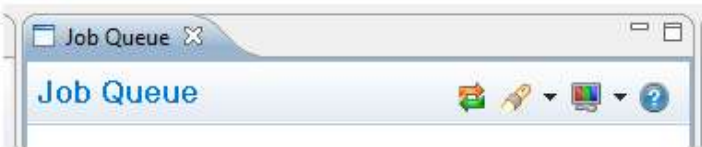


4. FD1 and FD2, FD Download and FD Web: Meaning and differences v1.0 2023-04-27

- **FD1:** Name for all FD components that have been developed in the early years, using the *Eclipse Rich Client Platform (RCP)* framework. **Only FD Download contains FD1 components.** You can recognize them in FD Download by the header of the views, which is slightly different from the FD2 view headers:

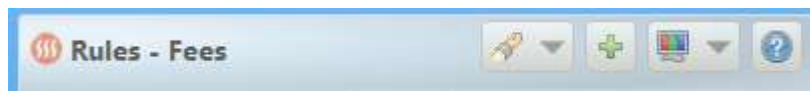
FD1 view in FD Download



FD2 view in FD Download



- **FD2:** Name for all FD components that have been developed from approx. 2014 on (when Asgard support in FD started), using the *Eclipse Remote Application Platform (RAP)* framework, resp. the *RAP Widget Toolkit (RWT)*. **FD Web consists only of FD2 components** (FD Download, in contrast, contains FD1 and FD2 components). This is possible, because the RWT API is based on RCP's Standard Widget Toolkit (SWT). All the Java code that is written for RWT (to be used in FD Web) also runs in the RCP world.
- **FD Download** (a.k.a. FD Desktop): The FD client based on the RCP framework that QC users download from nexus (path *net/quintessence/client/flightdeck*) for each release. It is a fat, single user client which runs on the user's desktop computer.
- **FD Web:** The FD client built with RWT. It is a plain java, multi user web application (in contrast to FD Download, it does not use RAP's workbench or OSGI mechanism):



Build and delivery:

- **FD Download:** The build is built via Jenkins (<http://cvs:8180/jenkins/job/quintessence-rcp/>) and available from nexus. It always contains everything - all Midgard, Kronos, Eos, Top-Client and other functionalities. We provide one global build per release for download on our homepage (<https://www.quintessence.net/fd/>). Hence, we have no versioning, there is always only one latest state for each release.
- **FD Web:** The different FD2 functionalities are separated in different modules (e.g. *Core, Midgard, Asgard, User Management* and so on). These modules are built on Jenkins (see <http://cvs:8180/jenkins/view/FD-M/>). The actual application is always a set of these modules. This is called a bundle. There is one job in Jenkins for building the bundles (<http://cvs:8180/jenkins/view/FD-B/job/quintessence-client-flightdeck/>). These bundles only contains the modules required for the actual purpose. Currently, we have these bundles:
 - **Asgard Portal:** Only Asgard, with an "Asgard Portal" flag, to indicate the portal mode (modules: *Core, RWT, Asgard*)
 - **Asgard:** For Midgard servers with Asgard (modules: *Core, RWT, User Management, Flight, Midgard, Asgard*)
 - **Dayreturns:** For Midgard servers with Dayreturns (modules: *Core, RWT, User Management, Flight, Midgard, Dayreturns*)
 - **Eos:** For plain Eos server (modules: *Core, RWT, User Management, Flight, Eos*)
 - **Eosportal:** Portal for tour operators for Eos (modules: *Core, RWT, Flight, Eosportal*)
 - **Kronos:** For plain Kronos server (modules: *Core, RWT, User Management, Flight, Kronos*)
 - **Midgard:** For plain Midgard server (modules: *Core, RWT, User Management, Flight, Midgard*)
 - **Rescount Portal:** Portal for tour operators for Midgard (modules: *Core, RWT, Flight, Rescount Portal*)
 - **Top Client:** For plain Top Client server (modules: *Core, RWT, Flight, Top Client*)
 - **Top Midgard:** Midgard and Asgard with Tour Operator Midgard (modules: *Core, RWT, User Management, Flight, Midgard,*

Top Midgard, Asgard)

- **Plain:** Only core functionalities (modules: *Core, RWT, User Management*)

All these bundles are available as war files from nexus (path *net/quintessence/client/quintessence-client-flightdeck*) for each release. All bundles will be built when the Jenkins job runs.

Installation: You can download the desired bundle's war file from nexus and install it on a server and configure the connection (see other documents). By default, FD Web bundles are also part of the Midgard, Kronos, Eos and Topclient server installation packages and will automatically be available. Which bundles are added for which customer is configured in the QC assembly project (*quintessence-installation-assembly*). Example from the file *midgard-custom-to.xml* (for customer TO):

```
<dependencySets>
  <dependencySet>
    <useProjectArtifact>false</useProjectArtifact>
    <outputDirectory>${packageDir}</outputDirectory>
    <includes>
      <include>net.quintessence.midgard.custom:quintessence-midgard-custom-to:jar:${releasename}</include>
      <include>net.quintessence.client:quintessence-client-flightdeck-custom:jar:${releasename}-custom-to</inc
      <include>net.quintessence.priceclassmapping:qc-priceclassmapping-integration:war</include>
      <include>net.quintessence.client:quintessence-client-flightdeck:war:dayreturns</include>
    </includes>
  </dependencySet>
  <dependencySet>
    <useProjectArtifact>false</useProjectArtifact>
    <outputDirectory>${packageDir}</outputDirectory>
    <includes>
      <include>net.quintessence.client:quintessence-client-flightdeck:war:rescount-portal</include>
    </includes>
    <outputFileNameMapping>TourOpSales.${artifact.extension}</outputFileNameMapping>
  </dependencySet>
</dependencySets>
```

Customer TO uses dayreturns, so the dayreturns bundle will be included

Customer TO also uses the rescount portal, so this bundle will also be included...

... with the war file named "TourOpSales"

For each release, the FD Web version to add to the installation package is defined in the same project, in the *pom.xml* (tag *<flightdeck.version>*).