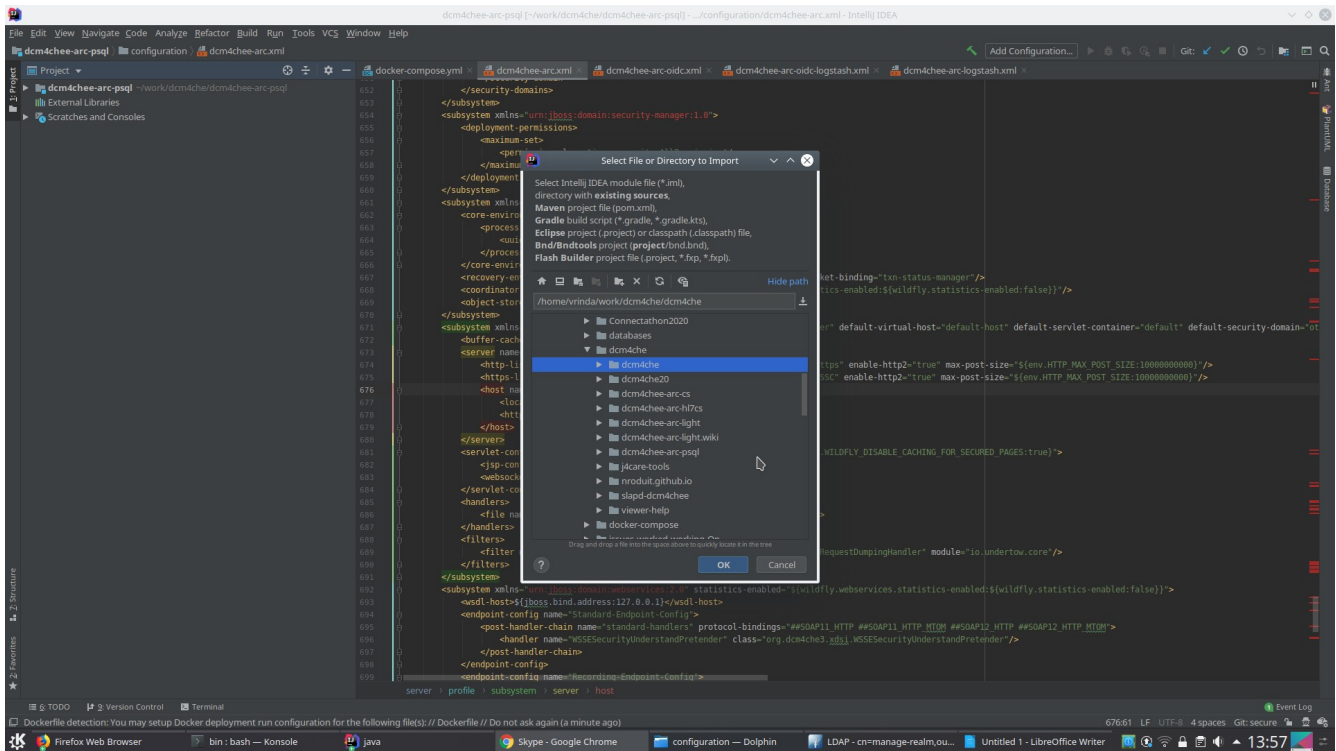
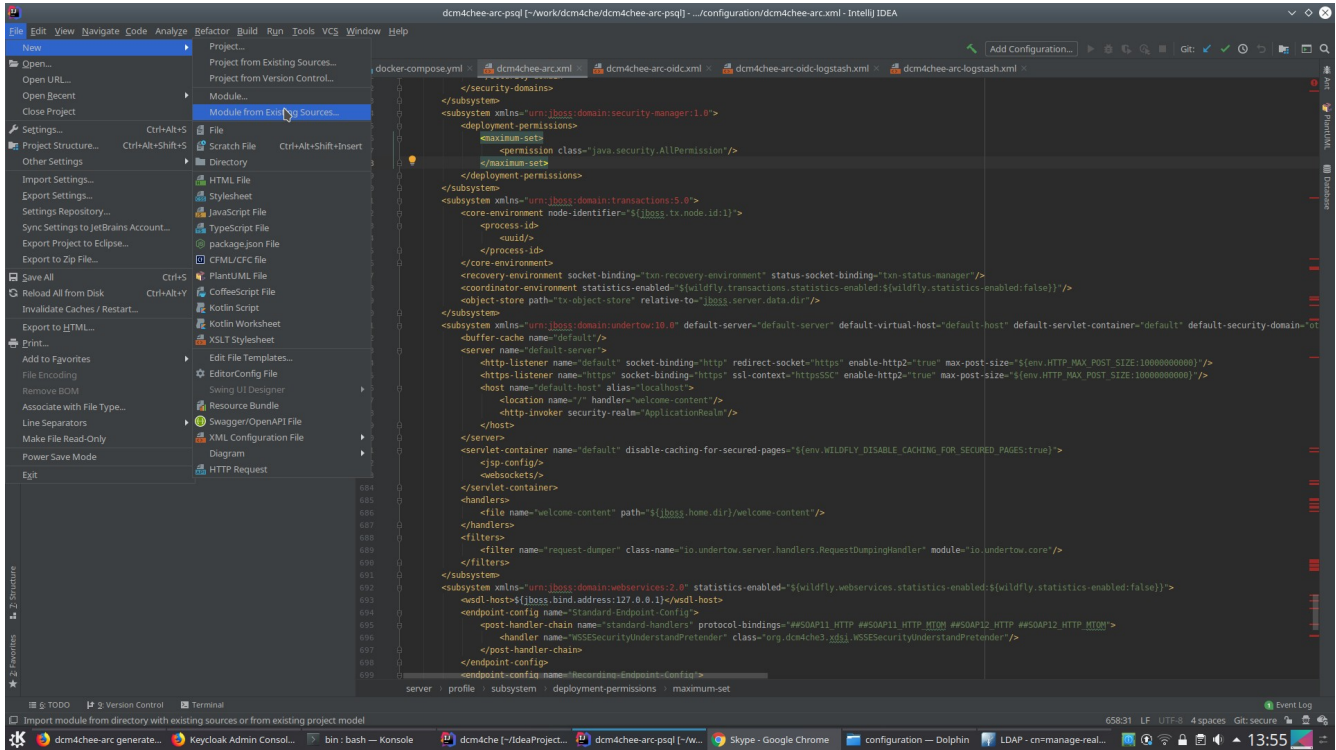
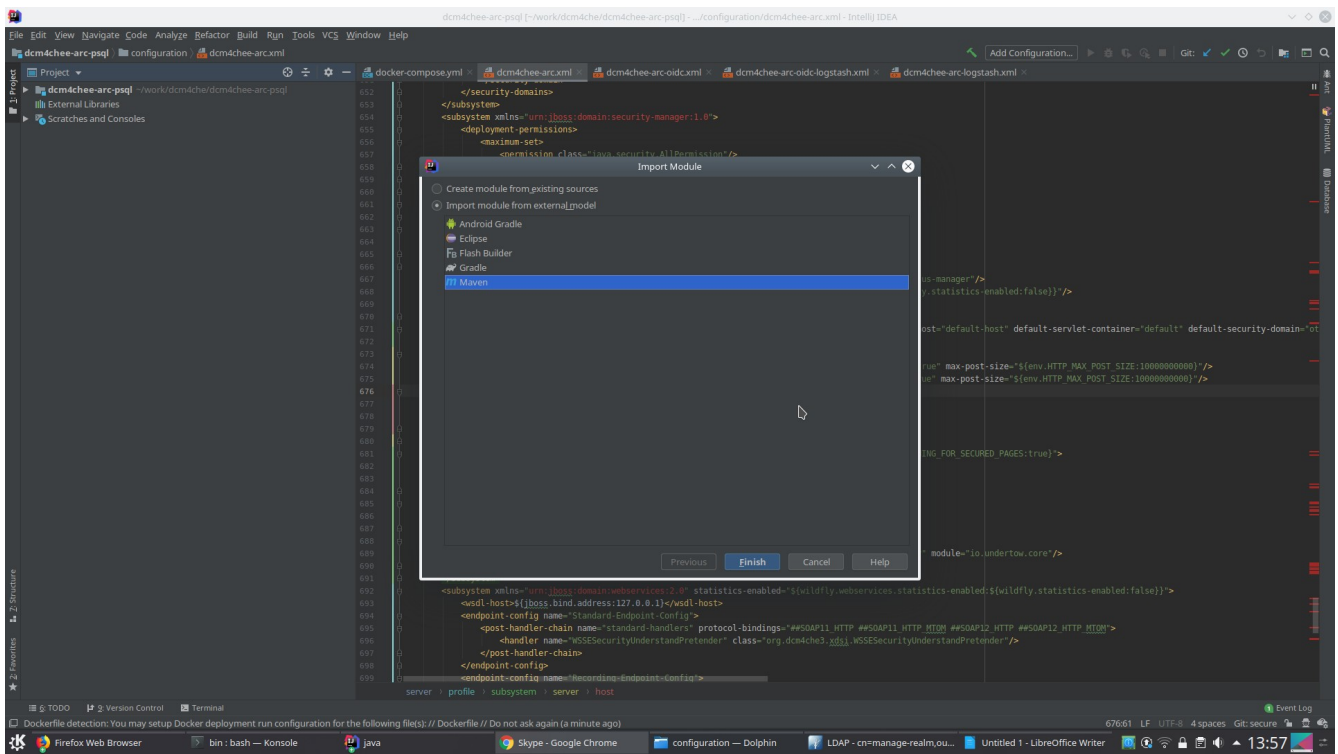


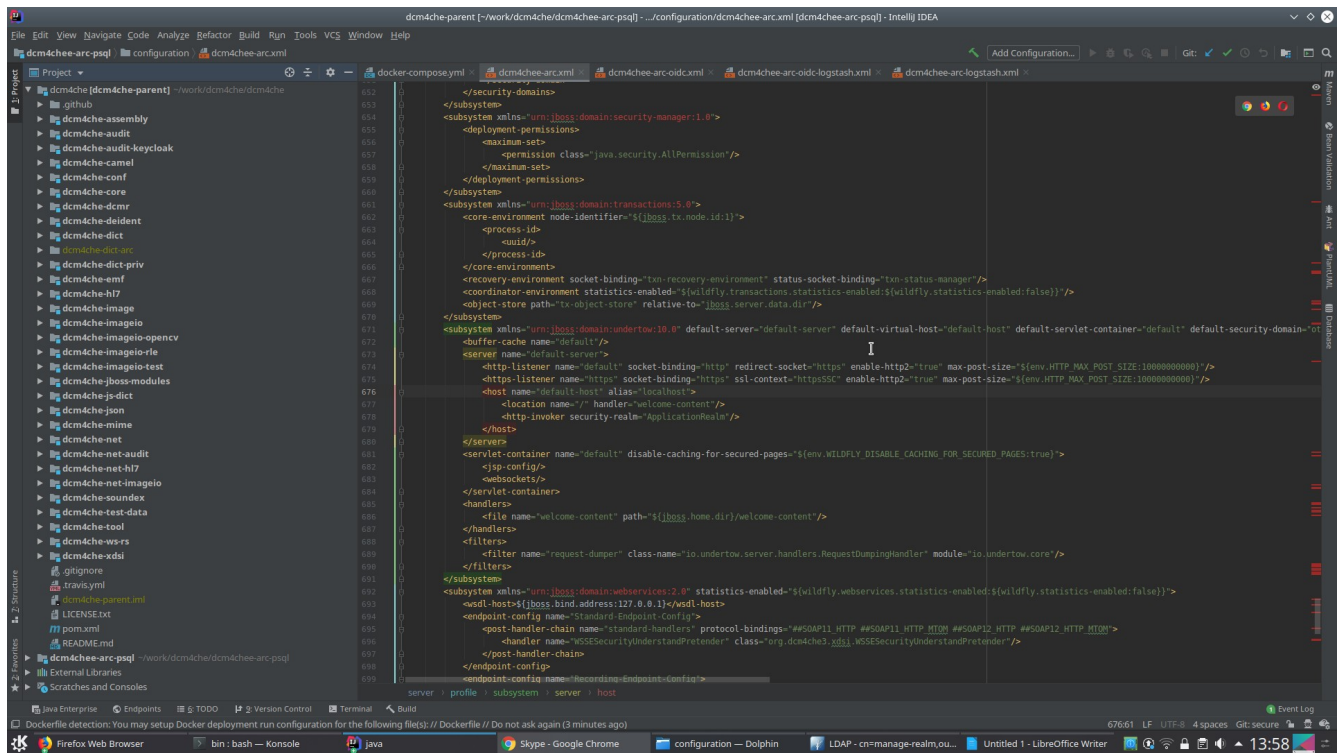
Git clone / pull *dcm4che* library and *dcm4chee-arc-light*

Add *dcm4che* library as *Module from Existing Sources*

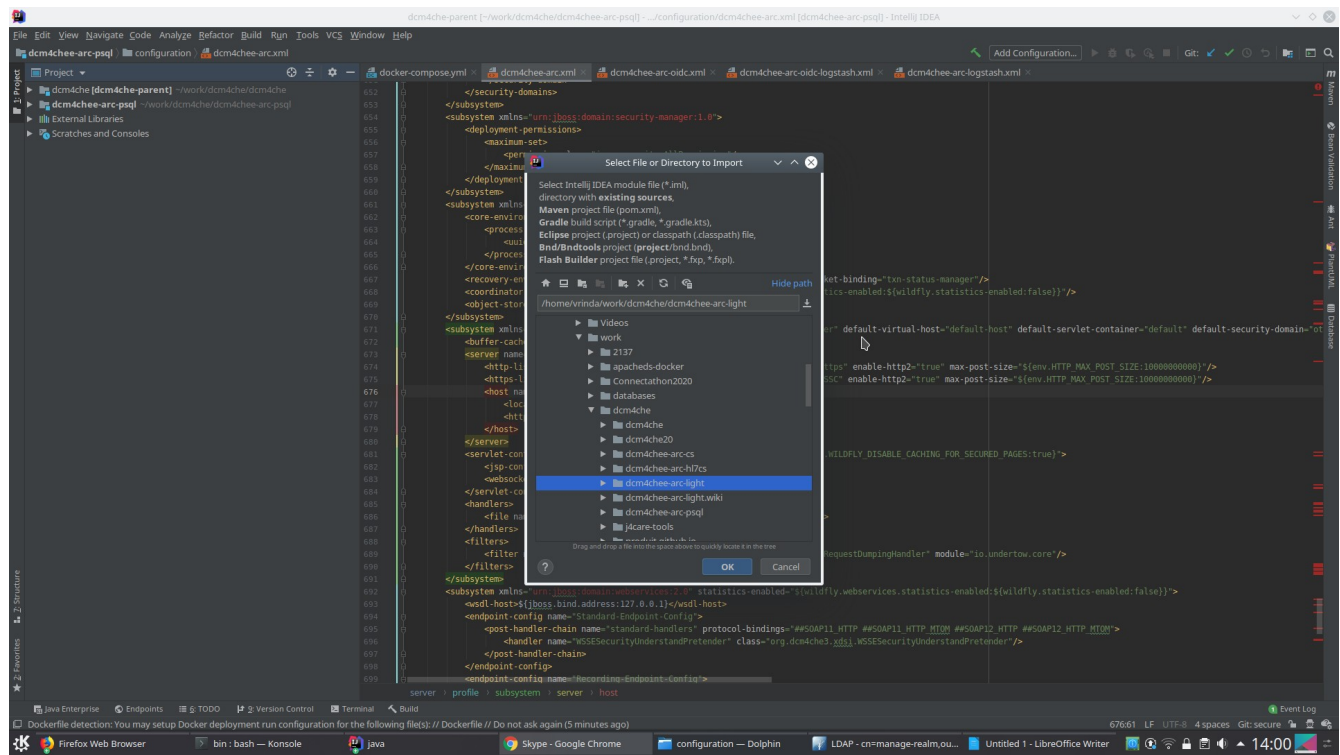
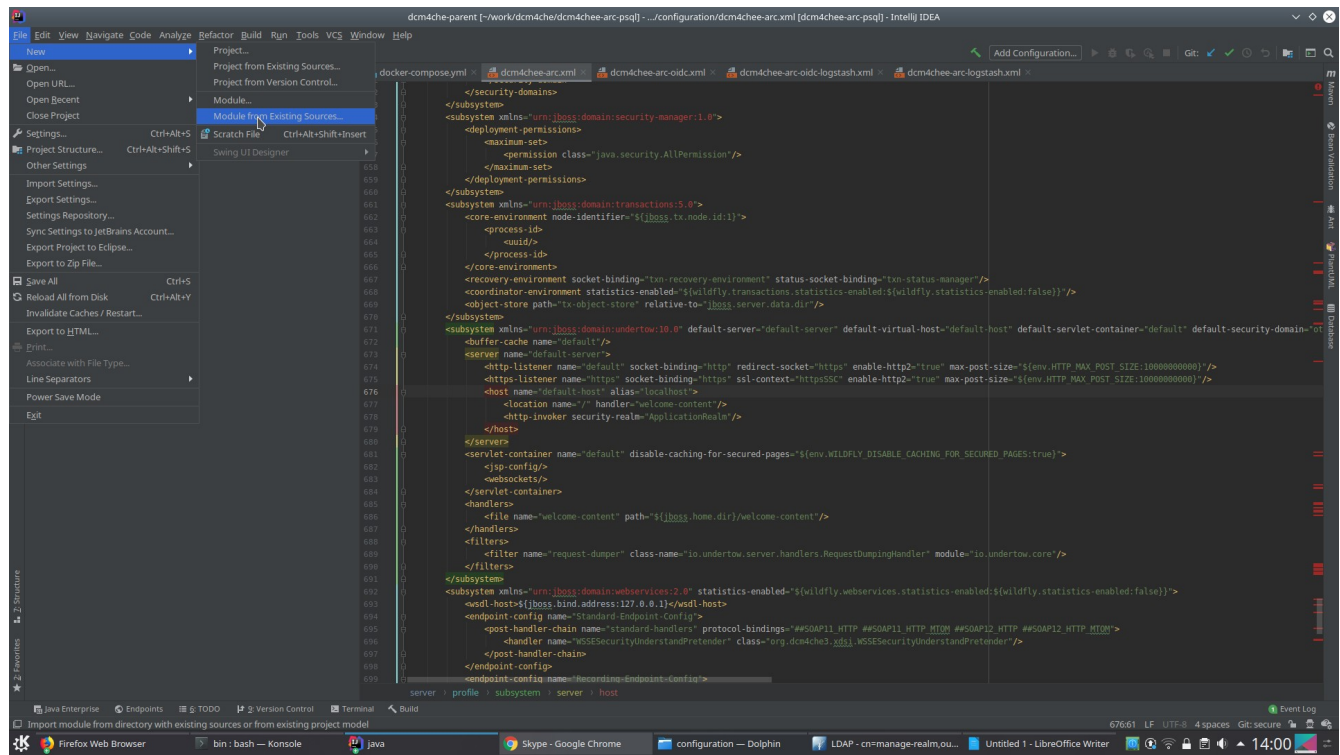


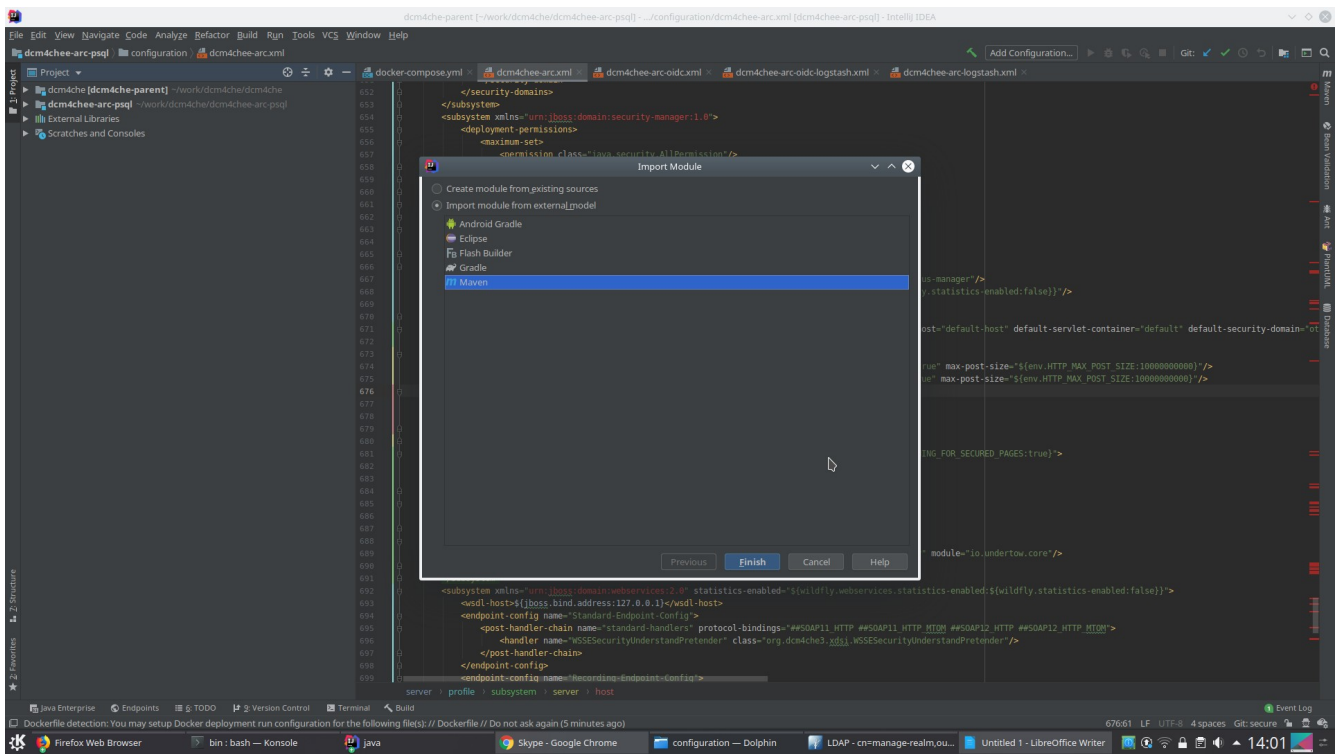


On *Finish*, you shall see this in left menu

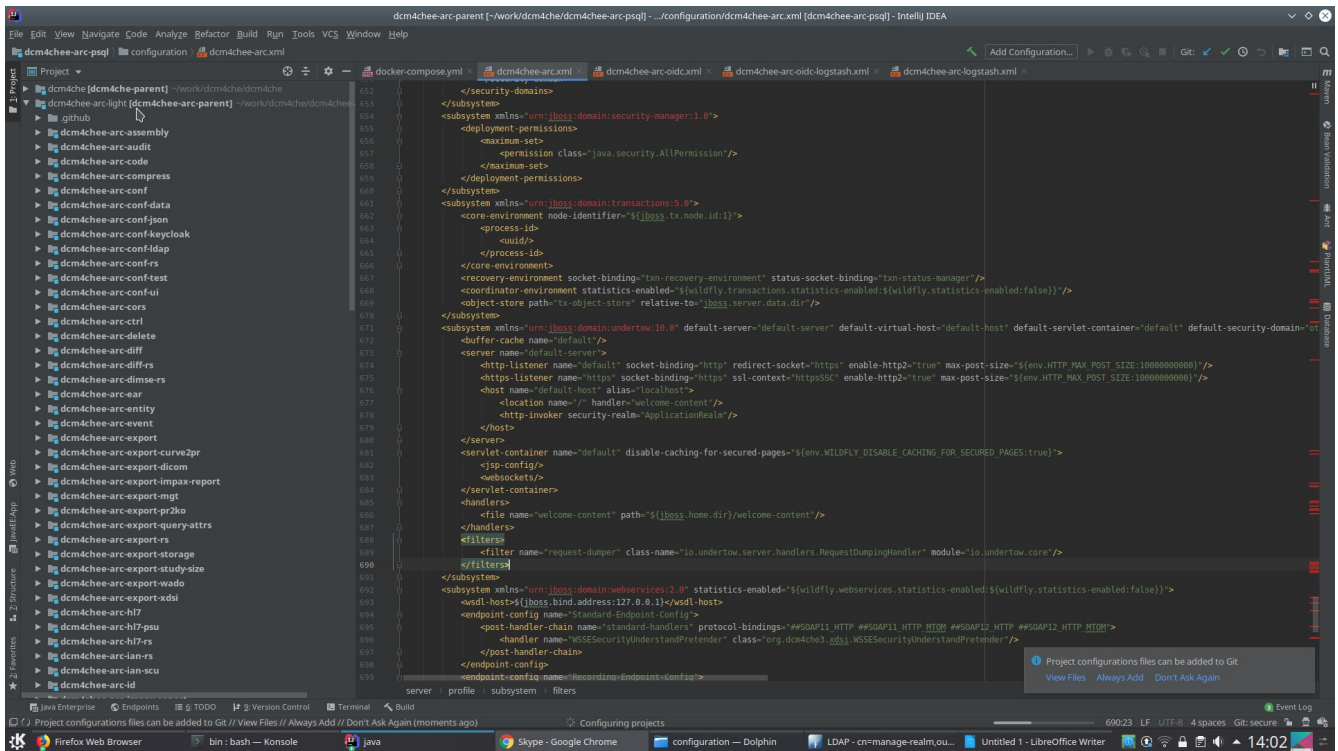


Add dcm4chee-arc-light as Module from Existing Sources



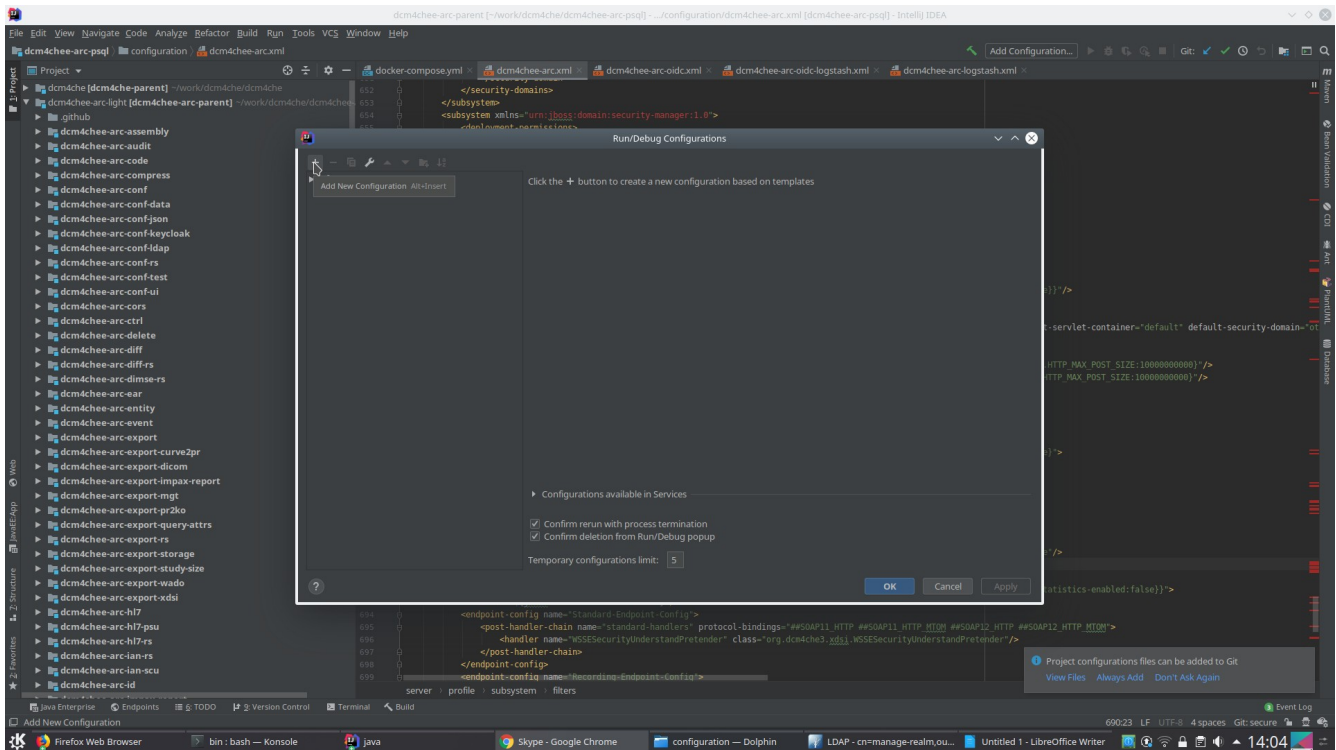
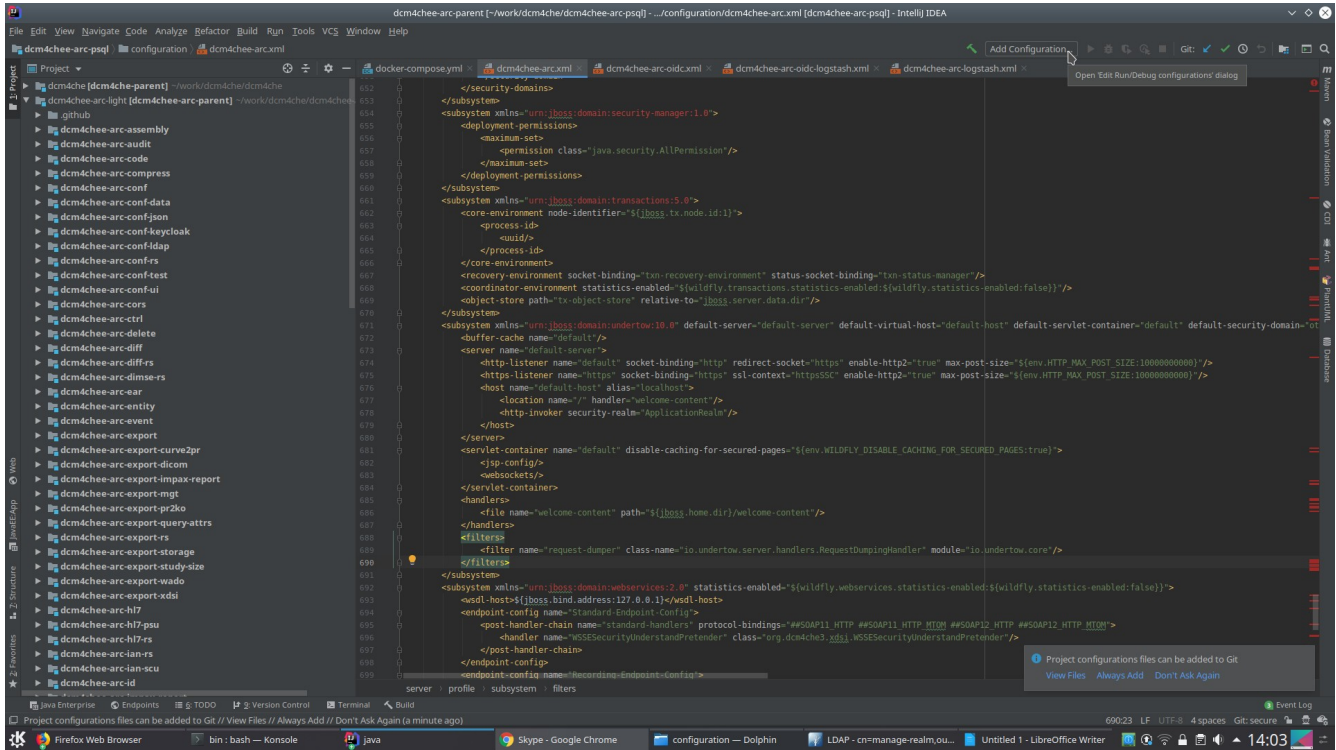


On *Finish*, you shall see this in left menu



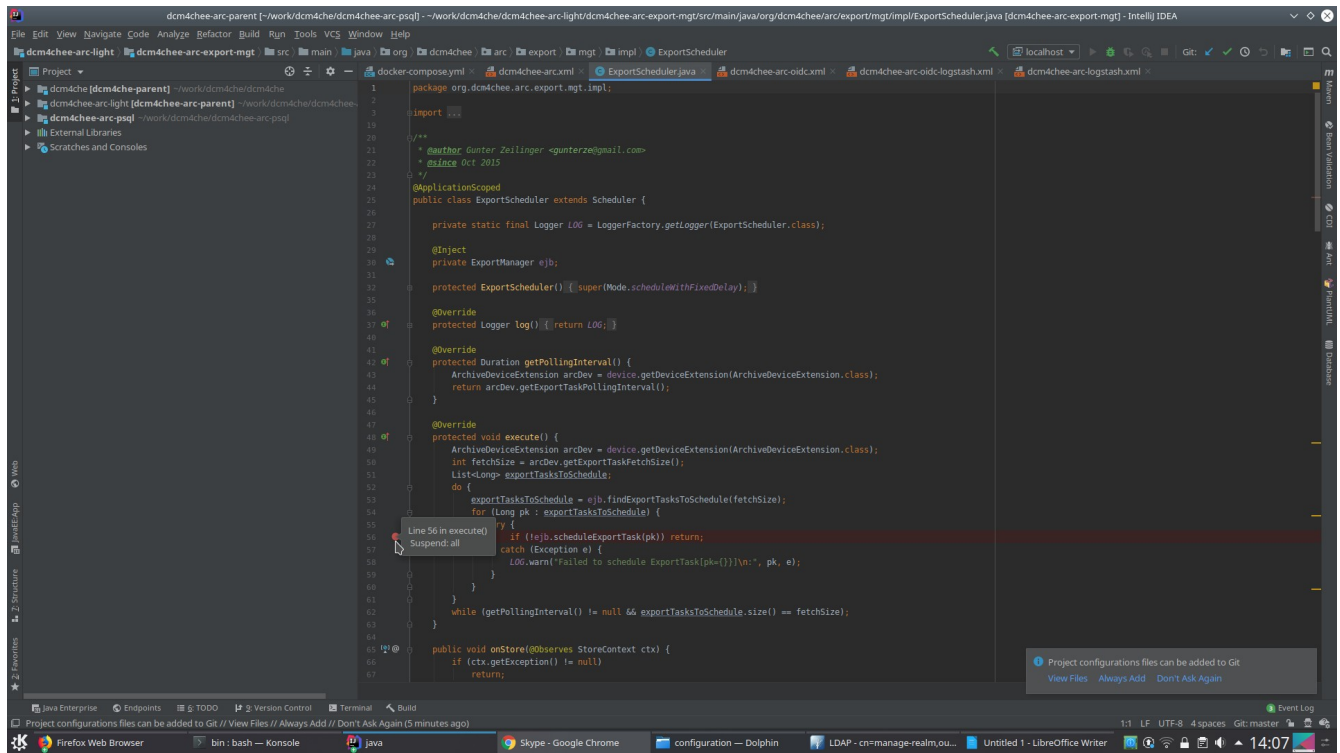
Add Remote Configuration for debugging

On top right you select Add Configuration



Add debug points

- ExportScheduler : Line 56 (Just click once)



```
package org.dcm4chee.arc.export.mgt.impl;

import java.util.*;

/**
 * @author Gunter Zeilinger <gunterze@gmail.com>
 * @since Oct 2015
 */
@ApplicationScoped
public class ExportScheduler extends Scheduler {

    private static final Logger LOG = LoggerFactory.getLogger(ExportScheduler.class);

    @Inject
    private ExportManager ejb;

    protected ExportScheduler() { super(Mode.SCHEDULE_ONCE); }

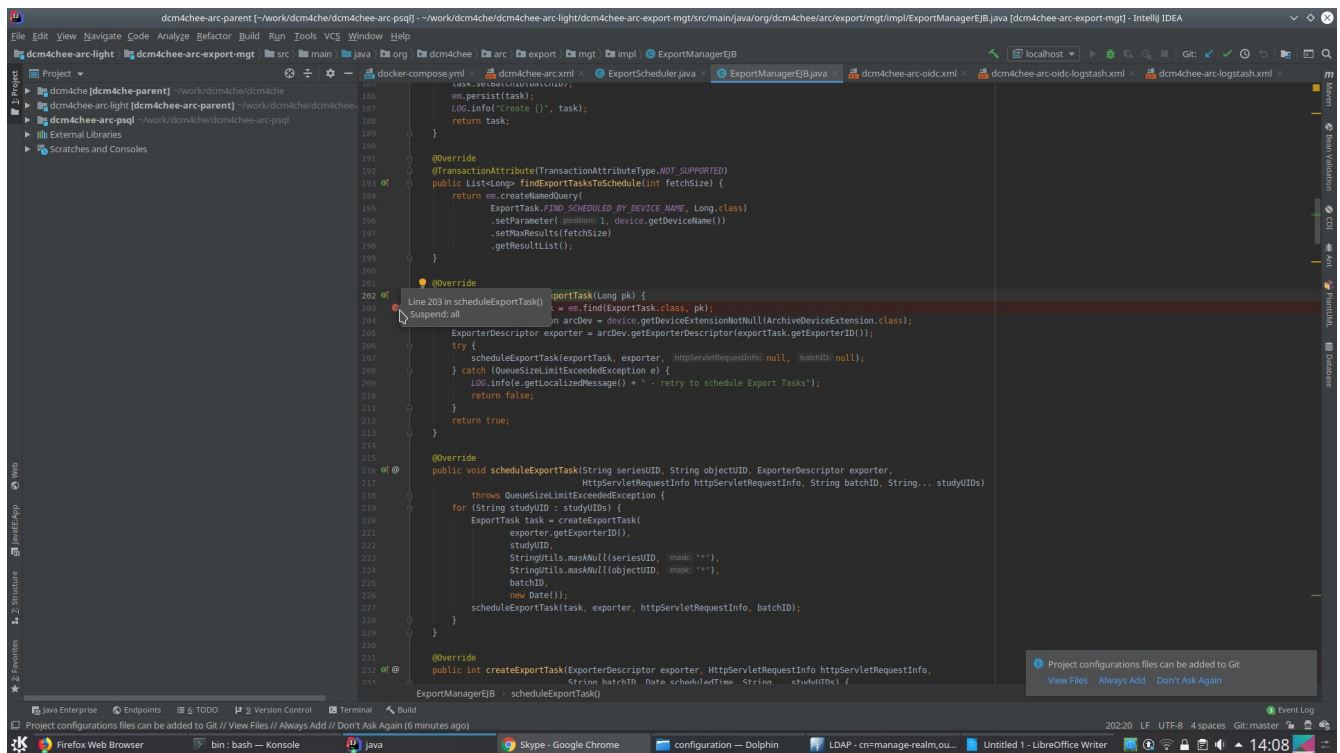
    @Override
    protected Logger log() { return LOG; }

    @Override
    protected Duration getPollingInterval() {
        ArchiveDeviceExtension arcDev = device.getDeviceExtension(ArchiveDeviceExtension.class);
        return arcDev.getExportTaskPollingInterval();
    }

    @Override
    protected void execute() {
        ArchiveDeviceExtension arcDev = device.getDeviceExtension(ArchiveDeviceExtension.class);
        int fetchSize = arcDev.getExportTaskFetchSize();
        List<Long> exportTasksToSchedule =
            do {
                exportTasksToSchedule = ejb.findExportTasksToSchedule(fetchSize);
                for (Long pk : exportTasksToSchedule) {
                    try {
                        if (ejb.scheduleExportTask(pk)) return;
                    } catch (Exception e) {
                        LOG.warn("Failed to schedule ExportTask[{}]", pk, e);
                    }
                }
            } while (getPollingInterval() != null && exportTasksToSchedule.size() == fetchSize);
    }

    public void onStore(@Observes StoreContext ctx) {
        if (ctx.getException() != null)
            return;
    }
}
```

- ExportManagerEJB : Line 203



```
...
    }
}

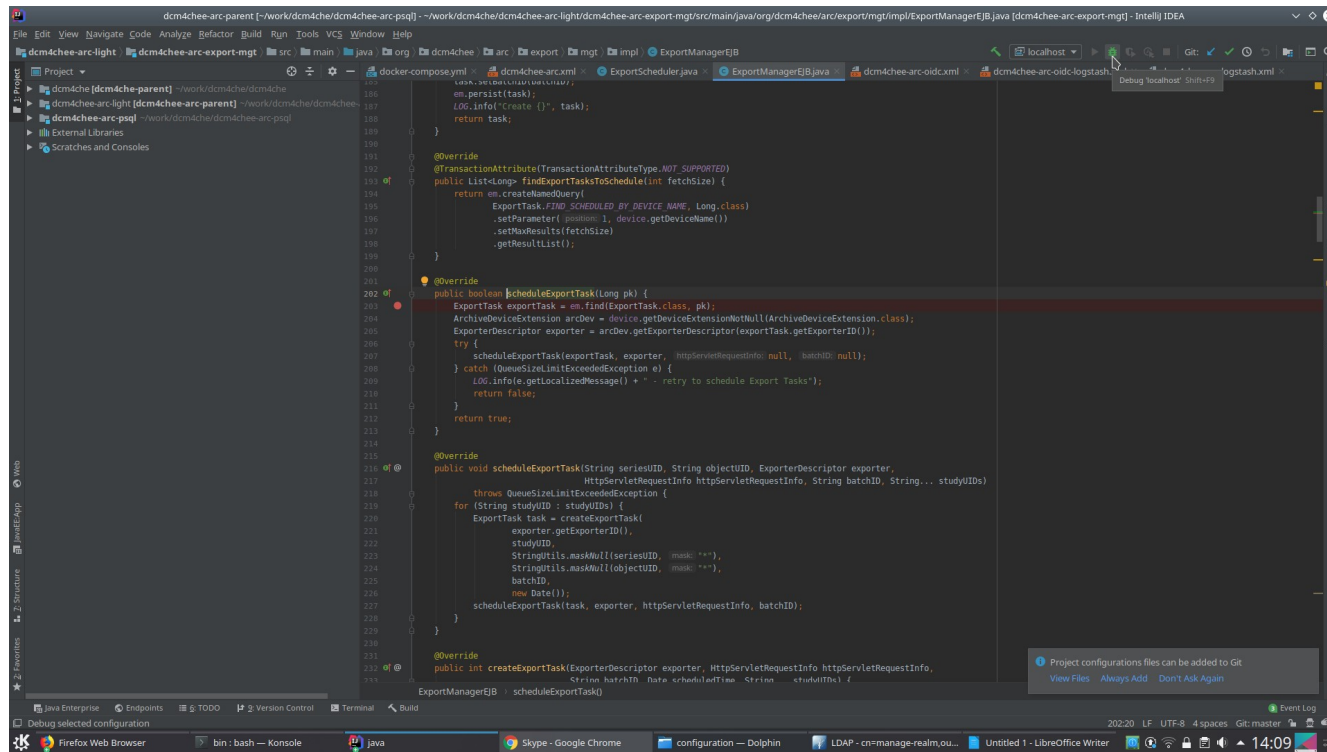
@Override
@Transactional(TransactionalAttributeType.NOT_SUPPORTED)
public List<Long> findExportTasksToSchedule(int fetchSize) {
    return em.createQuery("select pk from ExportTask pk where pk.seriesUID = :seriesUID and pk.objectUID = :objectUID",
        ExportTask.class)
        .setParameter("seriesUID", device.getDeviceName())
        .setMaxResults(fetchSize)
        .getResultList();
}

@Override
public boolean scheduleExportTask(Long pk) {
    ExportTask exportTask = em.find(ExportTask.class, pk);
    ArchiveDeviceExtension arcDev = device.getDeviceExtensionNotNull(ArchiveDeviceExtension.class);
    ExporterDescriptor exporter = arcDev.getExporterDescriptor(exportTask.getExporterID());
    try {
        scheduleExportTask(exportTask, exporter, HttpServletResponseInfo.NULL, batchID);
    } catch (QueueSizeLimitExceededException e) {
        LOG.info("getLocalizedName() + " - retry to schedule Export Task");
        return false;
    }
    return true;
}

@Override
public void scheduleExportTask(String seriesUID, String objectUID, ExporterDescriptor exporter,
    HttpServletResponseInfo httpServletRequestInfo, String batchID, String... studyUIDs)
    throws QueueSizeLimitExceededException {
    for (String studyUID : studyUIDs) {
        ExportTask task = createExportTask(
            exporter.getExporterID(),
            studyUID,
            StringUtils.maskNull(seriesUID, "mask: ***"),
            StringUtils.maskNull(objectUID, "mask: ***"),
            batchID,
            new Date());
        scheduleExportTask(task, exporter, httpServletRequestInfo, batchID);
    }
}

@Override
public int createExportTask(ExporterDescriptor exporter, HttpServletResponseInfo httpServletRequestInfo,
    String batchID, Date scheduledTime, String... studyUIDs) {
    ExportManagerEJB scheduleExportTask()
}
```

Start debug mode : On top right, you find green icon with Debug



You should see that debug mode is started

