These are some tips of how to make run the new mw 2.0 in eclipse.

The new version includes very important fixing bugs in the UI, some less visible at data representation and serialization level but also important, and some others in Context and LDDI components. For further details click [here](http://forge.universaal.org/mediawiki/index.php?title=support:RD_Release_History).

As you may know apart from the previous listed features there is another big change, which is that it has been developed with Apache Karaf OSGi implemenation. So, some of the features provided by MW2.0 are only available with Apache Karaf, while other ones are independent from the target platform. It means that you don’t need Karaf features and we can continue working as always (although if you want to use it, you can ;-)… just ask me how to create a feature and start using karaf distro)

So after this introduction we can go in with it….

**WHAT SECTIONS THIS README CONTAINS:**

1. Run your service in eclipse with the new mw version (without Karaf)
   1. What you need to include in your launch file
   2. Configuring mw2.0 and needed managers
   3. Configuring Konnex
2. Important tip

**--------------------------------------------------**

1. **RUN YOUR SERVICE IN ECLIPSE WITH THE NEW MW VERSION (without karaf)**
   1. **What you need to include in your launch file**

You need to go to the Run Configurations. Then will appear a dialog asking you about resolving composites, just say **NO**, otherwise it will take long and Eclipse will get crazy (at least in my case):



After it the Run Configurations screen should be opened, then go to

* **Tag Pax Runner** (not uAAL Runner). In the “Profiles” section, keep checked the CONFIG option. Add the following provisions in different levels and be sure that (1) **mw**, (2) **ontologies**, (3) **UI**, and (4) **your** **service**:
  + Mw composite
    - scan-composite:mvn:org.universAAL.middleware/mw.composite/2.0.0/composite
  + WP2 ontologies composite
    - scan-composite:mvn:org.universAAL.ontology/ont.composite/2.0.0/composite
  + UI artifacts
    - mvn:org.ufacekit.osgi/swingx.osgi/0.9.2
    - mvn:org.universAAL.ui/ui.dm/2.0.0
    - mvn:org.universAAL.ui/ui.handler.gui.swing/2.0.0
    - wrap:mvn:org.universAAL.ui/ui.handler.gui.swing.bluesteelLAF/2.0.0@nostart
  + LDDI artifacts
    - mvn:org.openrdf.sesame/sesame-runtime-osgi/2.6.0
    - mvn:commons-lang/commons-lang/2.6
    - mvn:org.universAAL.ontology/ont.device/2.0.0
    - mvn:org.apache.felix/org.apache.felix.dependencymanager.lddi/3.0.0
    - mvn:org.apache.felix/org.apache.felix.devicemanager/0.9.0-SNAPSHOT
    - wrap:mvn:org.jdom/jdom/1.1.3
    - wrap:mvn:jaxen/jaxen/1.1.3
    - mvn:org.universAAL.lddi/lddi.knx.library/2.0.0
    - mvn:org.universAAL.lddi/lddi.knx.networkdriver/2.0.0
    - mvn:org.universAAL.lddi/lddi.knx.devicemanager/2.0.0
    - mvn:org.universAAL.lddi/lddi.knx.exporter/2.0.0
* **Tab Bundles,** deselect all the bundles and Felix 3.0.2 at “Framework” section.
* **Tag (x)= Arguments**, at the “Working Directory” section, in the “Other” option change the rundir folder for the one of your service.
  1. **Configuring mw2.0 and needed managers**
* Unzip the file called “**your\_workspace**” (pass: uaal). Inside you have all needed to make mw2.0 run just needed some adjustments.
  + **rundir/**
    - **confadmin/**
      * **ctxt.che/**
        + **CHe.properties:** with a basic configuration, enough to start working. If you want to go in, click [here](http://forge.universaal.org/wiki/uaal_context:Context_Management_Building_Block#Configuration) for more information.
      * **mw.bus.model.osgi/**
        + **sodapop.key**
      * **ResourceServer/:**It is a server that retrieves multimedia resources from this common universAAL configuration directory via HTTP requests. You may use it or not in your service. For further information click [here](•%09http:/forge.universaal.org/wiki/uaal_ui:UI_Framework#The_Resource_Server_.28ui.resource.server.29).
      * **services/**
        + **lddi.hw.simulator.activityhub.properties**. This file is used by an event simulator.
        + **lddi.knx.devicemanager.properties**
        + **lddi.knx.networkdriver.properties**
        + **org.ops4j.pax.logging.properties**. This file is used to configure level of information that will be shown in the console prompt.
        + **mw.connectors.communication.jgroups.core.properties**. You don’t need to change anything.
        + **mw.connectors.discovery.slp.core.properties.** You don’t need to change anything.
        + **mw.managers.deploy.core.properties.** You don’t need to change anything.
        + **mw.modules.aalspace.core.properties.** You don’t need to change anything.
        + **org.ops4j.pax.logging.properties.** You don’t need to change anything.
        + **mw.managers.aalspace.core.properties.** Here you need to change the following properties:

**aalSpaceConfigurationPath. Here must be defined the absolute path where the etc/ folder is. *i.e C:/universAAL/dev/workspaces/YOUR\_WORKSPACE/rundir/etc/***

**aalSpaceSchemaURL. Same approach as in the previous point. *i.e. C:/universAAL/dev/workspaces/YOUR\_WORKSPACE/rundir/etc/***

* + - * + **org.universAAL.mw.data.representation.properties**

**org.universAAL.middleware.peer.member\_of=**urn:org.universAAL.aal\_space:**test\_environment**. What you need is to check that the aal\_space name is the same that you are using in your launch file.

* + - * **ui.dm/ You need to add you service URI to the main\_menu\_saied\_en.txt (for English language, for other languages you need to modify main\_menu\_saied\_XX.txt files) file to make it available in the main menu** **(i.e. /Agenda|http://www.anco.gr|http://ontology.universAAL.org/PersonalAgenda.owl#CalendarUIService|app/Personal\_Agenda.png) and be sure that the file is finished with a blank line. (**If your service has GUI, you need to add your service to the “ui.dm” config files. When you run your launcher a login screen will be shown. Type “saied” for username and whatever you want for password.)
        + **persistency/**
        + **main\_menu.txt**
        + **main\_menu\_saied.txt**
        + **main\_menu\_saied\_en.txt**
        + **messages.properties**
      * **ui.handler.gui.swing/** There are general images, icons, etc that you can use in the main menu or in your own service.
        + **icons/**
        + **renderer.properties**
      * **YOUR\_SERVICE\_CONF/ It must be renamed with your service name (the same as your service project).**
    - **etc/** MW2.0 configuration files. You don’t need to change anything.
      * **HOME.space.** You don’t need to change anything.
      * **aalspace.xsd.** You don’t need to change anything.
    - **YOUR\_SERVICE\_RUNNER/** 
      * **Runner/**
        + **Felix/**

**System.properties.** If you want to use the ResourceServer artifact, be sure that the configured port is the 8080 (org.osgi.service.http.port=8080)

* 1. **Configuring Konnex**

**For Konnex network.** It requires as input a file with the extension “.knxproj” which represents the real KNX installation available at the end-user home. This file can only be generated with the last version of the ETS software, namely ETS4 professional edition. The cost of this software is 960 € (plus 15 € as an additional service fee). The problem is that uAAL partners mostly have an old version of the ETS software (ETS3 professional) where the “Export to .knxproj file” is not supported. In that case, there are only options in order to be able to test this service:

* 1. Update our ETS software from version 3 to 4. The cost of this upgrade license is 310 €.
  2. Talk with a colleague with the required version of the ETS software to do the expected format conversion. The process should be as follows: export our KNX project in any format available in the version 3, share this file with our colleague who has to load our project in the version 4 of the software, export this project to a file with the “.knxproj” extension and back to us again.

KNX installation available in your Living Lab MUST come with a KNX/IP gateway module (interface between KNX and IP networks). This is mandatory. These hardware devices are quite expensive (600 – 1000 €, more or less) so it is probably that are not available in some laboratories.

After getting .knxproj:

* 1. Copy that file into

    \rundir\configurations\services\data

* 1. Go to

   \rundir\configurations\services

and modify the ***lddi.knx.devicemanager.properties*** file by changing the knxConfigFile property:

***knxConfigFile = configurations/services/data/YOUR\_KNX\_PROJ.knxproj***

* 1. In the same folder

\rundir\configurations\services

open ***lddi.knx.networkdriver.properties*** file and modify:

* + 1. IP of the computer running knxnetworkdriver

***myIP=192.168.233.251***

* + 1. IP of the KNX/IP network

***knxGatewayIp=192.168.230.1***

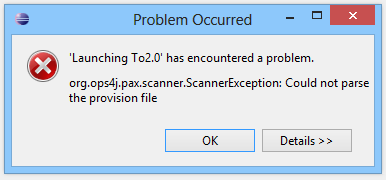
1. **IMPORTANT TIP**

* A small tip in case your run config takes ~very~ long to start up: one of the servers that is somewhere referenced in a maven file is down (scm.ops4j.org). You can considerably increase the starting procedure by adding the following line to your hosts-file under C:\Windows\System32\drivers\etc\hosts. You may need to run a Notepad app with admin privileges for being able to save the modifications à127.0.0.1 scm.ops4j.org
* If you get the following error when trying to run your new configuration (launch file). Don't worry, it happens sometimes when the rundir cache crashes so you can fix it :).
  + Close Eclipse
  + Go to your workspace (i.e. C:\universAAL\dev\workspaces\prototype\rundir\help\_when\_outdoor\runner)
  + Remove the following folders:

/bundles

/cache

* + Open Eclipse again
  + Open Run Configurations again
  + Try to run your configuration file again. If you get the same error, just click on **OK** button and try again. You might need several attempts but believe me...it works ;).

[](http://forge.universaal.org/wiki/File:Problem_ocurred.png)