

Examples

- [Minimal Rebranding](#)
- [Rebranded Application](#)
- [Minimal User Interaction](#)

Minimal Rebranding

[top](#)

In this example, the installer is rebranded with a minimum of work. References to “Yellowfin” in the installer user-interface are replaced, and some images are replaced with rebranded versions.

First, set the name of the application as displayed in the installer:

```
installer.appname=Real Reports
```

Override text properties that refer to “Yellowfin”:

```
installer.footer.msg= -- www.realreports.com --  
installer.licencefilepanel.msg=A valid licence file is required to install and use Real Reports. If you do not  
have a licence file, please see your software provider.  
installer.finishpanel.supportlink.url=http://www.example.com/support
```

Override Windows service name and description:

```
installer.winservice.name=Real Reports  
installer.winservice.description=This service starts the Real Reports Server
```

Tell the installer not to install any Yellowfin-branded sample content:

```
action.yellowfincontent=false
```

Create a new html file to display at the start of the graphical installer, and add a property to use the new html:
resource.infopanel.html=resources/installsplash.html

Create new images to use as the pack icons in the graphical installer (one for each pack). Add properties to use these new images:

```
resource.pack.core.img=resources/corepack.png  
resource.pack.tutorial.img=resources/tutorialpack.png  
resource.pack.geopack.us.img=resources/usgeopack.png  
resource.pack.geopack.world.img=resources/worldgeopack.png
```

Put all the properties above into a file called `custom.properties`:

```

# text properties
installer.appname=Real Reports
installer.footer.msg= == www.realreports.com ==
installer.licencefilepanel.msg=A valid licence file is required to install and use Real Reports. If you do not
have a licence file, please see your software provider.
installer.finishpanel.supportlink.url=http://www.example.com/support
installer.winservice.name=Real Reports
installer.winservice.description=This service starts the Real Reports Server

# installer actions
action.yellowfincontent=false

# resources
resource.infopanel.html=resources/installsplash.html
resource.pack.core.img=resources/corepack.png
resource.pack.tutorial.img=resources/tutorialpack.png
resource.pack.geopack.us.img=resources/usgeopack.png
resource.pack.geopack.world.img=resources/worldgeopack.png

```

To create the customised installer:

1. Extract the contents of the existing installer jar file

```
jar xvf /path/to/yellowfin-yyyymmdd-full.jar
```

2. Create a new directory called resources and copy the new html file and image files into it
3. Copy the custom properties file created above into the root directory of the install folder structure.
The directory should now contain the existing files from the installer archive, as well as your new files, eg:

```

META-INF/MANIFEST.MF
custom.properties
com/
default.properties
resources/corepack.png
resources/tutorialpack.png
resources/usgeopack.png
resources/worldgeopack.png
resources/installsplash.html
...

```

4. Create the modified jar file

```
jar cvmf META-INF/MANIFEST.MF /path/to/yellowfin-modified.jar
```

Rebranded Application

[top](#)
In the [Minimal Rebranding](#) example, elements of the installer were rebranded, but the Yellowfin application itself was not. This example builds further on this by providing rebranding for some application UI elements.

Set the application name that is displayed within the Yellowfin application (not the installer):

```
action.config.appname=Real Reports
```

Change the names of some files created in the installation directory:

```

installer.warfilename=realreports.war
installer.applogfilename=realreports.log

```

Change the support email address:

```
action.config.supportemail=support@example.com
```

Change the email address and name that broadcast emails are sent from:

```
action.config.broadcastemail=noreply@example.com  
action.config.broadcastname=Real Reports
```

Change the credentials of the default administrator logon account:

```
action.adminuser.username=admin@example.com  
action.adminuser.password=secret  
action.adminuser.email=admin@example.com
```

Replace Yellowfin-branded images used in the Yellowfin application. Firstly, create a subdirectory under `resources` called `customimages`. Then create new images for each image you want to replace, and save them into the `customimages` directory using the same filenames as the original images. In this example we create new versions of the files `header_logo.gif`, `logo.png` and `logoFooter.png`, saving them into the `customimages` directory.

```
action.customimagespath=resources/customimages
```

Create a custom login page, and add it to the `resources` directory:

```
action.customindexpage=resources/login.html
```

The custom login page also uses a few image files which are not part of Yellowfin. These can be added to the `customimages` directory we created above, and will automatically be added to the application.

Combining these properties with those in Example 1, the full custom properties file now contains:

```
# text properties  
installer.appname=Real Reports  
installer.footer.msg= -- www.realreports.com ==  
installer.licencefilepanel.msg=A valid licence file is required to install and use Real Reports. If you do not  
have a licence file, please see your software provider.  
installer.finishpanel.supportlink.url=http://www.example.com/support  
installer.winservice.name=Real Reports  
installer.winservice.description=This service starts the Real Reports Server  
  
# installer actions  
action.yellowfincontent=false  
installer.warfilename=realreports.war  
installer.applogfilename=realreports.log  
  
# post-installation application behaviour  
action.config.appname=Real Reports  
action.config.supportemail=support@example.com  
action.config.broadcastemail=noreply@example.com  
action.config.broadcastname=Real Reports  
action.adminuser.username=admin@example.com  
action.adminuser.password=secret  
action.adminuser.email=admin@example.com  
action.customindexpage=resources/login.html  
action.customimagespath=resources/customimages  
  
# resources  
resource.infopanel.html=resources/installsplash.html  
resource.pack.core.img=resources/corepack.png  
resource.pack.tutorial.img=resources/tutorialpack.png  
resource.pack.geopack.us.img=resources/usgeopack.png  
resource.pack.geopack.world.img=resources/worldgeopack.png
```

The jar file will now contain:

```
META-INF/MANIFEST.MF
custom.properties
com/
default.properties
resources/corepack.png
resources/tutorialpack.png
resources/usgeopack.png
resources/worldgeopack.png
resources/installsplash.html
resources/login.html
resources/customimages/header_logo.gif
resources/customimages/logo.png
resources/customimages/logoFooter.png
resources/customimages/loginimage1.png
resources/customimages/loginimage2.png
resources/customimages/loginimage3.png
```

Minimal User Interaction

[top](#)

This example shows how to create an installer where most user interaction is bypassed. The user only has to select the target installation directory. All other options are pre-selected and hidden from the user.

Custom properties file contents:

```
# Set default values for installation options
option.pack.tutorial=true
option.pack.geopack.us=false
option.pack.geopack.world=false
option.serverport=8181
option.winservice=false
option.db.dbtype=POSTGRESQL
option.db.hostname=localhost
option.db.dbname=yellowfin
option.db.createdb=true
option.db.username=yfuser
option.db.userpassword=yfpassword
option.db.createuser=true
option.db.dbusername=postgres
option.db.dbpassword=secret

# Tell the installer to hide various options from the user
action.hidedboptions=true
action.hidepacks=true
action.hideserverport=true
action.hidewinservice=true

# Embed a licence file in the installer
resource.licencefile=resources/licence.lic
```

In this example, the installer will create a new Yellowfin configuration database in a PostgreSQL server running on localhost (on the default port) called "yellowfin". The installer will connect using the username "postgres" and the password "secret" to create the database, and will create a user "yfuser" with password "yfpassword".

The installer archive must also contain a valid licence file for the installation at resources/licence.lic.

[top](#)