
> > >
> > > - The web driver shows certain messages and menus that reflect the CURRENT-LANGUAGE set by 4GL.
> > > - CURRENT-LANGUAGE is by OpenEdge definition unknown (?) or a list of one or more languages (in full text), separated by comma, e.g. "English,German". FWD somehow uses locales instead of full text languages.
> >
> > Are you asking a question?
>
> Thanks for spotting this one. I was going to ask later. If you know the answer, it will be helpful to share it.

Actually, this is a question because I see from the code that TranslationManager.getIdentifier does not map English to en_+country_code

```
/**
 * Converts the supplied string to a valid Java identifier by skipping the unsupported chars from the
 * result.
 *
 * @param   str
 *          The input string.
 *
 * @return  Valid Java identifier.
 */
public static String getIdentifier(String str)
{
   StringBuilder sb = new StringBuilder();
   for (int i = 0; i < str.length(); i++)
   {
      if ((i == 0 && Character.isJavaIdentifierStart(str.charAt(i))) ||
         (i > 0 && Character.isJavaIdentifierPart(str.charAt(i))))
      {
         sb.append(str.charAt(i));
      }
   }
   return sb.toString();
}
```

4GL statement CURRENT-LANGUAGE = ?. is converted into EnvironmentOps.setCurrentLanguage

```
/**
 * Sets the name of the current language being used for session.
 *
 * @param    val
 *           The new CURRENT-LANGUAGE value.
 */
public static void setCurrentLanguage(String val)
{
   work.obtain().currentLanguage = val;
   TranslationManager tm = TranslationManager.getInstance();
   tm.setCurrentLanguages(val);
   Environments.setKeyValue("client", "lang", val);
}
```

Thus there is a deviation with 4GL in language identifiers. At that moment the code can work correctly only if the input val is ISO language and countries identifier in this form language_code _ country_code.
So you found a hot issue.

#9 - 09/19/2023 04:49 AM - Galya B

Sergey Ivanovskiy wrote:

> So you found a hot issue.

Yep. I thought it's intentional and customer apps are converted with some modification to support it.

#10 - 09/19/2023 05:04 AM - Hynek Cihlar

Galya B wrote:

- one language has multiple locales, for example English has 13 and French 6, but translations are likely to provide only one, so locales will have to be mapped to languages instead of adding 6 times the same texts or loading the default English for a fr_BE user. On the other hand when the locale is defined as a config in directory, it's supposed to be provided in the resources, so there is no mapping;

This is typically solved by making the target language bundle resolution hierarchical. I.e. when the requested language is fr_CH and the bundle WhateverBundle_fr is available (without the country suffix) then this country-agnostic bundle will be resolved. When WhateverBundle_fr and WhateverBundle_fr_CH are both present, then WhateverBundle_fr_CH will be resolved.

I think this is the way we should go, too.

- loading label and button texts in the ui with javascript will cause flickering, if the page is well written and layout shifts, if badly.

Why is that? If the language is known (by falling back to browser's locale) then the correct texts will be used for the first-time render.

- the high traffic low functionality pages will need to load a js library and all translations;

The bundles are served by the web server as the other resources. So only the bundle for the target language is needed and loaded from the web server.

- the mere fact the login page can't change the language, but will load all languages;

Only the bundle for the target language needs to be loaded.

#11 - 09/19/2023 05:07 AM - Hynek Cihlar

Galya B wrote:

> Sergey Ivanovskiy wrote:
>
> > So you found a hot issue.
>
> Yep. I thought it's intentional and customer apps are converted with some modification to support it.

CURRENT-LANGUAGE is interpreted by the customer app, it can be any valid identifier, not just ISO codes. For FWD runtime we will need to deal with ISO codes and so some mapping will be required.

#12 - 09/19/2023 05:13 AM - Sergey Ivanovskiy

It seems that enough to fix TranslationManager.parse (transfroms to identifiers) and TranslationManager.getIdentifier but the other code that uses them can be preserved.

#13 - 09/19/2023 05:17 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> It seems that enough to fix TranslationManager.parse (transfroms to identifiers) and TranslationManager.getIdentifier but the other code that uses them can be preserved.

For the legacy code we still need to keep the legacy-code-assigned identifiers. They are needed to resolve the resource bundles for the legacy app. We will need to keep both, the legacy and the ISO ones.

#14 - 09/19/2023 05:20 AM - Sergey Ivanovskiy

We should decide who takes this issue because it can block this task.

#15 - 09/19/2023 05:27 AM - Sergey Ivanovskiy

Then in this EmbeddedWebServerImpl

```
      public Supplier<Entry<String, String>> getTranslations(String lang,
                                                              Set<String> cachedLanguages,
```

```
                                                        String defaultValue)
    {
       return () ->
       {
          List<String> langs = TranslationManager.parseLanguages(lang, true);
          Iterator<Entry<String, String>> bundles = TranslationManager.getBundles(bundleBaseName, langs);
          Resource jsonResource = null;
          boolean found = false;
          boolean isCodeLanguage = false;
          String langIdentifier = null;
          String firstLangIdentifier = null;
          while (bundles.hasNext())
```

this method

```
List<String> langs = TranslationManager.parseLanguages(lang, true);
```

should be replaced with

```
List<String> langs = TranslationManager.toISOLanguagesCodes(lang);
```

toISOLanguagesCodes should return ISO codes iff there is a correponding value in the language codes map or unchanged identifiers.

#16 - 09/19/2023 05:33 AM - Sergey Ivanovskiy

I can apply the changes from #7814-15 to the required task branch if there are no objections.

#17 - 09/19/2023 05:52 AM - Galya B

Hynek Cihlar wrote:

- loading label and button texts in the ui with javascript will cause flickering, if the page is well written and layout shifts, if badly.

> Why is that? If the language is known (by falling back to browser's locale) then the correct texts will be used for the first-time render.

There will be nothing in the html on first-time render. gettext works in js.

- the high traffic low functionality pages will need to load a js library and all translations;

> The bundles are served by the web server as the other resources. So only the bundle for the target language is needed and loaded from the web server.

Hynek, I believe you don't follow everything that is written. Currently what you say is right, but you don't read the change in the requirements for the new implementation.

- the mere fact the login page can't change the language, but will load all languages;

> Only the bundle for the target language needs to be loaded.

That is impossible with locale coming from the browser.

#18 - 09/19/2023 05:53 AM - Galya B

Sergey Ivanovskiy wrote:

> We should decide who takes this issue because it can block this task.

I think it's also blocked by #3931:

- the logout page is added there;
- there is a major change in the html and js of the login page.

#19 - 09/19/2023 05:56 AM - Galya B

Hynek Cihlar wrote:

> CURRENT-LANGUAGE is interpreted by the customer app, it can be any valid identifier, not just ISO codes. For FWD runtime we will need to deal with ISO codes and so some mapping will be required.

You can't have that freedom developing a framework. currentLanguages is locale to resolve the bundles, but it's also set to CURRENT-LANAGUAGE and sent to the 4GL app. Are you suggesting that WebClient_nl_NL.json translation file part of the FWD repo will be renamed for a customer using full text languages?

#20 - 09/19/2023 06:08 AM - Hynek Cihlar

Galya B wrote:

> Hynek Cihlar wrote:
>
> - loading label and button texts in the ui with javascript will cause flickering, if the page is well written and layout shifts, if badly.
>
> Why is that? If the language is known (by falling back to browser's locale) then the correct texts will be used for the first-time render.
>
> There will be nothing in the html on first-time render. gettext works in js.

The point is to supply already translated messages. The logic responsible for rendering the page on the Java side will use the Java resource bundles and the Java side translation logic. The strings rendered on JS (the strings defined in JS sources) will be translated in JS using gettext js.

> - the high traffic low functionality pages will need to load a js library and all translations;
>
> The bundles are served by the web server as the other resources. So only the bundle for the target language is needed and loaded from the web server.
>
> Hynek, I believe you don't follow everything that is written. Currently what you say is right, but you don't read the change in the requirements for the new implementation.

Well, the main requirement is to use gettext as a proven and approved solution and do it right.

> - the mere fact the login page can't change the language, but will load all languages;
>
> Only the bundle for the target language needs to be loaded.
>
> That is impossible with locale coming from the browser.

How is that? You ask the language code and fetch the resource bundle according to the code. If you have a session and get the language by other means you use that preferably.

#21 - 09/19/2023 06:10 AM - Hynek Cihlar

Galya B wrote:

> Hynek Cihlar wrote:
>
> > CURRENT-LANGUAGE is interpreted by the customer app, it can be any valid identifier, not just ISO codes. For FWD runtime we will need to deal with ISO codes and so some mapping will be required.
>
> You can't have that freedom developing a framework. currentLanguages is locale to resolve the bundles, but it's also set to CURRENT-LANAGUAGE and sent to the 4GL app. Are you suggesting that WebClient_nl_NL.json translation file part of the FWD repo will be renamed for a customer using full text languages?

I have been suggesting to map the legacy codes assigned to CURRENT-LANGUAGE to ISO codes. The translation will be per project defined in server directory as the legacy language codes have meaning to each specific legacy app.

#22 - 09/19/2023 06:11 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> I can apply the changes from #7814-15 to the required task branch if there are no objections.

Sergey, do we actually need the bundles to be served by TranslationManager for JS driver? Can't we just load it directly from the web server as all the other resources?

#23 - 09/19/2023 07:20 AM - Sergey Ivanovskiy

Did you mean that this code Iterator<Entry<String, String>> bundles = TranslationManager.getBundles(bundleBaseName, langs); can be handled directly by I18NWebResources.getTranslations?

#24 - 09/19/2023 08:27 AM - Greg Shah

Greg, you want to preserve the current approach. If that's the case, then the login page should use currentLanguage locale from directory instead of the browser locale.

The cons of using the browser locale:

- it's not consistent with the current approach;

The idea is make it nicer for users of a multi-lingual application, such that we can choose a reasonable language based purely on the user's existing browser settings. To use the default language configured in the directory will work if the admin has configured the same language as the user is using, but is not nice for the case the user has a different language.

I agree, it is different than the standard CURRENT-LANGUAGE approach. If we can't make it work cleanly, then we can drop the idea but it does seem to be an improvement from the end-user's perspective.

- the high traffic low functionality pages will need to load a js library and all translations;

I don't understand this point. The FWD web client is a single-page application.

- with auto-login the browser locale can't be fetched and fed to the client, except if the login page / script / flow is modified to show a loading screen on auto-login (currently it doesn't show a page, but directly redirects to the client url).

We can always fetch the browser locale when we load the web client itself.

#25 - 09/19/2023 08:29 AM - Galya B

Greg Shah wrote:

- the high traffic low functionality pages will need to load a js library and all translations;

I don't understand this point. The FWD web client is a single-page application.

FWD web client is already translated. We're speaking about the login / logout page that are not part of the FWD web client.

- with auto-login the browser locale can't be fetched and fed to the client, except if the login page / script / flow is modified to show a loading screen on auto-login (currently it doesn't show a page, but directly redirects to the client url).

> We can always fetch the browser locale when we load the web client itself.

The web client is not loaded on the login / logout page.

#26 - 09/19/2023 08:30 AM - Hynek Cihlar

Sergey Ivanovskiy wrote:

> Did you mean that this code Iterator<Entry<String, String>> bundles = TranslationManager.getBundles(bundleBaseName, langs); can be handled directly by I18NWebResources.getTranslations?

I meant to get the bundle directly:

```
function initI18n()
{
  // figure out the locale (browser, session, params, whatever)
  let lang = ...;

  // compute the bundle path based on the lang above
  let bundle = ... lang ...;

  // request the bundle from server (TODO: error handling, caching, etc.)
  fetch(bundle)
    .then(response => {
        return response.json();
    })
    .then(data => {
       p2j.i18n.loadJSON(data, "messages");
    });
}
```

No need to involve TranslationManager except using the current languages it holds.

#27 - 09/19/2023 08:33 AM - Greg Shah

Galya B wrote:

> Greg Shah wrote:
>
> - the high traffic low functionality pages will need to load a js library and all translations;
>
> I don't understand this point. The FWD web client is a single-page application.
>
> FWD web client is already translated. We're speaking about the login / logout page that are not part of the FWD web client.

OK. As noted by Hynek, I would expect only the needed translations to be sent down. Also, are we really worried about the translations of 20 +/- strings being too big?

> - with auto-login the browser locale can't be fetched and fed to the client, except if the login page / script / flow is modified to show a loading screen on auto-login (currently it doesn't show a page, but directly redirects to the client url).
>
> We can always fetch the browser locale when we load the web client itself.
>
> The web client is not loaded on the login / logout page.

Right, so why do we care about translations if no page is being loaded?

#28 - 09/19/2023 08:39 AM - Galya B

Greg Shah wrote:

> The web client is not loaded on the login / logout page.

Right, so why do we care about translations if no page is being loaded?

What? The login page (html, css, js, images) is loaded. Web client being loaded is something else, especially in the context of FWD. So... I kind of see where this is going...

An empty html styled by css, without texts on first render, then default English labels loaded by default by gettext, while waiting for the request to the back-end to fetch the actual texts and then loading them in the 3rd rendering of the 5 labels page.

Or you mean no default English labels before the resources come back? So.. if any issue, no texts at all on the page.

#29 - 09/19/2023 08:44 AM - Galya B

Also SPA is a form of web app I'm very well familiar and it loads all resources at once in a compressed format, so it won't be loading more resources on demand later on. It's not found in FWD and won't be.

#30 - 09/19/2023 08:45 AM - Sergey Ivanovskiy

Hynek Cihlar wrote:

> Sergey Ivanovskiy wrote:
>
> > Did you mean that this code Iterator<Entry<String, String>> bundles = TranslationManager.getBundles(bundleBaseName, langs); can be handled directly by I18NWebResources.getTranslations?
>
> I meant to get the bundle directly:
>
> [...]
>
> No need to involve TranslationManager except using the current languages it holds.

It seems that we need to translate progress bar messages. At this moment I do not understand what are the issues related to the current implementation.

#31 - 09/19/2023 08:45 AM - Hynek Cihlar

Galya B wrote:

> Greg Shah wrote:
>
> > The web client is not loaded on the login / logout page.
>
> Right, so why do we care about translations if no page is being loaded?

What? The login page (html, css, js, images) is loaded. Web client being loaded is something else, especially in the context of FWD. So... I kind of see where this is going...

An empty html styled by css, without texts on first render, then default English labels loaded by default by gettext, while waiting for the request to the back-end to fetch the actual texts and then loading them in the 3rd rendering of the 5 labels page.

No, the page should load with the correct target language (assuming the target language can be determined on the first load from the request, browser locale, session, etc.). Whatever gets rendered with the Java page rendering engine on FWD server gets translated there with Java-side gettext, whatever gets rendered on JS with strings appearing in JS sources is rendered in JS gettext with the correct bundle fetched from web server.

#32 - 09/19/2023 08:48 AM - Hynek Cihlar

Galya B wrote:

> Also SPA is a form of web app I'm very well familiar and it loads all resources at once in a compressed format, so it won't be loading more resources on demand later on. It's not found in FWD and won't be.

How and when is the json bundle delivered to the browser is an implementation detail. It can be also delivered together with the rendered initial page on the first request.

#33 - 09/19/2023 09:03 AM - Greg Shah

> The web client is not loaded on the login / logout page.

Right, so why do we care about translations if no page is being loaded?

> What? The login page (html, css, js, images) is loaded. Web client being loaded is something else, especially in the context of FWD. So... I kind of see where this is going...

This was in regard to auto-login where we don't need to load a page.