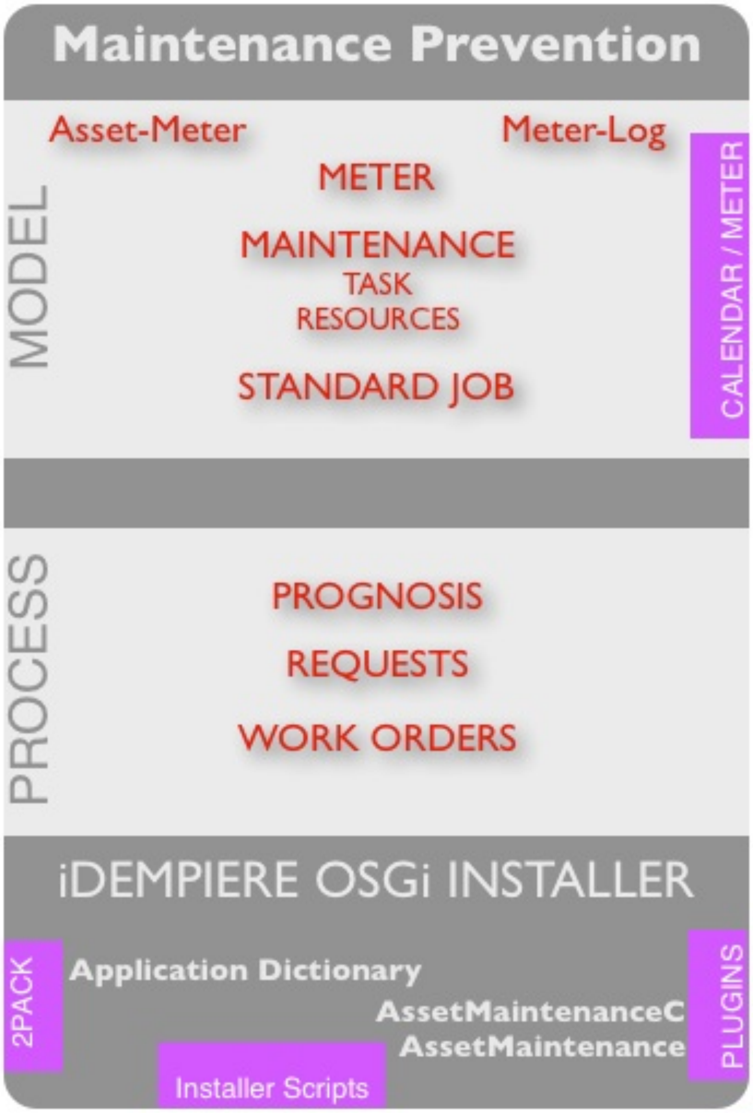
Redhuan D. Oon (red1)

A GUIDE ON HOW TO INSTALL, IMPLEMENT, INNOVATE



as an OSGi feature add-on for

IDEMPIERE

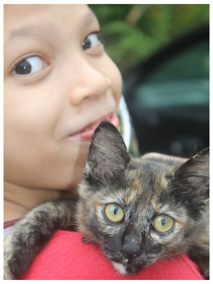
Open Source ERP on OSGi Technology

Version 1.0

New Calendar 2013

Almost everyone survived the Mayan Doomsday prophecy but not my MacBook which promptly refused to boot on the very eve of 2012. It suffered through 'the following year' of anguish where even the local Apple support centre refused to give any idea other than reformat the whole hard-disk. However, since I am an excellent hacker of the highest calibre, I managed to *single user* boot and transfer out files onto an attached USB disk. I have to thank a small friend called Google who understood what I asked.

With some days lost during my hard-disk recovery, and old photos still in the corrupted disk, I now begin finishing off this present task of a good old tutorial and I have to pluck out from my camera's sim-card to show you the idyllic lifestyle I operate from in a farm home on the outskirts of Kuala Lumpur. This lifestyle may disrupt again if our plans for an iDempiere World Conference in Germany happens this April.



Nasem, my third child, constantly hugs his very close pal, Boom-boom, in the garden of our two acre farm home.

A monkey turns to flee when I threw a fire-cracker during one of the horde's regular raids.

Occasionally I miss civilization and will give a presentation to curious humans in the city.

Further up the shoulders of giants

My last work, the iDempiere Migration Guide for Openbravo POS Integration is a beginner's break-through in learning how to convert a simple java project into an OSGi feature for deploying onto the iDempiere application. Now we shall proceed even more when we learn how to use further extensions such as Callouts. We also see the use of dependent plugin approach where heavy code is lifted off into a separate base plugin. There is also the alternative plugin hot install without stopping the iDempiere app.

I have also rearranged the menu structure, refactoring some code while making it compatible with the latest source particularly ZK Ajax 6.5.1, and simplify the many field labels and while testing it alot manually, got to delve deep enough to write this tutorial under the rhyming main sections of Install, Implement and Innovate.

Install is about how to setup or run the script to install automatically without download and configuring anything. **Implement** is about how to use the Maintenance application which is no different from what was written earlier. But I did give a once over of the original work of OFB Consulting and noted the better introduction of the Meter Log as a separate window for the workshop floor operations use. **Innovate** is the benefit of many long hours I spent combing the code as an OSGi plugin framework within iDempiere. I will show how to actually modify the field labels properly, debugging from Eclipse enroute, doing a Pack Out for distributing as a new plugin.

From here, this gives more added know-how to attempt our next and bigger plugin - the Libero HR Payroll module greatly enhanced by DoubleClick of Venezuela hopefully concluded before April Fool's Day.

- Monday, 7th January, 2013, Kuang farm, Selangor, Malaysia.

This work is sponsored by

BANGLADESH

sysnova

> *'For them it's a joyous event, not an apocalyptic event. What is coming is the end of a calendar and the beginning of a new one. This is the 5th time it happened.'*
>
> *Leonzo Barreno,*
>
> *Journalism professor who happens to be a Mayan descendant.*

DISCLAIMER:

IT IS THE NEW MAYAN CALENDAR. THE OLD PROPRIETARY AND PSEUDO FOSS SOFTWARE ARE DEAD AND THE PRESENT TRUE OPEN CODE KEEPS FORKING AND IS ALIVE. BEEN OPEN SOURCE ALLOWS ANYONE TO MODIFY AND PUBLISH THE IMPROVED CODE AND DOCUMENTATION. THIS GUIDE SERVES TO ASSIST FURTHER AS CODE ALONE IS ALWAYS CRYPTIC IF THERE IS NO COMMENTARY TO HELP. FOLLOWING THIS GUIDE IS ESSENTIAL BEFORE REQUESTING FOR ANY TECHNICAL ASSISTANCE.

CREDITS:

OFB CONSULTING of CHILE for the original Asset Maintenance contribution as open source.

SYSNOVA of BANGLADESH for sponsoring its improvement and as an iDempiere plugin.

TREKGLOBAL of USA for sponsoring iDempiere progress.

JOERG VIOLA of GERMANY who first improved the OSGi implementation.

LOW HENG SIN continuing iDempiere and his expert coaching on how to use it.

CARLOS RUIZ for his *preventive maintenance* of source-code and release of iDempiere v1.0.a.

DOMINIK ZAJAC for maintaining the Jenkins and JIRA servers of the project.

COMMUNITY that is giving a constant flow of functional reviews, bug reports and fixes.

COMPIERE, the main parent project that ADempiere forked from and then iDempiere that retains most of the code.

FOSS Projects can also experience a doomsday effect. Here we review our own Adempiere Mayan Calendar.

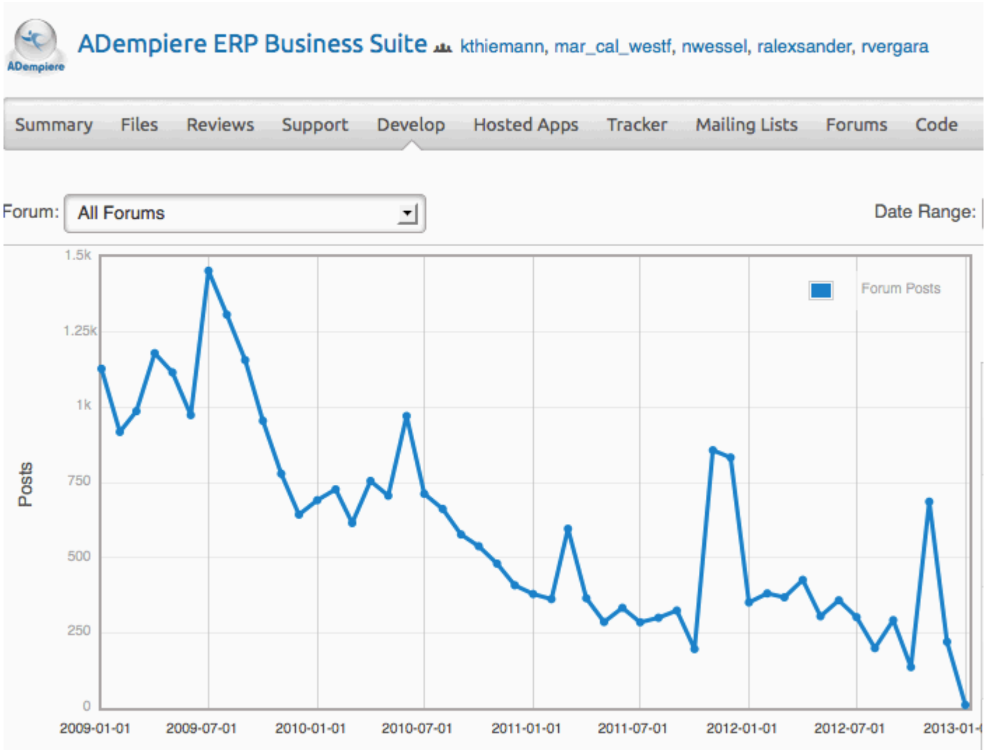
First discussion on what to do about the OSGi project:

http://sourceforge.net/projects/adempiere/forums/forum/610546/topic/4008276

Last discussion on what to do about the OSGi project:

http://sourceforge.net/projects/adempiere/forums/forum/610547/topic/6552457

Adempiere project forum activity graph to date.



Forum: All Forums

Date Range:

Source: http://sourceforge.net/projects/adempiere/stats/forums?forum=&dates=2009-01-01+to+2013-01-08

But we can have renewal instead of death. FOSS allows for forks and guarantees community freedom. The change of words or double-speak in the two forum threads above prompt me to quote from a famous political satire written in the last century:

> "Man is the only creature that consumes without producing. He does not give milk, he does not lay eggs, he is too weak to pull the plough, he cannot run fast enough to catch rabbits. Yet he is lord of all the animals. He sets them to work, he gives back to them the bare minimum that will prevent them from starving, and the rest he keeps for himself."
>
> - George Orwell, *Animal Farm, Chapter 1*

> "They had come to a time when no one dared speak his mind, when fierce, growling dogs roamed everywhere, and when you had to watch your comrades torn to pieces after confessing to shocking crimes."
>
> - George Orwell, *Animal Farm, Chapter 7*

TABLE OF CONTENTS

Cheat Sheet

The introduction of a cheat sheet and quick-take reference here is for advanced users or otherwise have read and gone through this manual. If you are reading this in e-book format, then just put your mouse over the blue colored references and they have hyperlinks where you can click to get to the download or referred pages on the Internet. If any of the links are broken please report them to the author.

DOWNLOAD iDEMPIERE:

Unix: V1.0.a

Windows version with Upgrade Assistant for last migration scripts

UPGRADE iDEMPIERE SERVER

Unix: Server-Update.sh and apply *last migration scripts manually.

Windows: Server-Update.bat

INSTALL *MP* MODULE:

Unix: http://sourceforge.net/projects/red1/files/p2/MaintenanceInstaller.sh/download
Windows: http://sourceforge.net/projects/red1/files/p2/MaintenanceInstaller.bat/download
2Pack: http://sourceforge.net/projects/red1/files/p2/Maintenance/Maintenance2Pack.zip/download
Sample Data: MaintenanceSampleData_postgresql.sql

LAST CHANGES USED:

LAST MIGRATION SCRIPT APPLIED IN IDEMPIERE CORE: 201212271855_TICKET-1001758.sql

LAST CODE CHANGE TO ADEMPIERE CORE: http://jira.idempiere.com/browse/IDEMPIERE-542

*DB AND ENVIRONMENT USED:

POSTGRESQL 9.2, MAC OSX LION, JAVA 1.6, ECLIPSE INDIGO.

PUBLIC FORUM:

Participation or feedback on this particular work can be directed at this forum thread:

http://red1.org/adempiere/viewtopic.php?f=29&t=1727

AUTHOR:

Main landing page: http://www.red1.org

Email: red1@red1.org

(Send private emails only if you intend to get paid support)

Introduction

First we will present in general what is this module about before diving into its 3-topic specifics in detail.

Module Name

Though this module was originally named as Asset Maintenance, its database tables' prefix is MP which seems to stand for Maintenance Prevention, following another modern name of Preventive Maintenance. I chose the middle path, to be consistent with the prefix of MP. I will be using the abbreviation of MP to stand for the name.

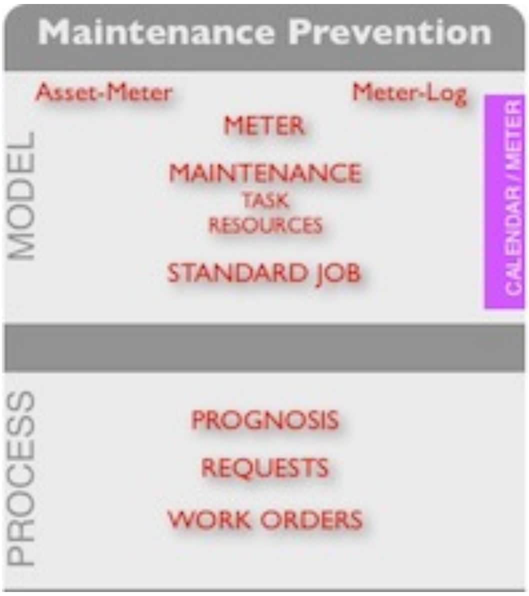
What is Maintenance Prevention?

Equipment or tools such as vehicles and machines are in constant threat of breakdown and arising costs due to loss of use unless properly maintained at regular recommended intervals to minimize sudden breakdowns. MP is intended to keep track of the schedules of maintenance as well as the ad-hoc requirement where a request for repair or checkup is needed.

Functions of the Module

MP has functions that covers the setup of the **Maintenance** Job based on **Standard Job** templates which allow the process of generating Work Orders.

The setup part is shown here in this diagram as the *Model*. The processes is aptly shown here as *Process*. Within the model, after a Standard Job is defined for an Asset, it is further defined in the Maintenance window as either a **Meter** or **Calendar** schedule type. **Meter Log** is then the daily routine of collecting readings from the **Asset Meter**. Periodically the Prognosis or Forecast is conducted to issue the right **Work Orders** (OT).



There can be ad-hoc requests for Work Order issuance outside the regular schedule. All Work Orders are derived from the Standard Job and Maintenance templates with details of **Tasks** and **Resources** needed for each task. Controls are provided for the user to mark each finished task and then the whole work order can be closed.

Thus this module gives operational visibility to the users from both the workshop floor and management office to view and process maintenance of assets as handled by the crew within the company.

In the Maintenance window, the Meter or Calendar panel keeps track of the last maintenance run to calculate its next run. It can further be refined with a desired range in between such runs.

Install

If you do not have any iDempiere installed prior to this, you can do so by fetching the base release V1.0.a and set it up manually according to this link -

http://www.globalqss.com/wiki/index.php/IDempiere/Installing_from_Installers

For the windows platform of course you can just use the wizard Windows Installer. Then run the Upgrade Assistant inside it to get the migration scripts applied. I have uploaded the to-date scripts which ends in

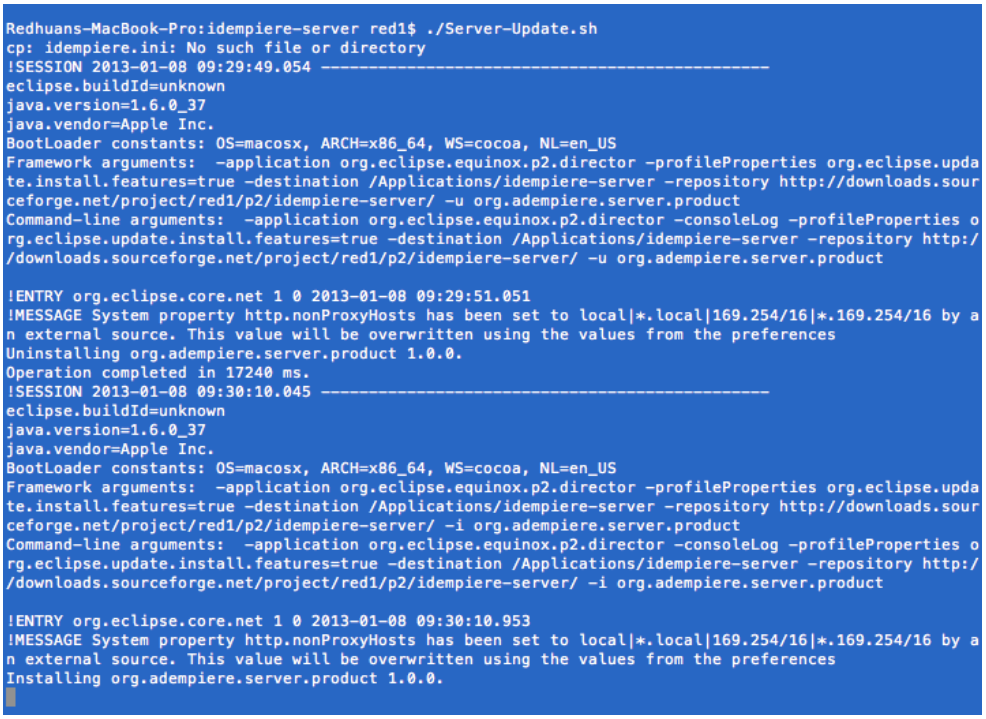
201212271855_TICKET-1001758.sql . For Unix environments, you have to apply the scripts manually.

If you come from the last episode of the iDempiere migration guide, you should be aware of the installer script to fetch the upgrade binaries from a remote P2 site. You can now easily upgrade to the right version for this tutorial by running these installer scripts inside your idempiere-server folder:

Server-Update.sh for Unix/Linux/Mac platforms and

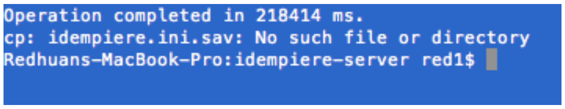
Server-Update.bat for Windows.

Source: http://sourceforge.net/projects/red1/files/p2/



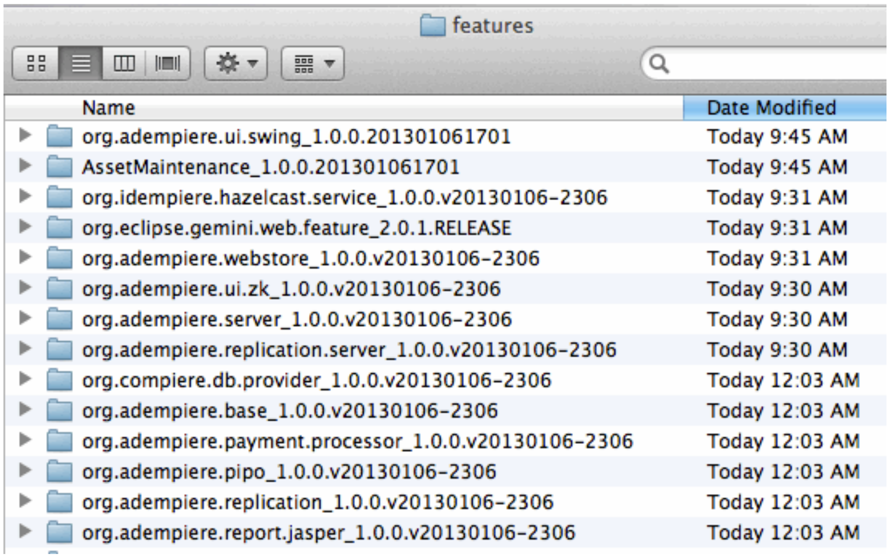
```
Redhuans-MacBook-Pro:idempiere-server red1$ ./Server-Update.sh
cp: idempiere.ini: No such file or directory
!SESSION 2013-01-08 09:29:49.054 ----------------------------------------------
eclipse.buildId=unknown
java.version=1.6.0_37
java.vendor=Apple Inc.
BootLoader constants: OS=macosx, ARCH=x86_64, WS=cocoa, NL=en_US
Framework arguments:  -application org.eclipse.equinox.p2.director -profileProperties org.eclipse.upda
te.install.features=true -destination /Applications/idempiere-server -repository http://downloads.sour
ceforge.net/project/red1/p2/idempiere-server/ -u org.adempiere.server.product
Command-line arguments:  -application org.eclipse.equinox.p2.director -consoleLog -profileProperties o
rg.eclipse.update.install.features=true -destination /Applications/idempiere-server -repository http:/
/downloads.sourceforge.net/project/red1/p2/idempiere-server/ -u org.adempiere.server.product

!ENTRY org.eclipse.core.net 1 0 2013-01-08 09:29:51.051
!MESSAGE System property http.nonProxyHosts has been set to local|*.local|169.254/16|*.169.254/16 by a
n external source. This value will be overwritten using the values from the preferences
Uninstalling org.adempiere.server.product 1.0.0.
Operation completed in 17240 ms.
!SESSION 2013-01-08 09:30:10.045 ----------------------------------------------
eclipse.buildId=unknown
java.version=1.6.0_37
java.vendor=Apple Inc.
BootLoader constants: OS=macosx, ARCH=x86_64, WS=cocoa, NL=en_US
Framework arguments:  -application org.eclipse.equinox.p2.director -profileProperties org.eclipse.upda
te.install.features=true -destination /Applications/idempiere-server -repository http://downloads.sour
ceforge.net/project/red1/p2/idempiere-server/ -i org.adempiere.server.product
Command-line arguments:  -application org.eclipse.equinox.p2.director -consoleLog -profileProperties o
rg.eclipse.update.install.features=true -destination /Applications/idempiere-server -repository http:/
/downloads.sourceforge.net/project/red1/p2/idempiere-server/ -i org.adempiere.server.product

!ENTRY org.eclipse.core.net 1 0 2013-01-08 09:30:10.953
!MESSAGE System property http.nonProxyHosts has been set to local|*.local|169.254/16|*.169.254/16 by a
n external source. This value will be overwritten using the values from the preferences
Installing org.adempiere.server.product 1.0.0.
```

The last line in the screen-shot takes about 10 minutes over a 1Mb line. At the end of it this will appear:



```
Operation completed in 218414 ms.
cp: idempiere.ini.sav: No such file or directory
Redhuans-MacBook-Pro:idempiere-server red1$
```

When you go into your idempiere-server/features folder you will see the date time-stamp of some of the jars are updated.



Name	Date Modified
org.adempiere.ui.swing_1.0.0.201301061701	Today 9:45 AM
AssetMaintenance_1.0.0.201301061701	Today 9:45 AM
org.idempiere.hazelcast.service_1.0.0.v20130106-2306	Today 9:31 AM
org.eclipse.gemini.web.feature_2.0.1.RELEASE	Today 9:31 AM
org.adempiere.webstore_1.0.0.v20130106-2306	Today 9:31 AM
org.adempiere.ui.zk_1.0.0.v20130106-2306	Today 9:30 AM
org.adempiere.server_1.0.0.v20130106-2306	Today 9:30 AM
org.adempiere.replication.server_1.0.0.v20130106-2306	Today 9:30 AM
org.compiere.db.provider_1.0.0.v20130106-2306	Today 12:03 AM
org.adempiere.base_1.0.0.v20130106-2306	Today 12:03 AM
org.adempiere.payment.processor_1.0.0.v20130106-2306	Today 12:03 AM
org.adempiere.pipo_1.0.0.v20130106-2306	Today 12:03 AM
org.adempiere.replication_1.0.0.v20130106-2306	Today 12:03 AM
org.adempiere.report.jasper_1.0.0.v20130106-2306	Today 12:03 AM

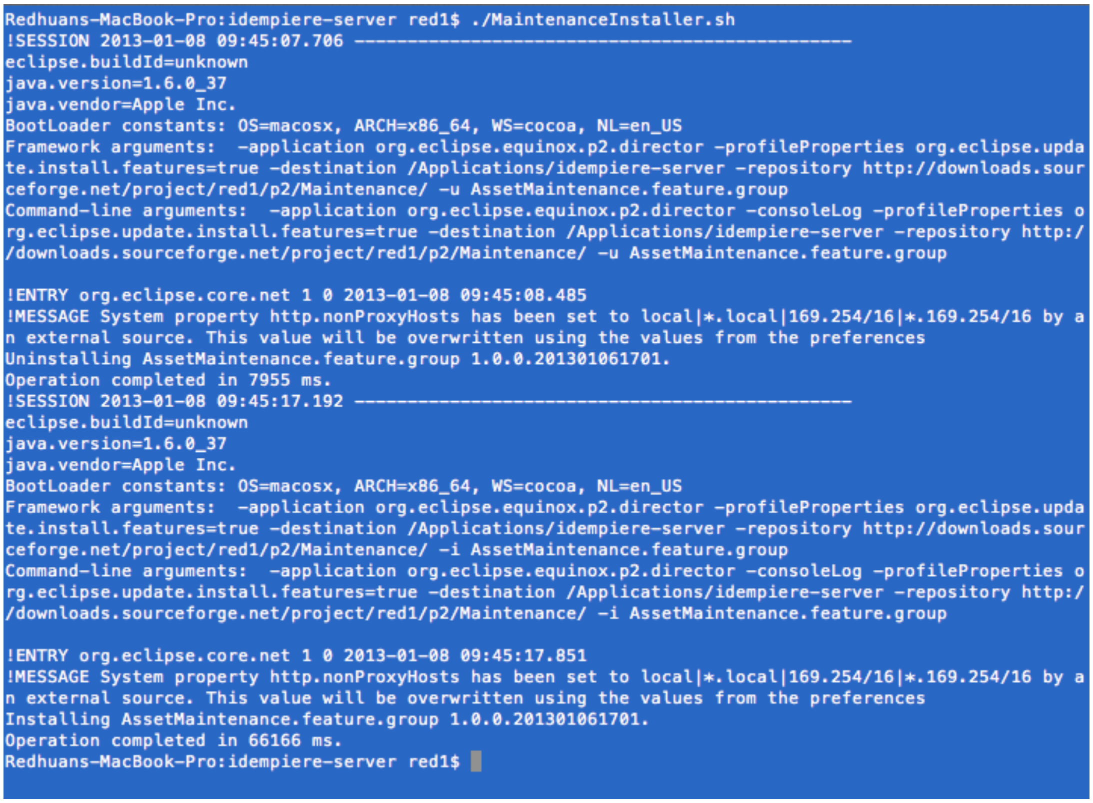
This update to latest idempiere jars are needed because there is a bug discovered during my review and Heng Sin has accepted it as a core change to AD_Tab_Panel which concerns the IColumnCallout to work. More of this is explained in the Innovate section. Just know that if you do not do this update, then your new module Callouts or any other future modules Callouts may not work.

The above screenshot also shows AssetMaintenance and ui.swing features been installed as we did so shortly after that. As noted, ui.swing is been packaged due to the original AssetMaintenance code using swing code instead of ZKAjax purely. In future, we may change that, but for now this works well and we leave it at that.

The P2 site is from my SourceForge.net repository so it is a controlled version site. Periodically I upload fresh binaries from Jenkins iDempiere Daily workspace, that I got to casually test them first. You may choose your own preferred P2 site but you need to modify the location within the script. More on this in the Innovate section. Using SourceForge.net as a download site is good as SourceForge.net has numerous mirror sites to download from all over the world.

Finally install the MP module:

Unix: http://sourceforge.net/projects/red1/files/p2/MaintenanceInstaller.sh/download
Windows: http://sourceforge.net/projects/red1/files/p2/MaintenanceInstaller.bat/download



```
Redhuans-MacBook-Pro:idempiere-server red1$ ./MaintenanceInstaller.sh
!SESSION 2013-01-08 09:45:07.706 -----------------------------------------------
eclipse.buildId=unknown
java.version=1.6.0_37
java.vendor=Apple Inc.
BootLoader constants: OS=macosx, ARCH=x86_64, WS=cocoa, NL=en_US
Framework arguments:  -application org.eclipse.equinox.p2.director -profileProperties org.eclipse.upda
te.install.features=true -destination /Applications/idempiere-server -repository http://downloads.sour
ceforge.net/project/red1/p2/Maintenance/ -u AssetMaintenance.feature.group
Command-line arguments:  -application org.eclipse.equinox.p2.director -consoleLog -profileProperties o
rg.eclipse.update.install.features=true -destination /Applications/idempiere-server -repository http:/
/downloads.sourceforge.net/project/red1/p2/Maintenance/ -u AssetMaintenance.feature.group

!ENTRY org.eclipse.core.net 1 0 2013-01-08 09:45:08.485
!MESSAGE System property http.nonProxyHosts has been set to local|*.local|169.254/16|*.169.254/16 by a
n external source. This value will be overwritten using the values from the preferences
Uninstalling AssetMaintenance.feature.group 1.0.0.201301061701.
Operation completed in 7955 ms.
!SESSION 2013-01-08 09:45:17.192 -----------------------------------------------
eclipse.buildId=unknown
java.version=1.6.0_37
java.vendor=Apple Inc.
BootLoader constants: OS=macosx, ARCH=x86_64, WS=cocoa, NL=en_US
Framework arguments:  -application org.eclipse.equinox.p2.director -profileProperties org.eclipse.upda
te.install.features=true -destination /Applications/idempiere-server -repository http://downloads.sour
ceforge.net/project/red1/p2/Maintenance/ -i AssetMaintenance.feature.group
Command-line arguments:  -application org.eclipse.equinox.p2.director -consoleLog -profileProperties o
rg.eclipse.update.install.features=true -destination /Applications/idempiere-server -repository http:/
/downloads.sourceforge.net/project/red1/p2/Maintenance/ -i AssetMaintenance.feature.group

!ENTRY org.eclipse.core.net 1 0 2013-01-08 09:45:17.851
!MESSAGE System property http.nonProxyHosts has been set to local|*.local|169.254/16|*.169.254/16 by a
n external source. This value will be overwritten using the values from the preferences
Installing AssetMaintenance.feature.group 1.0.0.201301061701.
Operation completed in 66166 ms.
Redhuans-MacBook-Pro:idempiere-server red1$ 
```

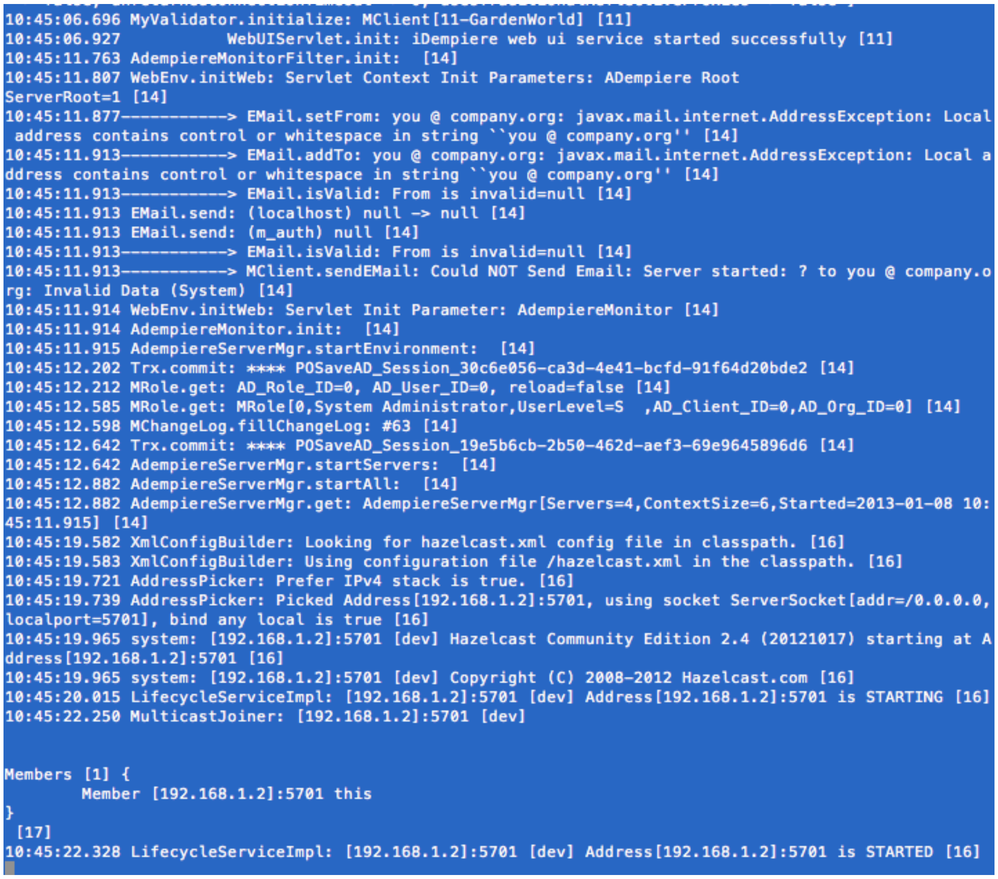
(Note: Whenever you update your idempiere-server, you need to update again your new module or do the lesser manual install call as shown for adempiere-client folder later).

Doing 2Pack Pack In

As shown in the iDempiereMigrationGuide.pdf there will be two ways you can do this. One is to do it manually through Pack In when login as System Admin. For that you can fetch the Maintenance2Pack.zip manually from the same Maintenance P2 site.

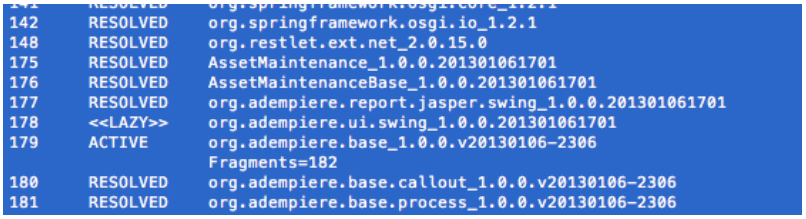
The other way will be the more exciting way of starting the base bundle in your OSGi console which already has the embedded 2Pack.zip without you needing to download that again. We will show the OSGi console way.

Run **idempiere-server.sh** from the idempiere-server folder. Below is what you will see in the end.



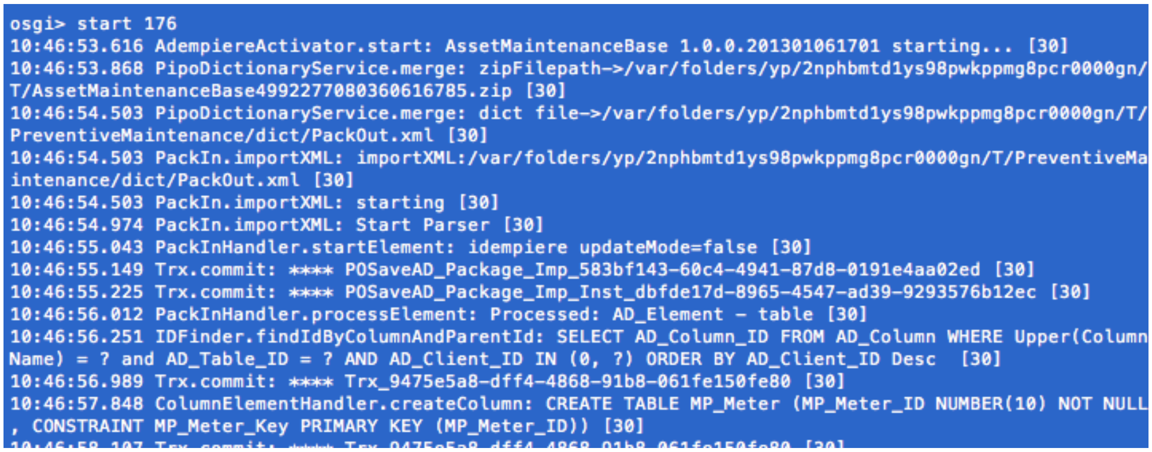
```
10:45:06.696 MyValidator.initialize: MClient[11-GardenWorld] [11]
10:45:06.927           WebUIServlet.init: iDempiere web ui service started successfully [11]
10:45:11.763 AdempiereMonitorFilter.init:  [14]
10:45:11.807 WebEnv.initWeb: Servlet Context Init Parameters: ADempiere Root
ServerRoot=1 [14]
10:45:11.877-----------> EMail.setFrom: you @ company.org: javax.mail.internet.AddressException: Local
 address contains control or whitespace in string ``you @ company.org'' [14]
10:45:11.913-----------> EMail.addTo: you @ company.org: javax.mail.internet.AddressException: Local a
ddress contains control or whitespace in string ``you @ company.org'' [14]
10:45:11.913-----------> EMail.isValid: From is invalid=null [14]
10:45:11.913 EMail.send: (localhost) null -> null [14]
10:45:11.913 EMail.send: (m_auth) null [14]
10:45:11.913-----------> EMail.isValid: From is invalid=null [14]
10:45:11.913-----------> MClient.sendEMail: Could NOT Send Email: Server started: ? to you @ company.o
rg: Invalid Data (System) [14]
10:45:11.914 WebEnv.initWeb: Servlet Init Parameter: AdempiereMonitor [14]
10:45:11.914 AdempiereMonitor.init:  [14]
10:45:11.915 AdempiereServerMgr.startEnvironment:  [14]
10:45:12.202 Trx.commit: **** POSaveAD_Session_30c6e056-ca3d-4e41-bcfd-91f64d20bde2 [14]
10:45:12.212 MRole.get: AD_Role_ID=0, AD_User_ID=0, reload=false [14]
10:45:12.585 MRole.get: MRole[0,System Administrator,UserLevel=S  ,AD_Client_ID=0,AD_Org_ID=0] [14]
10:45:12.598 MChangeLog.fillChangeLog: #63 [14]
10:45:12.642 Trx.commit: **** POSaveAD_Session_19e5b6cb-2b50-462d-aef3-69e9645896d6 [14]
10:45:12.642 AdempiereServerMgr.startServers:  [14]
10:45:12.882 AdempiereServerMgr.startAll:  [14]
10:45:12.882 AdempiereServerMgr.get: AdempiereServerMgr[Servers=4,ContextSize=6,Started=2013-01-08 10:
45:11.915] [14]
10:45:19.582 XmlConfigBuilder: Looking for hazelcast.xml config file in classpath. [16]
10:45:19.583 XmlConfigBuilder: Using configuration file /hazelcast.xml in the classpath. [16]
10:45:19.721 AddressPicker: Prefer IPv4 stack is true. [16]
10:45:19.739 AddressPicker: Picked Address[192.168.1.2]:5701, using socket ServerSocket[addr=/0.0.0.0,
localport=5701], bind any local is true [16]
10:45:19.965 system: [192.168.1.2]:5701 [dev] Hazelcast Community Edition 2.4 (20121017) starting at A
ddress[192.168.1.2]:5701 [16]
10:45:19.965 system: [192.168.1.2]:5701 [dev] Copyright (C) 2008-2012 Hazelcast.com [16]
10:45:20.015 LifecycleServiceImpl: [192.168.1.2]:5701 [dev] Address[192.168.1.2]:5701 is STARTING [16]
10:45:22.250 MulticastJoiner: [192.168.1.2]:5701 [dev]


Members [1] {
        Member [192.168.1.2]:5701 this
}
 [17]
10:45:22.328 LifecycleServiceImpl: [192.168.1.2]:5701 [dev] Address[192.168.1.2]:5701 is STARTED [16]
```

Then type 'ss' in the console. If ss cannot work you may need to modify the shell script by removing the time limit after the -console argument. You will see a long list of bundle IDs. Look for our new Maintenance jars. You will also note that swing.ui is added here and is in LAZY mode. This is because some Maintenance panel depends on swing and packaged it.



```
142     RESOLVED    org.springframework.osgi.io_1.2.1
148     RESOLVED    org.restlet.ext.net_2.0.15.0
175     RESOLVED    AssetMaintenance_1.0.0.201301061701
176     RESOLVED    AssetMaintenanceBase_1.0.0.201301061701
177     RESOLVED    org.adempiere.report.jasper.swing_1.0.0.201301061701
178     <<LAZY>>    org.adempiere.ui.swing_1.0.0.201301061701
179     ACTIVE      org.adempiere.base_1.0.0.v20130106-2306
                    Fragments=182
180     RESOLVED    org.adempiere.base.callout_1.0.0.v20130106-2306
181     RESOLVED    org.adempiere.base.process_1.0.0.v20130106-2306
```

Here the bundle IDs are **175** and **176** and they are resolved which means they have no missing dependencies and can now be used by the application. The 2Pack is embedded in that 176. So activate that with '**start 176**'. Below is what will happen. Note that in your console the ID may be different and you have to note its displayed ID number.



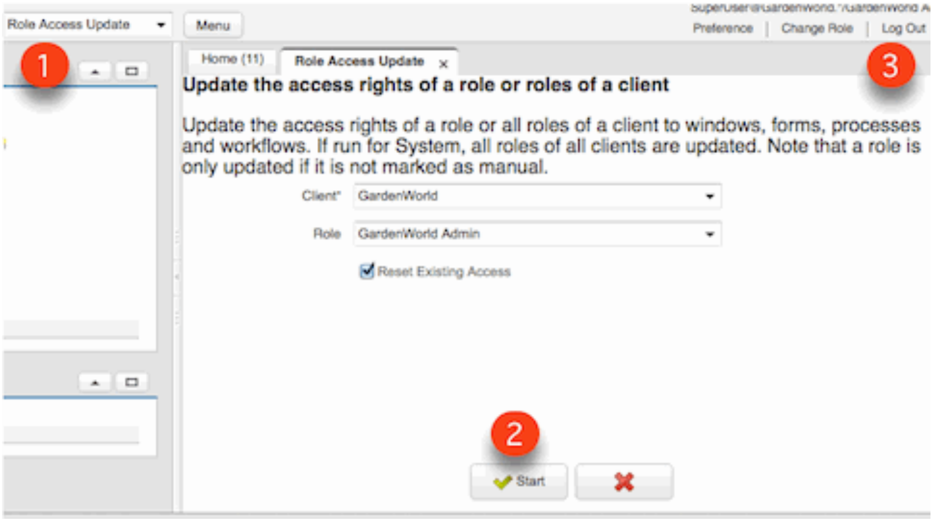
```
osgi> start 176
10:46:53.616 AdempiereActivator.start: AssetMaintenanceBase 1.0.0.201301061701 starting... [30]
10:46:53.868 PipoDictionaryService.merge: zipFilepath->/var/folders/yp/2nphbmtd1ys98pwkppmg8pcr0000gn/
T/AssetMaintenanceBase4992277080360616785.zip [30]
10:46:54.503 PipoDictionaryService.merge: dict file->/var/folders/yp/2nphbmtd1ys98pwkppmg8pcr0000gn/T/
PreventiveMaintenance/dict/PackOut.xml [30]
10:46:54.503 PackIn.importXML: importXML:/var/folders/yp/2nphbmtd1ys98pwkppmg8pcr0000gn/T/PreventiveMa
intenance/dict/PackOut.xml [30]
10:46:54.503 PackIn.importXML: starting [30]
10:46:54.974 PackIn.importXML: Start Parser [30]
10:46:55.043 PackInHandler.startElement: idempiere updateMode=false [30]
10:46:55.149 Trx.commit: **** POSaveAD_Package_Imp_583bf143-60c4-4941-87d8-0191e4aa02ed [30]
10:46:55.225 Trx.commit: **** POSaveAD_Package_Imp_Inst_dbfde17d-8965-4547-ad39-9293576b12ec [30]
10:46:56.012 PackInHandler.processElement: Processed: AD_Element - table [30]
10:46:56.251 IDFinder.findIdByColumnAndParentId: SELECT AD_Column_ID FROM AD_Column WHERE Upper(Column
Name) = ? and AD_Table_ID = ? AND AD_Client_ID IN (0, ?) ORDER BY AD_Client_ID Desc  [30]
10:46:56.989 Trx.commit: **** Trx_9475e5a8-dff4-4868-91b8-061fe150fe80 [30]
10:46:57.848 ColumnElementHandler.createColumn: CREATE TABLE MP_Meter (MP_Meter_ID NUMBER(10) NOT NULL
, CONSTRAINT MP_Meter_Key PRIMARY KEY (MP_Meter_ID)) [30]
10:46:58.107 Trx.commit: **** Trx_9475e5a8-dff4-4868-91b8-061fe150fe80 [30]
```

Note that the AdempiereActivator is starting which will launch the Pack In of the embedded 2Pack . Ensure that you have not login at your client yet. So do not call the browser yet. Do this start ID first and allow the Pack In to finish. Often you may see nothing at all due to the logger not been set to FINE. You need not do that unless you want to debug later when nothing happens after this next step.

Role Access Update

After a minute or so, go to the browser to check if your new module is there. Login as SuperUser/System at the

iDempiere webUI.



Update the access rights of a role or roles of a client

Update the access rights of a role or all roles of a client to windows, forms, processes and workflows. If run for System, all roles of all clients are updated. Note that a role is only updated if it is not marked as manual.

Client* GardenWorld

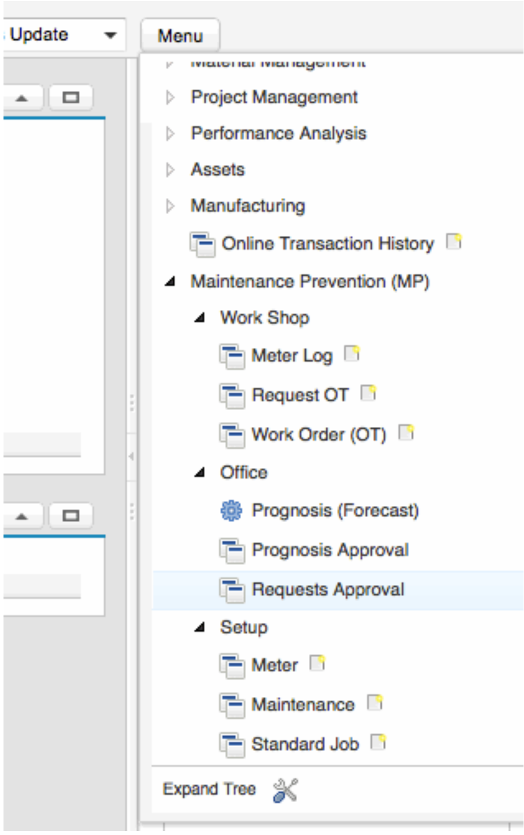
Role GardenWorld Admin

☑ Reset Existing Access

Role Access Update
Menu
SuperUser@GardenWorld.*/GardenWorld A
Preference | Change Role | Log Out
Home (11)
Role Access Update
1
2
3
Start

Select GardenAdmin and at the Lookup box on the upper right side, type in 'Role Ac' and the whole *Role Access Update* item will be found for you to click at **(1)**. Then *Start* **(2)** the process and then click on *Logout* **(3)** on the upper right side to login again. This time you should have the whole new module as shown on next page.

Review New Module Menu



Note the appearance of the new menu tree of a main node (Maintenance Prevention (MP) and three lower nodes of Work Shop, Office, and Setup. You can also look for the Asset Menu higher up and see that it has new sub-tabs which are Asset-Meter and Meter Log windows.

Your new modules are ready for you to use right away. You can check back at the console with 'ss' again and see that the AssetMaintenanceBase bundle is active and the other AssetMaintenance bundle is at resolved mode. That is a good sign.

This new arrangement of MP is to be more logical for ground operations to only use the Work Shop menu. The other operational items are more of the domain control of the office to select from prognosis forecast and generate any Work Orders.

Setup menu remains as it is to allow setting up of templates concerned.

In the Work Shop menu is a repeat of Meter Log which is to allow easier operational data entry by ground of field crew without calling up the whole long Asset window or accessing sensitive information belonging to the Accounts department.

Installing in Adempiere-Client

If you need to use the new module from the old Java client, then you have to install the same two jars but designed for the Java client side. Instead of AssetMaintenance bundle we will be using AssetMaintenanceC bundle.

If you go to the README.txt of the Maintenance P2 site at http://sourceforge.net/projects/red1/files/p2/Maintenance/

you will see there is a script you can copy and paste into your running adempiere-client's OSGi console:

install http://downloads.sourceforge.net/project/red1/p2/Maintenance/plugins/AssetMaintenanceC_1.0.0.201301061717.jar

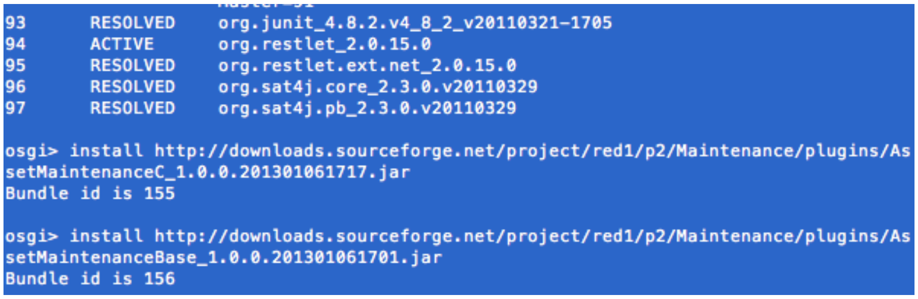
If you cannot access a console then you must add the parameter '-console' in your adempiere-client launch script. This is just a simple command to tell the install to look for the bundle with the URL parameter. If you have downloaded the bundle prior, then just give the location with:

install file:/folder/AssetMaintenanceC_1.0.0.201301061717.jar'.

You should see a bundle ID feedback from the console. When you type SS again you will see the bundle added.Then you need to add the same base bundle which is also used by the client. You can use the following script in the OSGi console:

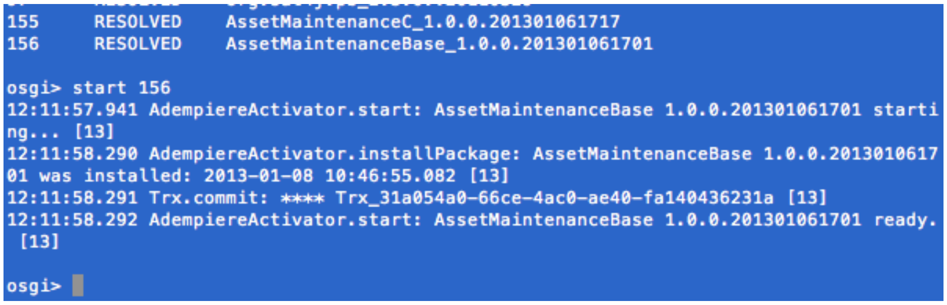
install http://downloads.sourceforge.net/project/red1/p2/Maintenance/plugins/AssetMaintenanceBase_1.0.0.201301061701.jar

You will be given a bundle ID in the display. Again like in idempiere-server case before this, note the ID value and do a ‘start <ID>’. In the case shown below here, it will be ‘**start 156**’.



```
93      RESOLVED    org.junit_4.8.2.v4_8_2_v20110321-1705
94      ACTIVE      org.restlet_2.0.15.0
95      RESOLVED    org.restlet.ext.net_2.0.15.0
96      RESOLVED    org.sat4j.core_2.3.0.v20110329
97      RESOLVED    org.sat4j.pb_2.3.0.v20110329

osgi> install http://downloads.sourceforge.net/project/red1/p2/Maintenance/plugins/As
setMaintenanceC_1.0.0.201301061717.jar
Bundle id is 155

osgi> install http://downloads.sourceforge.net/project/red1/p2/Maintenance/plugins/As
setMaintenanceBase_1.0.0.201301061701.jar
Bundle id is 156
```

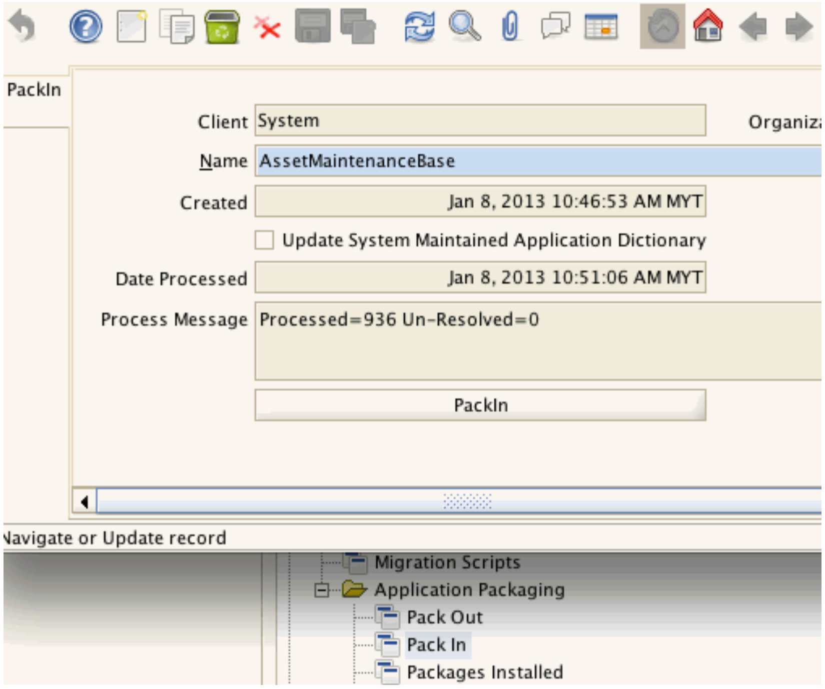
Note that if you have done a start in idempiere-server, the Pack In will not repeat the 2Pack pack in. It will do a check of the installed Packages within the database to see if it is already existing before doing so. Again if you wish to do so, do not login into any client first. Below is what it will say which is AssetMaintenanceBase 1.0.0.201301061701 was installed: 2013-01-08 10:46



```
155     RESOLVED    AssetMaintenanceC_1.0.0.201301061717
156     RESOLVED    AssetMaintenanceBase_1.0.0.201301061701

osgi> start 156
12:11:57.941 AdempiereActivator.start: AssetMaintenanceBase 1.0.0.201301061701 starti
ng... [13]
12:11:58.290 AdempiereActivator.installPackage: AssetMaintenanceBase 1.0.0.2013010617
01 was installed: 2013-01-08 10:46:55.082 [13]
12:11:58.291 Trx.commit: **** Trx_31a054a0-66ce-4ac0-ae40-fa140436231a [13]
12:11:58.292 AdempiereActivator.start: AssetMaintenanceBase 1.0.0.201301061701 ready.
 [13]

osgi>
```

Review Pack In Details

If you login into the Java client with SuperUser/System, go into System Admin role and call up the Pack In, you can read the *Process Message* to see that the new MP module is successfully installed with no Un-Resolved items.



PackIn

Client: System

Organiz

Name: AssetMaintenanceBase

Created: Jan 8, 2013 10:46:53 AM MYT

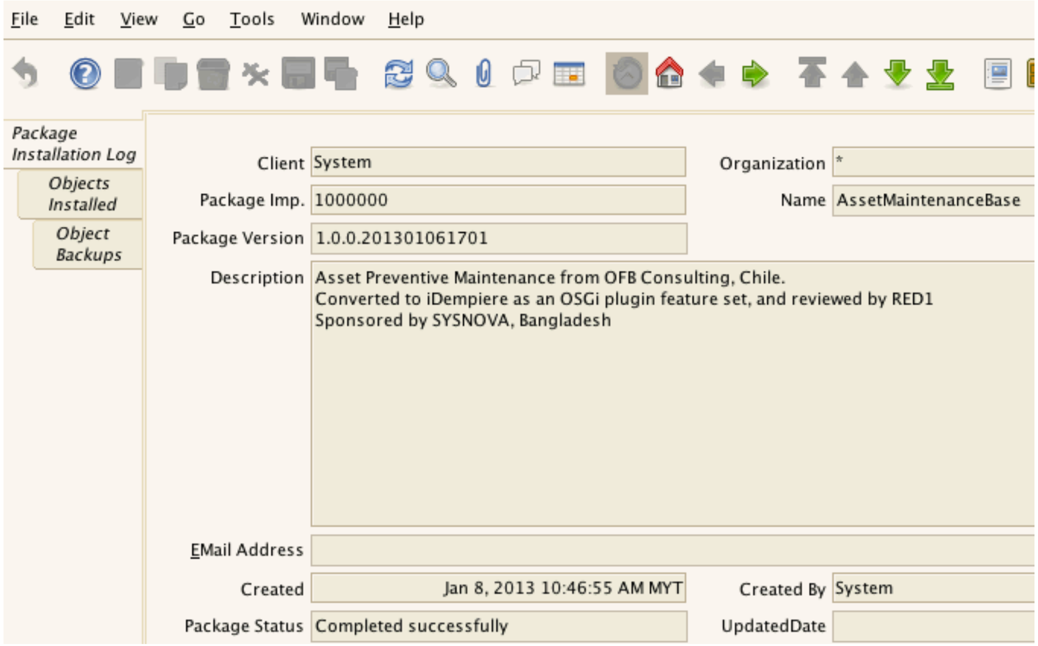
□ Update System Maintained Application Dictionary

Date Processed: Jan 8, 2013 10:51:06 AM MYT

Process Message: Processed=936 Un-Resolved=0

You can also open up the Package Maintenance window to look at the Package Installation log. Note the Name of the Package is AssetMaintenanceBase if you Pack In via the OSGi console. And the Package Version is the exact version of bundle jar. This is convenient to let you check and confirm your Package installed the right jar version.

At the Objects Installed tab, you can see the long list of objects installed too.



Client: System
Organization: *
Package Imp.: 1000000
Name: AssetMaintenanceBase
Package Version: 1.0.0.201301061701
Description: Asset Preventive Maintenance from OFB Consulting, Chile.
Converted to iDempiere as an OSGi plugin feature set, and reviewed by RED1
Sponsored by SYSNOVA, Bangladesh
EMail Address:
Created: Jan 8, 2013 10:46:55 AM MYT
Created By: System
Package Status: Completed successfully
UpdatedDate:

Then logout and change role to GardenAdmin, login and you can see the new module menu sub-tree.

(You may need to do a Role Access Update if you do not see it fully (that is if you have not done so while in idempiere-server side.)

Review of Achievement

To recap the steps you took to arrive at this complete installation of the new module are actually very simple and more advanced than the traditional approach from Compiere or ADempiere. Using the new OSGi P2 update approach reduces recompilation, re-setup of ADempiere customization jars, and later hassle of uncoupling if this new module is not needed.

In case of uninstalling these bundles is merely with an 'uninstall <ID> ' in the OSGi console for adempiere-client and the uninstall script with the -i setting for idempiere-server.

The data is still present and need the usual manual unchecking in the Application Dictionary. But the impact of putting in and taking out old code bundles are no longer the concern. OSGi is designed for bundles to be resolved on their own, working in a cohesive but loosely coupled manner with other bundles.

You are good and ready to use the new module from the Java client now.

Another cheat-sheet here to summarise all the steps:

1. Install iDempiere v1.0.a
2. Update the code with idempiere-server-Update
3. Apply latest migration scripts to for v1.0.a.
4. Install MP module and do 2Pack.
5. Login as SuperUser/System and run *Role Access Update* in the ClientAdmin
6. Relogin to view the new module and start using it under Implement section.
7. Install sample data (to be done in next Implement section)

Implement

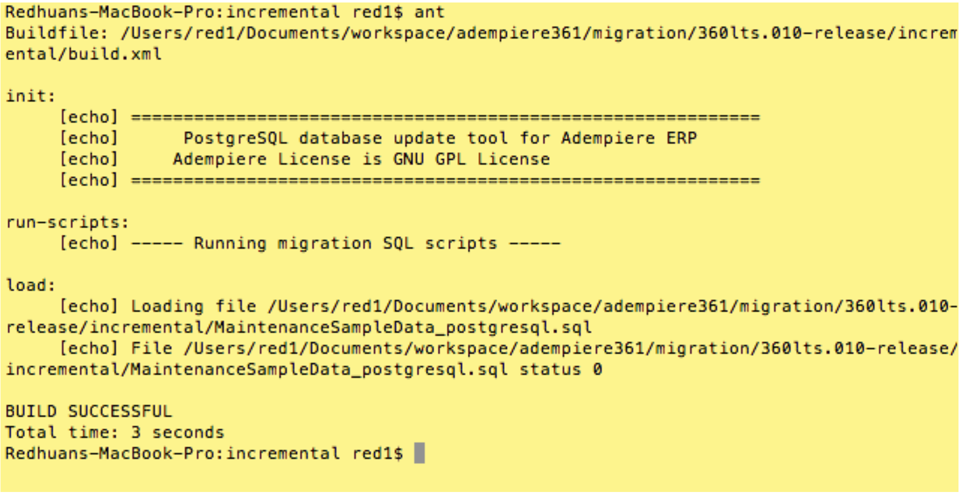
To start using the MP module you first have to setup the templates. That is to define:

1. a Meter
2. a Standard Job
3. a Maintenance record
4. an Asset
5. an Asset Meter
6. some Meter Logs.

If you wish to play around then I have prepare sample setup data for GardenWorld client. You can download and apply this migration scripts into your PostgerSQL database. I assume few are using Oracle to test this but if you need to, ask in the public forum and i will post it there.

Get the migration script here
http://sourceforge.net/projects/red1/files/p2/Maintenance/MaintenanceSampleData_postgresql.sql/download

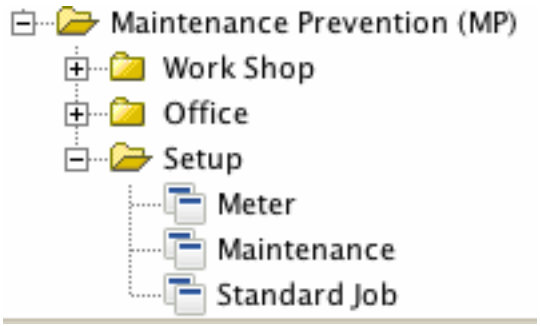
The usual user or developer of ADempiere would know how to apply this migration script to get the sample data working.



```
Redhuans-MacBook-Pro:incremental red1$ ant
Buildfile: /Users/red1/Documents/workspace/adempiere361/migration/360lts.010-release/increm
ental/build.xml

init:
     [echo] ============================================================
     [echo]      PostgreSQL database update tool for Adempiere ERP
     [echo]     Adempiere License is GNU GPL License
     [echo] ============================================================

run-scripts:
     [echo] ----- Running migration SQL scripts -----

load:
     [echo] Loading file /Users/red1/Documents/workspace/adempiere361/migration/360lts.010-
release/incremental/MaintenanceSampleData_postgresql.sql
     [echo] File /Users/red1/Documents/workspace/adempiere361/migration/360lts.010-release/
incremental/MaintenanceSampleData_postgresql.sql status 0

BUILD SUCCESSFUL
Total time: 3 seconds
Redhuans-MacBook-Pro:incremental red1$ █
```

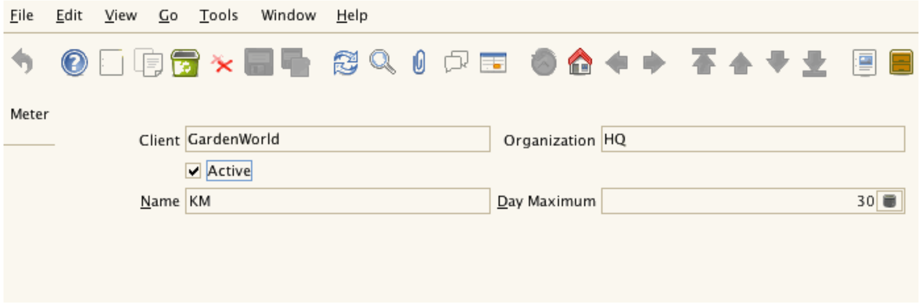
Otherwise you can also do it manually from the screenshots of the templates to be shown after this. We will start off to define the first three items on our list - a Meter, a Standard Job with an Asset, and then the Maintenance record.

METER
MAINTENANCE
TASK
RESOURCES
STANDARD JOB



Meter Record

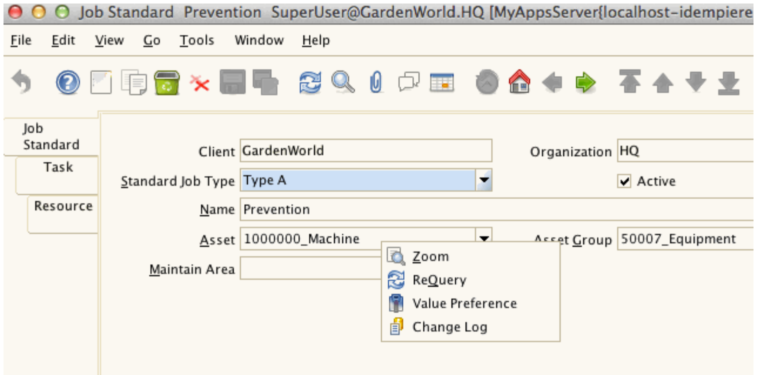
There are only two type of program schedule, either meter based or calendar based. The calendar type is singular and need no definition. The meter type has different units of measure such as in this case it is KM for kilometer.



Client GardenWorld
Organization HQ
☑ Active
Name KM
Day Maximum 30

With a simple meter record defined we can proceed to define other template data. Next we go to the Standard Job template.

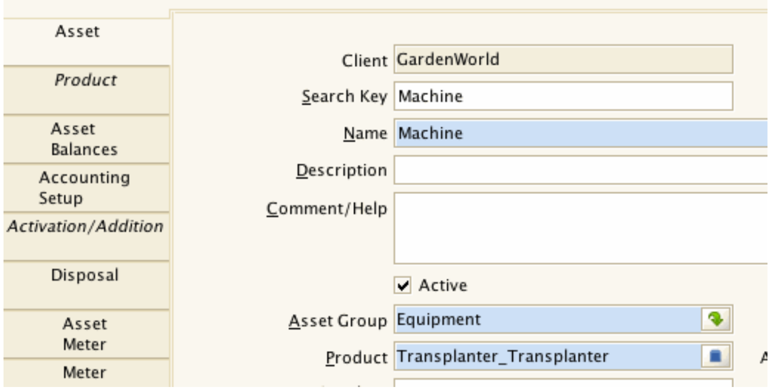
Standard Job



Job Standard Prevention SuperUser@GardenWorld.HQ [MyAppsServer{localhost-idempiere

Client GardenWorld
Organization HQ
Standard Job Type Type A
☑ Active
Name Prevention
Asset 1000000_Machine
Asset Group 50007_Equipment
Maintain Area

A Standard Job is for defining jobs that are repetitive or routine in most maintenance work. At the first tab, we define the header information and we need to associate it with an asset. But we have not defined any. We can do that from this tab by a right click on the field and selecting Zoom. This leads us to the Asset window automatically for us. Below is what our sample looks like.

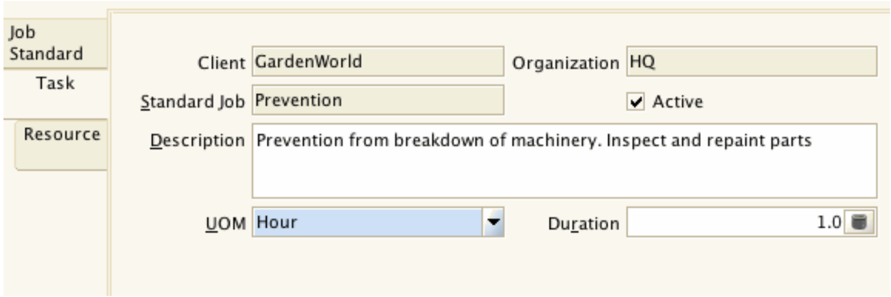


Client GardenWorld
Search Key Machine
Name Machine
Description
Comment/Help
☑ Active
Asset Group Equipment
Product Transplanter_Transplanter

Asset Record

After defining an asset, we then return to the Standard Job tab and ReQuery the Asset field for the asset to appear. Next we go to the second tab which is the Task tab.

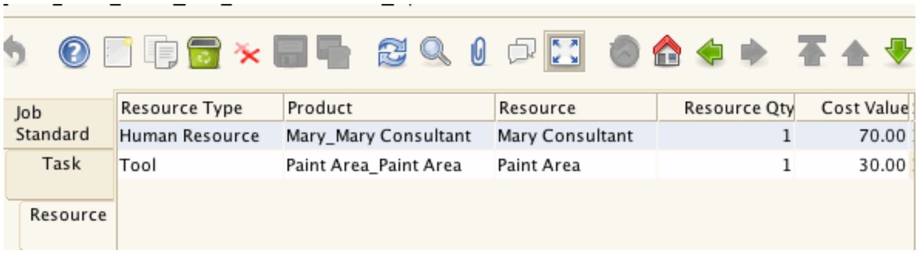
Standard Job Task



Client	GardenWorld	Organization	HQ
Standard Job	Prevention	☑ Active	
Description	Prevention from breakdown of machinery. Inspect and repaint parts		
UOM	Hour	Duration	1.0

The task tab defines what tasks or activity that are needed for this standard job to be accomplished. Here we are going to define just a single task that takes an hour to carry out but there will be two resources that we will use for it.

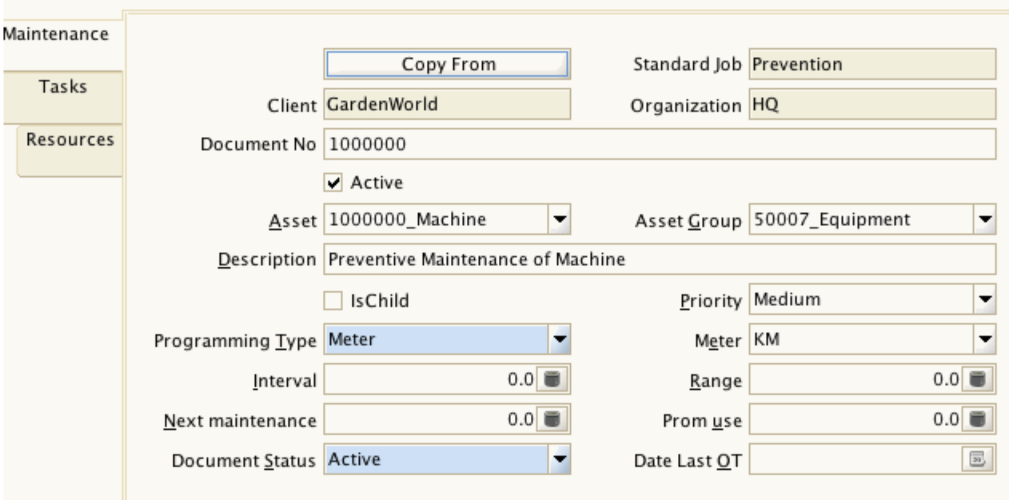
Job Task Resources



Resource Type	Product	Resource	Resource Qty	Cost Value
Human Resource	Mary_Mary Consultant	Mary Consultant	1	70.00
Tool	Paint Area_Paint Area	Paint Area	1	30.00

We create two records by filling in two different product resources and we can put a cost to each resource. This completes our sample Standard Job. It is ready for reuse by Maintenance jobs or Work Order requests that refer to this standard job without repeating the same tasks and resources. We see the first example of its reuse in a Maintenance record.

Maintenance record



	Copy From	Standard Job	Prevention
Client	GardenWorld	Organization	HQ
Document No	1000000		
	☑ Active		
Asset	1000000_Machine	Asset Group	50007_Equipment
Description	Preventive Maintenance of Machine		
	□ IsChild	Priority	Medium
Programming Type	Meter	Meter	KM
Interval	0.0	Range	0.0
Next maintenance	0.0	Prom use	0.0
Document Status	Active	Date Last OT	

A Maintenance record is the most important one that tracks a complete maintenance cycle comprising of not just tasks and resources but also the type of Programming Type if it is Meter or Calendar based, and how the asset is best maintained with the control information here and at the Asset Meter which we will look at next. Here we explain each field as we go along with the Meter based on KM and the Calendar been obvious:

1. **Asset** is associated in the Maintenance record so that its tasks are defined for each asset.

2. **Copy From** will now take the Standard Job defined. You click on this after assigning an asset to this record. After selecting the template you can examine the sub tabs of Tasks and Resources been populated from the selection.

3. **Interval** is used to calculate the value of the next Maintenance run. If it is Calendar ProgrammingType then it is a date value. If it is Meter ProgrammingType then it is the meter's unit value. For example if under Calendar the interval is 10 then it means the next 10 days after any run. For our Meter, a value of 100 will mean the next 100 KM.

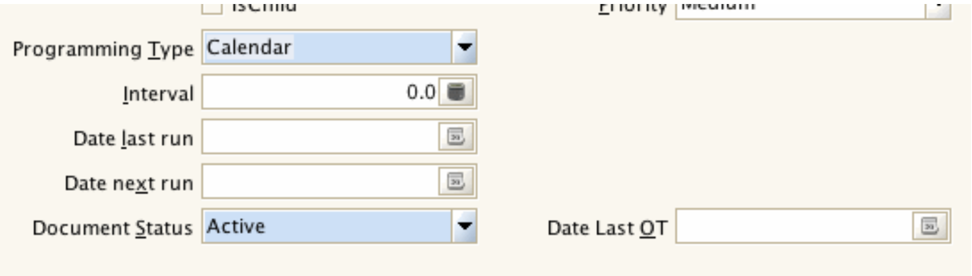
4. **Next Maintenance** gives the KM value the task is supposed to be carried out. Now it is zero as it is new. It will check the meter logs during Prognosis run whether any logs or meter readings are taken at that date or after to generate a Work Order. Under Calendar mode, this is not used. It will have its own field which we shall see later.

5. **Range** is the allowance of value to differ from the next maintenance date. For example if the date is not due but the range is added to make it fall within the range, then it is set for a Work Order too.

6. **Prom-use** is the field used in calculation used by the Meter, based on this formula: prom=(lastM-firstM) / (days/(60*60*24*1000)), where the difference in top and bottom meter readings are divided over the days between them divided further to the milliseconds. Prom value is added to the last reading during Prognosis run.

7. **Date Last OT** is the date the last work order is processed.

Now we switch the Maintenance programming type to Calendar based.



IsChild

Priority Medium

Programming Type Calendar

Interval 0.0

Date last run

Date next run

Document Status Active

Date Last OT

Now we see different fields become active and the others are removed. These are the values relevant to a Calendar type maintenance task:

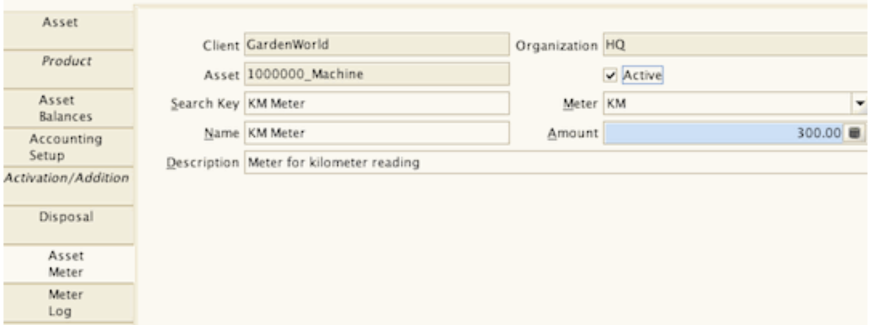
1. Interval as explained earlier is the number of days the next run will be set to after the last run.

2. Date last run is set after a prognosis generated work order based on this record is processed.

3. Date next run is calculated during that work order process based on the interval given.

Asset Meter

Now we come to the Asset Meter setting which is done under the Asset Window. We call out the Asset Meter tab:
This has the Meter set to KM meter and the Amount is already set due to some Meter Log records we already entered. That is what we will look at next. Asset meter is also used for Calendar type where the Meter type KM is not used.



Client GardenWorld

Organization HQ

Asset 1000000_Machine

☑ Active

Search Key KM Meter

Meter KM

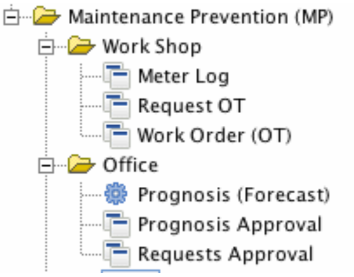
Name KM Meter

Amount 300.00

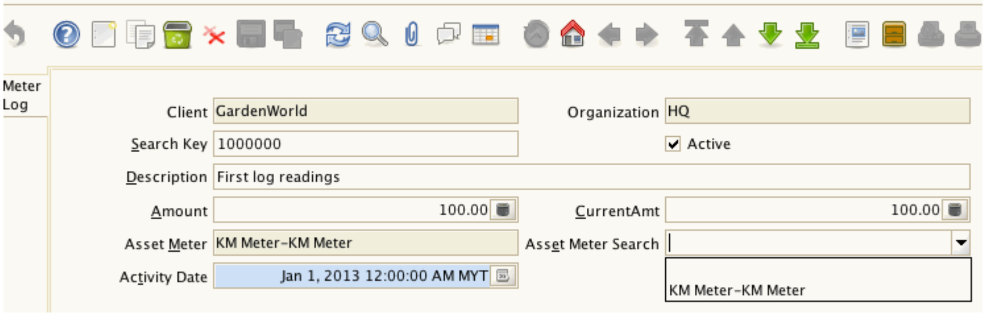
Description Meter for kilometer reading

Meter Log

Even though Meter Log window is defined under the Asset Window associated to its Asset Meter, it is also available under its own window in the Work Shop sub-menu. Reason is that the routine work may be done on a horizontal basis in the field where each asset meter reading is taken in a master log book by a dedicated team. Thus there is no need to go through each asset record which may contain sensitive accounting information.

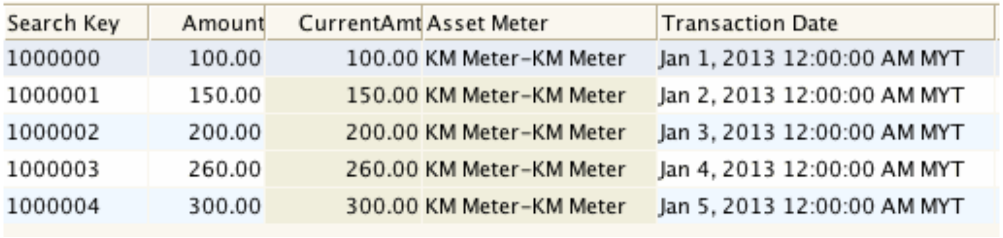


In the single record view of the Meter Log as shown below note the bottom right field called Asset Meter Search which has a pull down list to select the Asset Meter this log is tagged to. In such a manner there is no need to open this log window under an Asset and Asset Meter hierarchy. It can thus stand alone and be tagged separately.


Meter Log

Client	GardenWorld	Organization	HQ
Search Key	1000000	☑ Active	
Description	First log readings		
Amount	100.00	CurrentAmt	100.00
Asset Meter	KM Meter-KM Meter	Asset Meter Search	
Activity Date	Jan 1, 2013 12:00:00 AM MYT		KM Meter-KM Meter

Meter log window is used for data entry of meter readings on a regular basis according to the ground crew or work shop procedure or routine.



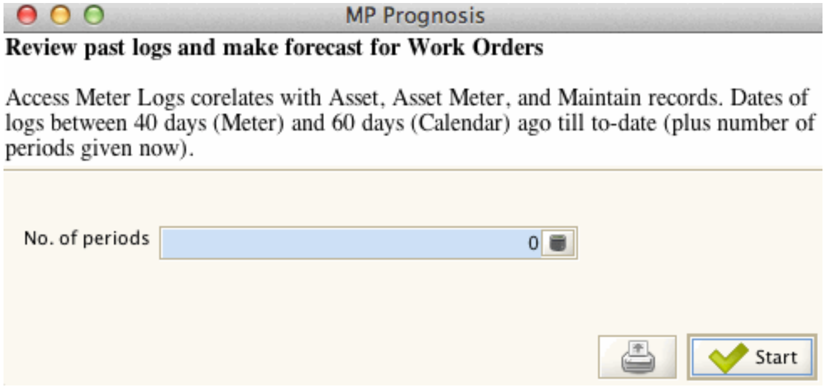
Search Key	Amount	CurrentAmt	Asset Meter	Transaction Date
1000000	100.00	100.00	KM Meter-KM Meter	Jan 1, 2013 12:00:00 AM MYT
1000001	150.00	150.00	KM Meter-KM Meter	Jan 2, 2013 12:00:00 AM MYT
1000002	200.00	200.00	KM Meter-KM Meter	Jan 3, 2013 12:00:00 AM MYT
1000003	260.00	260.00	KM Meter-KM Meter	Jan 4, 2013 12:00:00 AM MYT
1000004	300.00	300.00	KM Meter-KM Meter	Jan 5, 2013 12:00:00 AM MYT

Here we have 5 records entered on different daily basis with progressive meter readings. Note that these readings are of the KM Meter under the Asset Meter defined earlier. And this is associated with the Maintenance record as defined earlier which indirectly says what its ProgrammingType is. We shall use this values under the Meter type first and then later use the Calendar type to see how it works.

To get this logs to some use we run the next item on our process menu.

Prognosis (Forecast)

We will run the prognosis process today (8th January) and see what will happen.

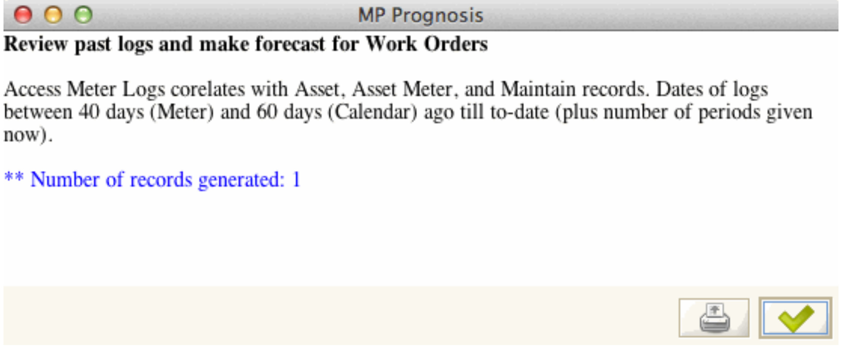

MP Prognosis

Review past logs and make forecast for Work Orders

Access Meter Logs corelates with Asset, Asset Meter, and Maintain records. Dates of logs between 40 days (Meter) and 60 days (Calendar) ago till to-date (plus number of periods given now).

No. of periods: 0

The No. of periods field can be set to allow more Work Orders be generated in a future forecasted manner. Thus the management office that handles this process has a flexible tool suitable to its management plan. We leave the value at zero and press the Start button. The following response is shown.



Review past logs and make forecast for Work Orders

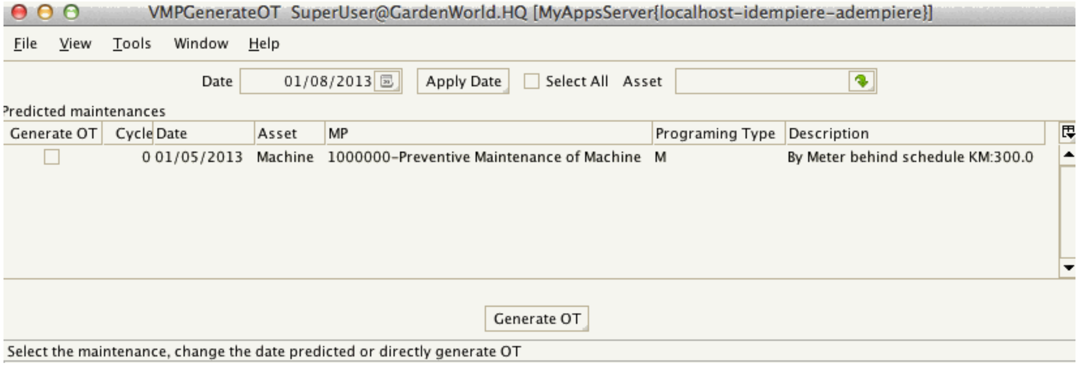
Access Meter Logs corelates with Asset, Asset Meter, and Maintain records. Dates of logs between 40 days (Meter) and 60 days (Calendar) ago till to-date (plus number of periods given now).

** Number of records generated: 1

(This is one of the simple but nice enhancement I did during this review. I return the number of records generated. Previously such feedback were left out from the code).

Prognosis Approval

Now we look up the generated prognosis record to approve it and convert it to a Work Order.


VMPGenerateOT SuperUser@GardenWorld.HQ [MyAppsServer{localhost-idempiere-adempiere}]
File
View
Tools
Window
Help
Apply Date
Generate OT

Date 01/08/2013 □ Select All Asset

Predicted maintenances

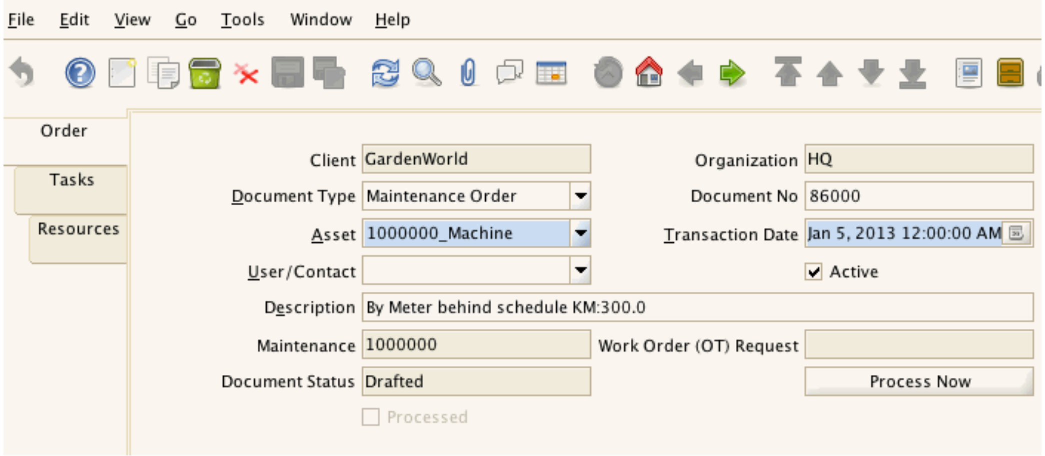
Generate OT	Cycle	Date	Asset	MP	Programing Type	Description
□	0	01/05/2013	Machine	1000000-Preventive Maintenance of Machine	M	By Meter behind schedule KM:300.0

Select the maintenance, change the date predicted or directly generate OT

We select the record displayed that we wish to approve. In this case there is only one. We select by checking the Generate OT checkbox on the left. Then at the bottom we press the Generate OT button. The record line disappears. We then look at the Work Order window.

Work Order (OT)

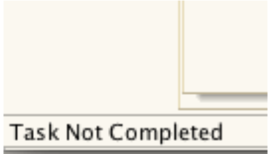
Under the Work Order window we see the generated job ready for attention by the field crew.



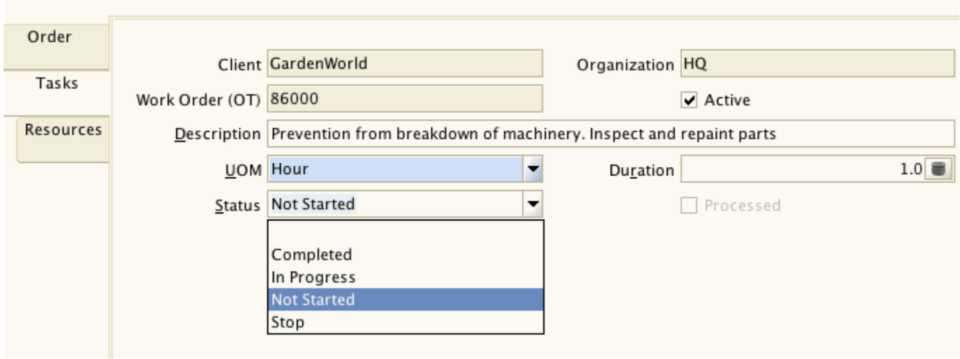
Client	GardenWorld	Organization	HQ
Document Type	Maintenance Order	Document No	86000
Asset	1000000_Machine	Transaction Date	Jan 5, 2013 12:00:00 AM
User/Contact			☑ Active
Description	By Meter behind schedule KM:300.0		
Maintenance	1000000	Work Order (OT) Request	
Document Status	Drafted		
	□ Processed		

Note that the Transaction Date is the last Log date entry. The Description is also making a 'behind schedule' remark due to the calculation during Prognosis earlier on that takes the Maintenance record values and match with the Asset Meter and concluding that the maintenance is past due. It is also indicating that this is a scheduled Maintenance record and not a requested Work Order. Now this window is passed to the maintenance crew for processing. They have to go through each task in the Tasks tab and see what is to be done. Under each task will be shown the resources or personnel responsible or the equipment or ingredients or tools needed for each task.

If you press the Process Now button it will only show a Task Not Completed status. Thus you need to change the status of the underlying Tasks tab.

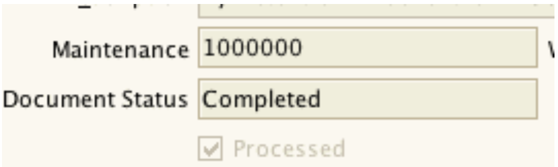

Task Not Completed

Under the Tasks tab, the task has to be modified to have its status changed to Completed before the Order can be finished. So this acts as a checklist for the crew to note and go through.

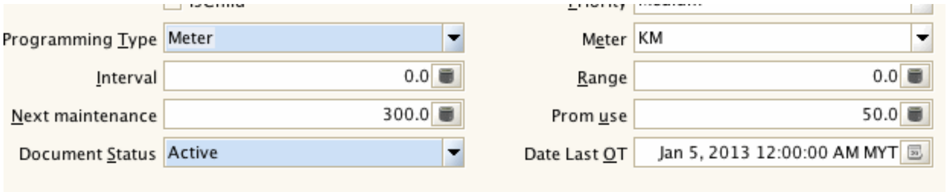


Client GardenWorld
Organization HQ
Work Order (OT) 86000
☑ Active
Description Prevention from breakdown of machinery. Inspect and repaint parts
UOM Hour
Duration 1.0
Status Not Started
□ Processed

Here we change the status to completed and then return to the Order tab for the Process Now action. This time when we select the Complete action it does change the status to complete.

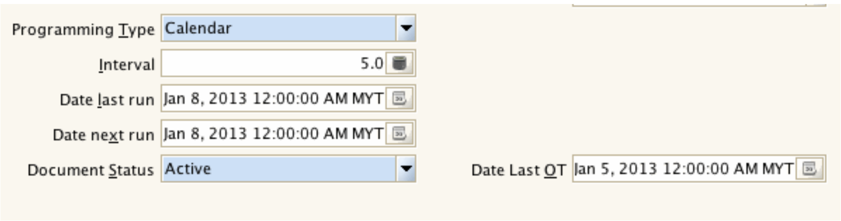

Maintenance 1000000
Document Status Completed
☑ Processed

This conclude one cycle of exercise of using the Maintenance module. We return to the Maintenance window and we can see it has been changed accordingly. The Asset Meter will also reflect the last reading which is 300.

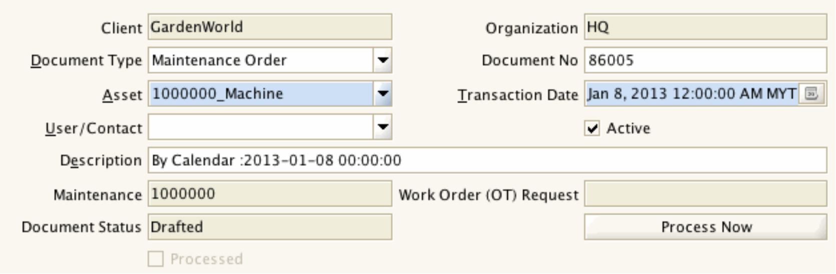

Programming Type Meter
Meter KM
Interval 0.0
Range 0.0
Next maintenance 300.0
Prom use 50.0
Document Status Active
Date Last OT Jan 5, 2013 12:00:00 AM MYT

Try this exercise again with an interval of 100 and we get the following Next Maintenance as 400.

You can try another exercise using the Calendar type and setting the Date next run as today or else Prognosis will find nothing to run. We also set the Interval to 5 days.

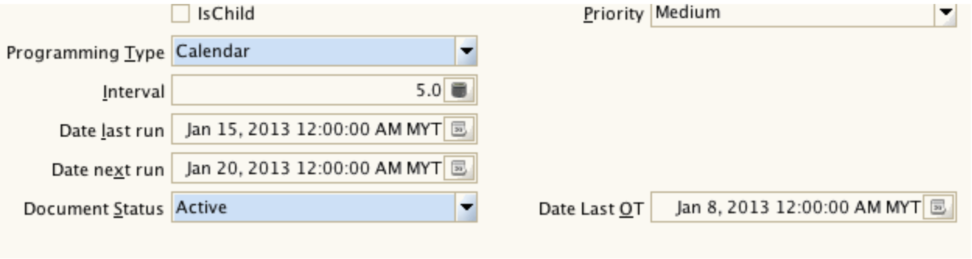

Programming Type Calendar
Interval 5.0
Date last run Jan 8, 2013 12:00:00 AM MYT
Date next run Jan 8, 2013 12:00:00 AM MYT
Document Status Active
Date Last OT Jan 5, 2013 12:00:00 AM MYT

When we process Prognosis Approval and Generate OT we get the following remarks in the Description to be by Calendar. After processing it we can check back the Maintenance record to see the effect.


Client GardenWorld
Organization HQ
Document Type Maintenance Order
Document No 86005
Asset 1000000_Machine
Transaction Date Jan 8, 2013 12:00:00 AM MYT
User/Contact
☑ Active
Description By Calendar :2013-01-08 00:00:00
Maintenance 1000000
Work Order (OT) Request
Document Status Drafted

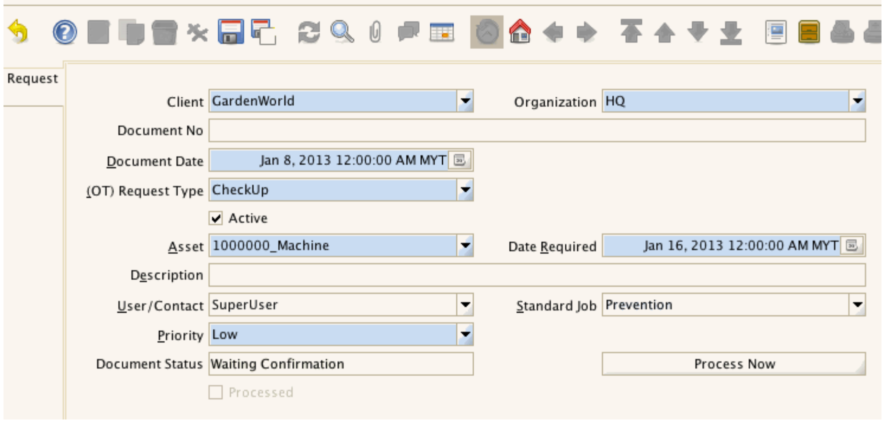
□ Processed

After running Prognosis one more time and processing the resulting Work Order we have the following result in the Maintenance tab. Note the next run date is set at 5 days interval as required.



□ IsChild
Priority: Medium
Programming Type: Calendar
Interval: 5.0
Date last run: Jan 15, 2013 12:00:00 AM MYT
Date next run: Jan 20, 2013 12:00:00 AM MYT
Document Status: Active
Date Last OT: Jan 8, 2013 12:00:00 AM MYT

Next we see how to do ad-hoc Work Order requests called through Request OT in the Work Shop menu.



Client: GardenWorld
Organization: HQ
Document No:
Document Date: Jan 8, 2013 12:00:00 AM MYT
(OT) Request Type: CheckUp
☑ Active
Asset: 1000000_Machine
Date Required: Jan 16, 2013 12:00:00 AM MYT
Description:
User/Contact: SuperUser
Standard Job: Prevention
Priority: Low
Document Status: Waiting Confirmation
Process Now
□ Processed

The main importance in this window is the selection of a Standard Job template to allow final generation of a ready made Job template similar to the Prognosis process. From there the Office can approve the requests and likewise they will make their way to the Work Order window for completion.

Things Not Available

However I like to note some things that do not seem to be available in this module.

1. No reports. There are no reports whatsoever. Some suggested useful reports can be:

    a. Scheduled work orders not completed

    b. Deployment of resources, their calendar of activity derived from the planned Maintenance schedules

2. The generation of internal invoices to convert working hours to payable hours for payroll and costing purposes.

3. The housekeeping of meter logs. They are not deactivated or removed from further similar prognosis process.

Innovate

Now we come to the most important purpose of Free and Open Source Software which is the freedom to innovate. We shall examine the source-code, the OSGi framework and the specific module code that determines the processes logic which we seen through while trying out the module above.

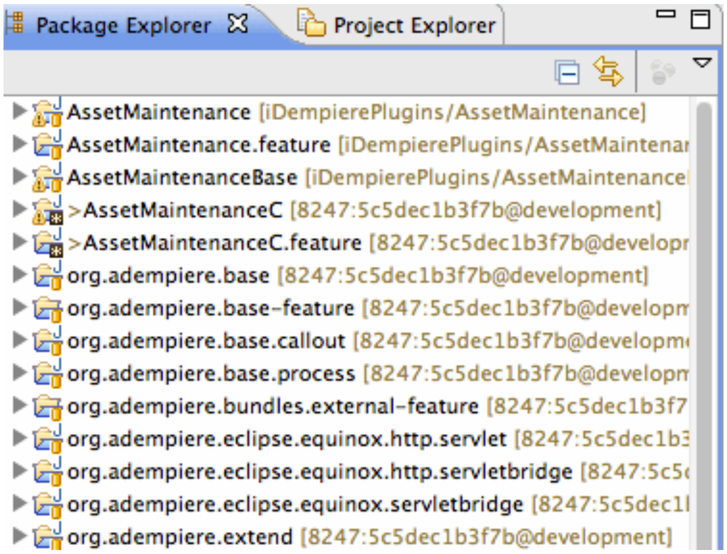
Source Code

The source-code of this project is uploaded as three sub projects in http://sf.net/p/red1

Below is the screenshot of the Eclipse space that shows the three projects alongside the iDempiere work-space.

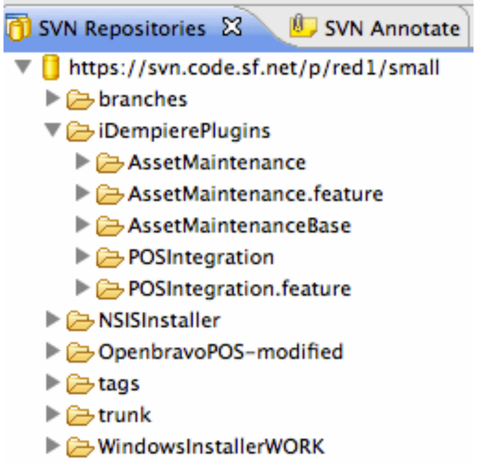
They are :

1. AssetMaintenance that contains code for ZK Ajax UI
2. AssetMaintenanceC that contains code for Swing UI
3. AssetMaintenanceBase that is common code shared by the two UIs.
4. AssetMaintenance.feature that packages both 1 and 2 above as a single installable unit for idempiere-server.
5. AssetMaintenanceC.feature which does the same for Swing UI or adempiere-client but is not necessary as we install them directly with the two bundles as shown early on.



```
AssetMaintenance [iDempierePlugins/AssetMaintenance]
AssetMaintenance.feature [iDempierePlugins/AssetMaintenar
AssetMaintenanceBase [iDempierePlugins/AssetMaintenance
>AssetMaintenanceC [8247:5c5dec1b3f7b@development]
>AssetMaintenanceC.feature [8247:5c5dec1b3f7b@developr
org.adempiere.base [8247:5c5dec1b3f7b@development]
org.adempiere.base-feature [8247:5c5dec1b3f7b@developm
org.adempiere.base.callout [8247:5c5dec1b3f7b@developm
org.adempiere.base.process [8247:5c5dec1b3f7b@developm
org.adempiere.bundles.external-feature [8247:5c5dec1b3f7
org.adempiere.eclipse.equinox.http.servlet [8247:5c5dec1b3
org.adempiere.eclipse.equinox.http.servletbridge [8247:5c5
org.adempiere.eclipse.equinox.servletbridge [8247:5c5dec1l
org.adempiere.extend [8247:5c5dec1b3f7b@development]
```

The source-code is uploaded into SVN repositories under the path of http://svn.code.sf.net/p/red1/small as shown here.



```
https://svn.code.sf.net/p/red1/small
  branches
  iDempierePlugins
    AssetMaintenance
    AssetMaintenance.feature
    AssetMaintenanceBase
    POSIntegration
    POSIntegration.feature
  NSISInstaller
  OpenbravoPOS-modified
  tags
  trunk
  WindowsInstallerWORK
```

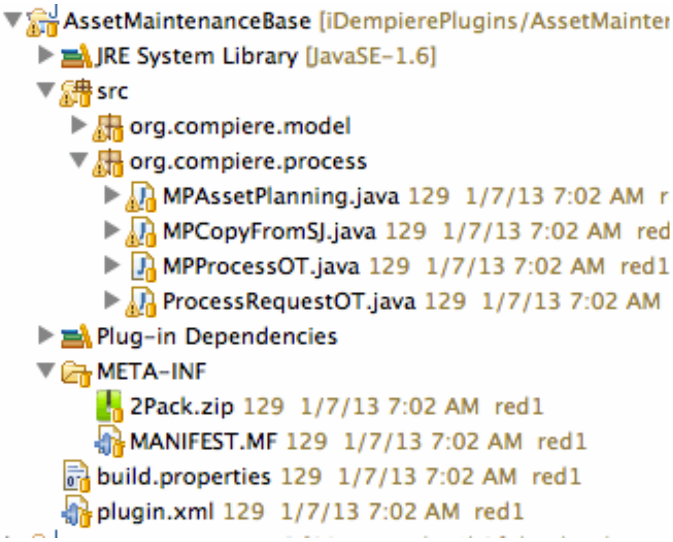
I also uploaded the installer scripts as shown early at the Install section which can be openly changed to suit your development purposes.

Next we shall go into the structure of the AssetMaintenance plugins where the plugins are separated for different UIs with a base plugin that back-end processes and models are kept. We shall first look at the base plugin.

If you follow the iDempiereMigrationGuide.pdf you would have been familiar with how a plugin is created and how extensions in iDempiere are made.

AssetMaintenanceBase plugin

This plugin is basically used by both UIs, so common code are retained and UI specific code are taken out into the other plugins. It also contain the 2Pack.zip that has the migration scripts to install the menu, windows, tabs, tables, fields and reference types as well as the reference lists of the module.

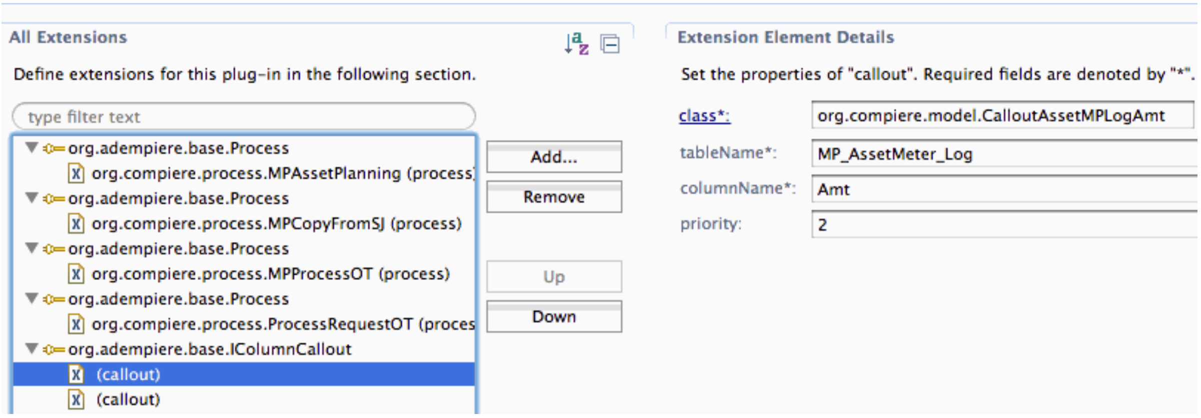


```
▼ AssetMaintenanceBase [iDempierePlugins/AssetMainte
   ▶ JRE System Library [JavaSE-1.6]
   ▼ src
      ▶ org.compiere.model
      ▼ org.compiere.process
         ▶ MPAssetPlanning.java 129  1/7/13 7:02 AM  r
         ▶ MPCopyFromSJ.java 129  1/7/13 7:02 AM  red
         ▶ MPProcessOT.java 129  1/7/13 7:02 AM  red1
         ▶ ProcessRequestOT.java 129  1/7/13 7:02 AM
   ▶ Plug-in Dependencies
   ▼ META-INF
        2Pack.zip 129  1/7/13 7:02 AM  red1
        MANIFEST.MF 129  1/7/13 7:02 AM  red1
     build.properties 129  1/7/13 7:02 AM  red1
     plugin.xml 129  1/7/13 7:02 AM  red1
```

org.compiere.model contains the many table model classes together with its X and I classes. The MANIFEST overview will have the usual AdempiereActivator from utils plugin to make the 2Pack process work upon startup.

Plugin.xml Callout Extension

The plugin.xml that defined the Extensions are as shown here.



Define extensions for this plug-in in the following section.

type filter text

```
▼ org.adempiere.base.Process
     org.compiere.process.MPAssetPlanning (process)
▼ org.adempiere.base.Process
     org.compiere.process.MPCopyFromSJ (process)
▼ org.adempiere.base.Process
     org.compiere.process.MPProcessOT (process)
▼ org.adempiere.base.Process
     org.compiere.process.ProcessRequestOT (proces
▼ org.adempiere.base.IColumnCallout
     (callout)
     (callout)
```

Set the properties of "callout". Required fields are denoted by "*".

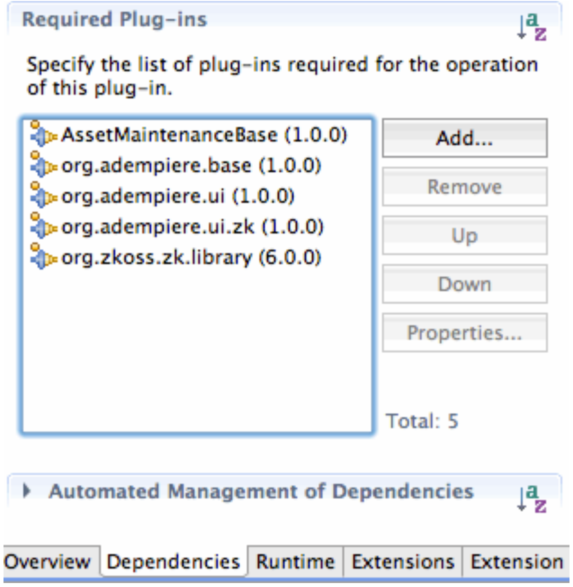
class*:	org.compiere.model.CalloutAssetMPLogAmt
tableName*:	MP_AssetMeter_Log
columnName*:	Amt
priority:	2

Note the usual Process extensions are easily defined as similar to our OpenbravoPOS integration work before. This time the new stuff is the Extension from IColumnCallout extension point. It is used to tag the Callouts that are used in this module. We only have two Callouts and both are defined within one Extension as shown above. Each callout item is defined accordingly as shown. The actual Callout within the Table-Column field is then removed as it will not work outside the iDempiere core modules, which has to work via such extensions. In order for the callout class to be accessible by the extension, it has to implement the IColuimnCallout interface as shown on the next page.

```
public class CalloutAssetMPLogAmt extends CalloutAssetMPLog implements
IColumnCallout {

       @Override
       public String start(Properties ctx, int WindowNo, GridTab mTab,
                    GridField mField, Object value, Object oldValue) {
              return assetLog(ctx, WindowNo, mTab, mField,value);
       }

}
```

The return statement then call our legacy Callout class CalloutAssetMPLog which contains the original code that handles the callout.

AssetMaintenance ZK UI



Specify the list of plug-ins required for the operation of this plug-in.

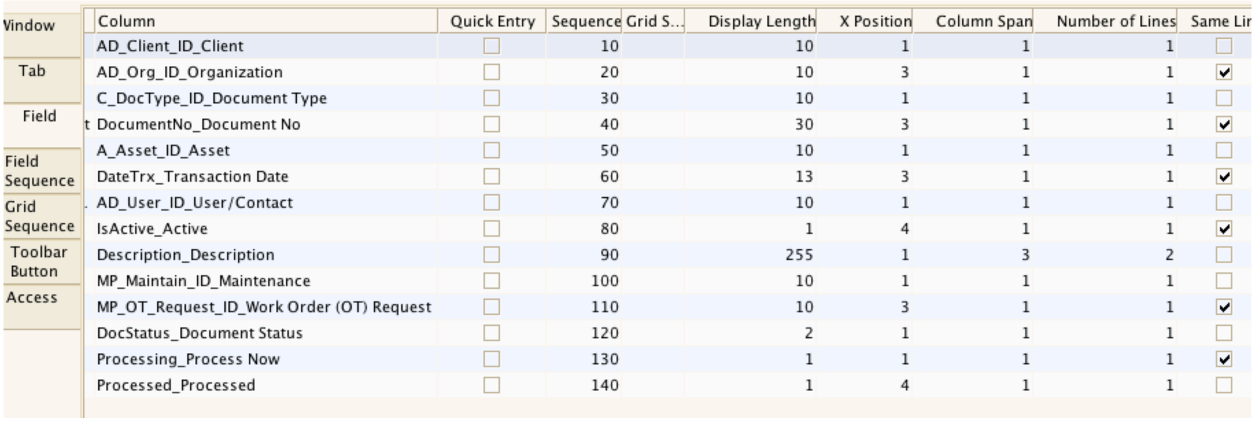
- AssetMaintenanceBase (1.0.0)
- org.adempiere.base (1.0.0)
- org.adempiere.ui (1.0.0)
- org.adempiere.ui.zk (1.0.0)
- org.zkoss.zk.library (6.0.0)

Total: 5

As stated, this plugin and the Swing code plugin are separated out for easier maintenance and more lightweight deployment. This plugin depends on the base plugin. Thus in its MANIFEST this is made clear as shown here on the right.

The ZK UI code in this module are a bit outdated and require upgrading to the latest version 6.5.1 now in used by iDempiere's ZK UI. I have done that as well as taken advantage of the more advanced UI layout feature as done by Low Heng Sin.

Note the *X Position* columns until *Same Line* at the right. They allow separate layout control for both ZK and Swing User Interface. There is also the *Quick Entry* feature as explained in http://wiki.idempiere.org/en/NF001_QuickEntry.



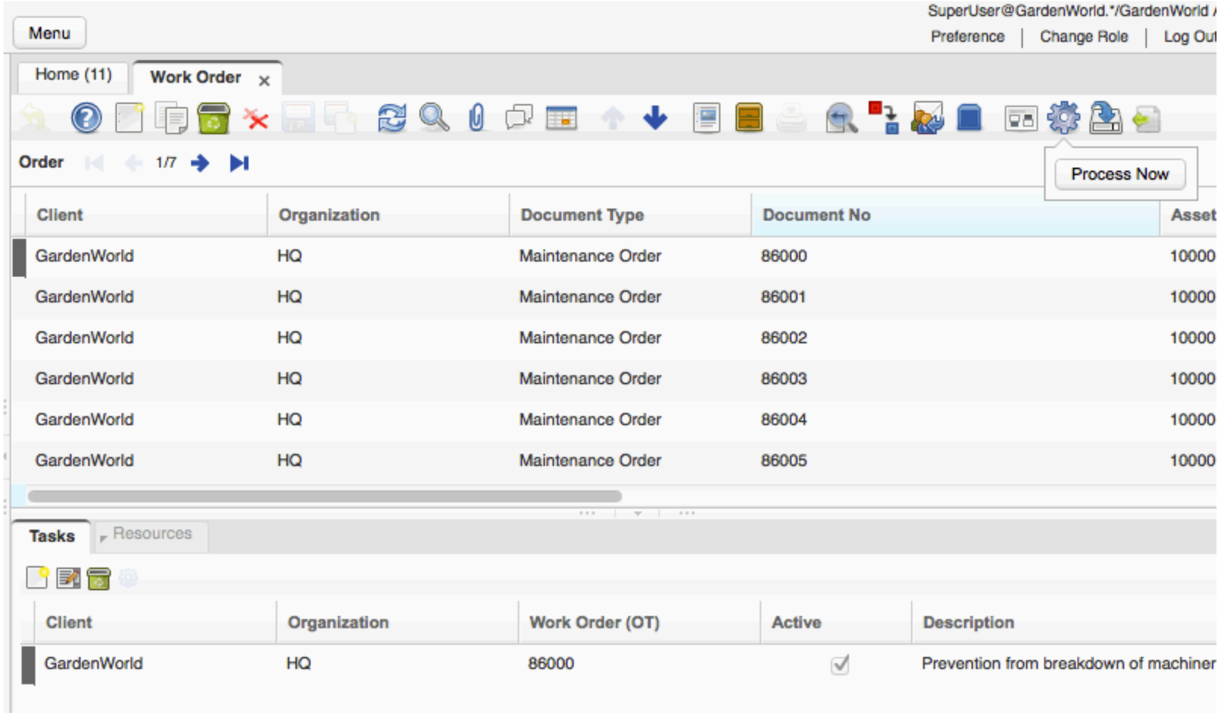
Column	Quick Entry	Sequence	Grid S...	Display Length	X Position	Column Span	Number of Lines	Same Lir
AD_Client_ID_Client	□	10		10	1	1	1	□
AD_Org_ID_Organization	□	20		10	3	1	1	☑
C_DocType_ID_Document Type	□	30		10	1	1	1	□
DocumentNo_Document No	□	40		30	3	1	1	☑
A_Asset_ID_Asset	□	50		10	1	1	1	□
DateTrx_Transaction Date	□	60		13	3	1	1	☑
AD_User_ID_User/Contact	□	70		10	1	1	1	□
IsActive_Active	□	80		1	4	1	1	☑
Description_Description	□	90		255	1	3	2	□
MP_Maintain_ID_Maintenance	□	100		10	1	1	1	□
MP_OT_Request_ID_Work Order (OT) Request	□	110		10	3	1	1	☑
DocStatus_Document Status	□	120		2	1	1	1	□
Processing_Process Now	□	130		1	1	1	1	☑
Processed_Processed	□	140		1	4	1	1	□

As this module has some labels that are crude, the easy way to modify them across many tables for the same column is done via its System Element field. Under the Table-Column window, zoom into its System Element window and defined its name according to the way you want it and it will be imposed on Table and Window during definition.

Been OSGi, the Eclipse IDE is able to debug ZK Ajax code easily similar to how Swing Java is debugged from Compiere days. There is no need to define remote-debug where the Web UI code has to be compiled beforehand. Not only that, while still in break stoppage mode, any change to source-code usually gets picked up without stopping and restarting the application.

Ajax Process Button

Heng Sin also introduced a better way to display and operate the process button commonly used in our application. As shown below, it is now standardised to be on the top right side of toolbar and can be used without finding it within the column. Just right click on gear icon and its defined process appears for selection. This makes the process button more accessible and the whole layout more simpler to navigate. The master-detail layout as shown is more true as compared to last version in Adempiere.



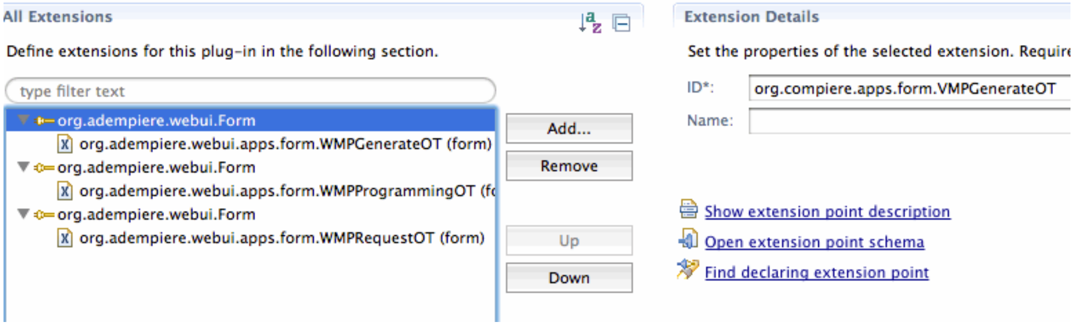
SuperUser@GardenWorld.*/GardenWorld
Preference | Change Role | Log Out
Menu
Home (11)
Work Order
Order 1/7
Process Now
Tasks
Resources

Client	Organization	Document Type	Document No	Asset
GardenWorld	HQ	Maintenance Order	86000	10000
GardenWorld	HQ	Maintenance Order	86001	10000
GardenWorld	HQ	Maintenance Order	86002	10000
GardenWorld	HQ	Maintenance Order	86003	10000
GardenWorld	HQ	Maintenance Order	86004	10000
GardenWorld	HQ	Maintenance Order	86005	10000

Client	Organization	Work Order (OT)	Active	Description
GardenWorld	HQ	86000	✓	Prevention from breakdown of machiner

WebUI Form Extension

In Adempiere legacy, the different UI code for Forms has different namespace. But for extensions to work they are both using the same Class ID as shown here.



All Extensions

Define extensions for this plug-in in the following section.

type filter text

```
▼ org.adempiere.webui.Form
    org.adempiere.webui.apps.form.WMPGenerateOT (form)
▼ org.adempiere.webui.Form
    org.adempiere.webui.apps.form.WMPProgrammingOT (fo
▼ org.adempiere.webui.Form
    org.adempiere.webui.apps.form.WMPRequestOT (form)
```

Extension Details

Set the properties of the selected extension. Require

ID*: org.compiere.apps.form.VMPGenerateOT

Name:

Show extension point description
Open extension point schema
Find declaring extension point

On the left panel you can see the class-name as org.adempiere.webui.apps.form.WMPGenerateOT but its ID on the right panel is declared as org.compiere.apps.form.VMPGenerateOT which is the same as its Swing UI counterpart. You can retain different classes, but you define the same form extensions in the plugin.xml for both UI platforms.

Prognosis Logic

For MP module, its heaviest logic is in the Prognosis process which is contained in **MPAssetPlanning**. Below is the logic for handling Calendar type maintenance jobs which then create Prognosis records.

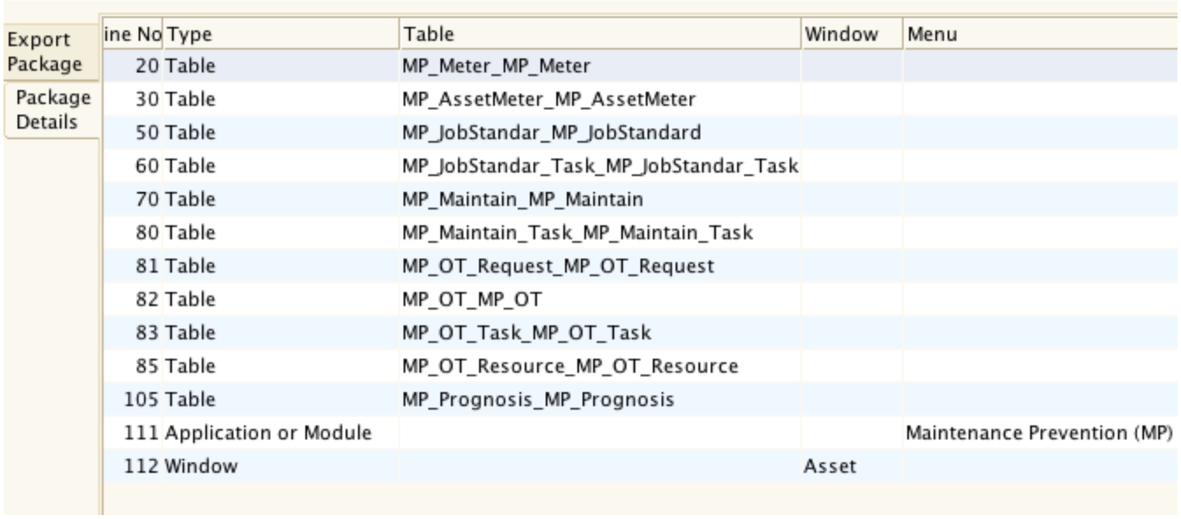
```
//searching by calendar
              StringBuffer byCalendar = new StringBuffer();
              byCalendar.append("SELECT 
MA.AD_CLIENT_ID,MA.AD_ORG_ID,MA.MP_MAINTAIN_ID,MA.DESCRIPTION,MA.PROGRAMMINGTYPE,
A.A_ASSET_ID,DATENEXTRUN ,MA.DATELASTRUN,MA.INTERVAL");
              byCalendar.append(" FROM MP_MAINTAIN MA");
              byCalendar.append(" INNER JOIN A_ASSET A ON (MA.A_ASSET_ID=A.A_ASSET_ID OR 
A.A_ASSET_Group_ID=MA.A_ASSET_Group_ID)");
              byCalendar.append(" WHERE PROGRAMMINGTYPE='C' AND DATENEXTRUN BETWEEN 
SYSDATE-60 AND SYSDATE+(7*"+ periodNo+")");
              byCalendar.append(" AND MA.DOCSTATUS<>'IT' Order by DATENEXTRUN asc");


              PreparedStatement pstmt = null;
              try
              {
                     pstmt = DB.prepareStatement(byCalendar.toString(), get_TrxName());
                     ResultSet rs = pstmt.executeQuery();
                     while (rs.next())
                     {
                            Calendar datefrom = Calendar.getInstance();
                            Calendar dateto = Calendar.getInstance();
                            int diaA=0, diaB=7,period=1;
                            int ciclo=0;

                            while(period<=periodNo){

                             Timestamp currentDate=rs.getTimestamp("DATENEXTRUN");
                             datefrom.add(Calendar.DATE, diaA*period);
                             dateto.add(Calendar.DATE, diaA*diaB);

                            if(currentDate.compareTo(datefrom.getTime())>=0 && 
currentDate.compareTo(dateto.getTime())<=0)
                          ciclo=period;

                            period++;
                            }


                            MMPPrognosis mp=new MMPPrognosis(getCtx(), 0, get_TrxName());
                             mp.setA_Asset_ID(rs.getInt("A_ASSET_ID"));
                          mp.setAD_Org_ID(rs.getInt("AD_ORG_ID"));
                          mp.setAD_PInstance_ID(p_PInstance_ID);
                          mp.setciclo(ciclo);
                          mp.setDateTrx(rs.getTimestamp("DATENEXTRUN"));
                          mp.setDescription("By Calendar :"+rs.getString("DATENEXTRUN"));
                          mp.setMP_Maintain_ID(rs.getInt("MP_MAINTAIN_ID"));
                          mp.setProgrammingType(rs.getString("PROGRAMMINGTYPE"));
                          mp.setselected(false);
                          mp.saveEx();
                          counter++;
                     }
```

The following is the logic for Meter type.

```
//searching by meter
             StringBuffer byMeter = new StringBuffer();
                           byMeter.append("SELECT
MA.AD_CLIENT_ID,MA.AD_ORG_ID,MA.MP_MAINTAIN_ID,MA.DESCRIPTION,MA.PROGRAMMINGTYPE,
A.A_ASSET_ID,MA.MP_METER_ID, MA.RANGE, MA.NEXTMP");
             byMeter.append(", (select MIN(log2.DateTrx) from MP_AssetMeter_Log log2
where  ME.MP_ASSETMETER_ID=log2.MP_ASSETMETER_ID");
             byMeter.append(" and log2.DateTrx BETWEEN (SysDate-40) AND SysDate)as
FirstDay, ");
             byMeter.append(" (select MAX(log2.DateTrx) from MP_AssetMeter_Log log2
where  ME.MP_ASSETMETER_ID=log2.MP_ASSETMETER_ID");
             byMeter.append(" and log2.DateTrx BETWEEN (SysDate-40) AND SysDate) as
LastDay, ");
             byMeter.append(" Count(1)as Qty,ME.MP_ASSETMETER_ID,MS.Name");
             byMeter.append(" FROM MP_MAINTAIN MA");
             byMeter.append(" INNER JOIN A_ASSET A ON (MA.A_ASSET_ID=A.A_ASSET_ID OR
A.A_ASSET_Group_ID=MA.A_ASSET_Group_ID)");
             byMeter.append(" INNER JOIN MP_METER MS ON
(MA.MP_METER_ID=MS.MP_METER_ID)");
             byMeter.append(" INNER JOIN MP_ASSETMETER ME ON
(MS.MP_METER_ID=ME.MP_METER_ID and a.a_asset_id= me.a_asset_id)");
             byMeter.append(" INNER JOIN MP_AssetMeter_Log MLOG ON
(ME.MP_ASSETMETER_ID=MLOG.MP_ASSETMETER_ID)");
             byMeter.append(" WHERE PROGRAMMINGTYPE='M' AND MA.DOCSTATUS<>'IT'");
             byMeter.append(" AND Mlog.DateTrx BETWEEN (SysDate-40) AND SysDate");
             byMeter.append(" Group by
MA.AD_CLIENT_ID,MA.AD_ORG_ID,MA.MP_MAINTAIN_ID,MA.DESCRIPTION,MA.PROGRAMMINGTYPE,
A.A_ASSET_ID,MA.MP_METER_ID,MA.RANGE, MA.NEXTMP,ME.MP_ASSETMETER_ID,MS.Name");

             pstmt = null;
             try
             {
                    pstmt = DB.prepareStatement(byMeter.toString(), get_TrxName());
                    ResultSet rs = pstmt.executeQuery();
                    while (rs.next())
                    {

                           int cumple=0;
                           float lastM=0,firstM=0,prom=0;
                     if(rs.getInt("Qty")>1){
                           log.info("ReadingLastM");
                           Object params[]=new
Object[]{rs.getInt("A_ASSET_ID"),rs.getInt("MP_Meter_ID"),rs.getTimestamp("LastDay")};


log.info("ReadingFirstM:"+rs.getInt("A_ASSET_ID")+"-"+rs.getInt("MP_Meter_ID")+"-"+
rs.getTimestamp("LastDay"));
                           lastM=DB.getSQLValueBD(get_TrxName(), " SELECT
mlog.CurrentAmt FROM MP_AssetMeter_Log mlog"+
                         " Inner join MP_AssetMeter met on
(mlog.MP_AssetMeter_ID=met.MP_AssetMeter_ID)"+
                         " WHERE met.A_ASSET_ID=? AND met.MP_Meter_ID=? AND 
mlog.DATETRX=? ",params ).floatValue();


log.info("ReadingFirstM:"+rs.getInt("A_ASSET_ID")+"-"+rs.getInt("MP_Meter_ID")+"-"+
rs.getTimestamp("FIRSTDAY"));

                           params=new 
Object[]{rs.getInt("A_ASSET_ID"),rs.getInt("MP_Meter_ID"),rs.getTimestamp("FIRSTDAY")}
;
                           firstM=DB.getSQLValueBD(get_TrxName(),"SELECT 
mlog.CurrentAmt "+
                                 " FROM MP_AssetMeter_Log mlog"+
                                 " Inner join MP_AssetMeter met on 
(mlog.MP_AssetMeter_ID=met.MP_AssetMeter_ID)"+
                                 " WHERE met.A_ASSET_ID=? AND met.MP_Meter_ID=? AND 
mlog.DATETRX=?",params ).floatValue();

                           float 
days=Math.abs(rs.getTimestamp("LastDay").getTime()-rs.getTimestamp("FIRSTDAY").getTime
());
                           prom=(lastM-firstM) / (days/(60*60*24*1000));

                           StringBuffer update=new StringBuffer();
                           update.append("UPDATE MP_MAINTAIN ");
                           update.append(" SET PROMUSE="+prom);
                           update.append(" WHERE 
MP_MAINTAIN_ID="+rs.getInt("MP_MAINTAIN_ID"));
                           DB.executeUpdate(update.toString(), get_TrxName());

                         int cont=0;
                         cumple=0;
                         Calendar dateExe=Calendar.getInstance();
                         dateExe.setTime(rs.getTimestamp("LastDay"));
                         //FROM HERE TO SEE IF THE NEXT DAY (7 * PREIODICAL)
                         log.info("Meter Loop");
                         do{
                             if((lastM >= (rs.getFloat("NEXTMP")-rs.getFloat("RANGE"))
&& lastM<= (rs.getFloat("NEXTMP")+rs.getFloat("RANGE")) )
                                        || (lastM>rs.getFloat("NEXTMP")) ){

                                 cumple=1;
                                 dateExe.add(Calendar.DATE,cont);

                             }
                                 else{
                                        lastM=lastM + prom;
                                 cont++;
                                 }

                         } while( cumple==0 && cont!=(10*periodNo));
                         Timestamp currentDate=new 
Timestamp(dateExe.getTimeInMillis());
                         if(cumple==1 || lastM>rs.getFloat("NEXTMP")){
                                int ciclo=0,period=1;
                                Calendar datefrom = Calendar.getInstance();
                                        Calendar dateto = Calendar.getInstance();
                                while(period<=periodNo){

                                  int diaA=0, diaB=7;
                                                datefrom.add(Calendar.DATE, diaA*pe-
riod);
                                                dateto.add(Calendar.DATE, diaA*diaB);

if(currentDate.compareTo(datefrom.getTime())>=0 &&
currentDate.compareTo(dateto.getTime())<=0)
                                             ciclo=period;
                                         period++;
                                        }
                                log.info("Insertmeter");

       MMPPrognosis mp=new MMPPrognosis(getCtx(), 0, get_TrxName());
                                mp.setA_Asset_ID(rs.getInt("A_ASSET_ID"));
                                mp.setAD_Org_ID(rs.getInt("AD_ORG_ID"));
                                mp.setAD_PInstance_ID(p_PInstance_ID);
                                mp.setciclo(ciclo);
                                mp.setDateTrx(currentDate);
                                mp.setDescription((ciclo==0?"By Meter behind schedule
":"By Meter on schedule")+rs.getString("name")+":"+lastM);
                                mp.setMP_Maintain_ID(rs.getInt("MP_MAINTAIN_ID"));

mp.setProgrammingType(rs.getString("PROGRAMMINGTYPE"));
                                mp.setselected(false);
                                mp.saveEx();
                                counter++;
                         }
```

Pack Out Your Work

To take advantage of the power now inherent in iDempiere, you can package out your work easily without the use of migration script which is unwieldy for the end-user to handle.



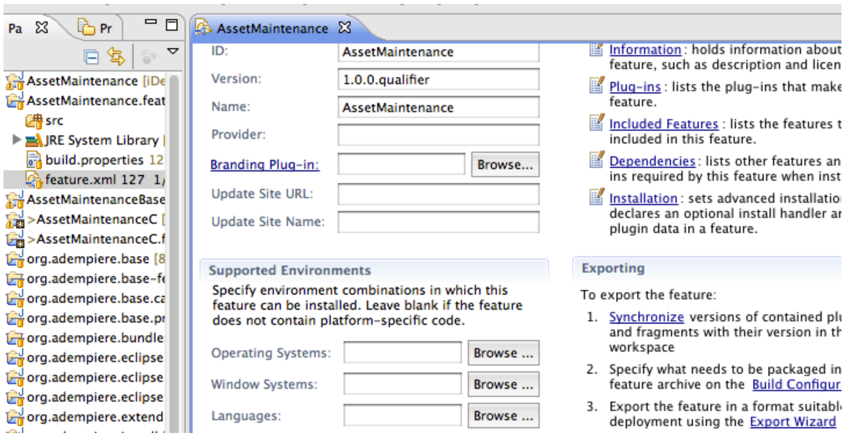
ine No	Type	Table	Window	Menu
20	Table	MP_Meter_MP_Meter		
30	Table	MP_AssetMeter_MP_AssetMeter		
50	Table	MP_JobStandar_MP_JobStandard		
60	Table	MP_JobStandar_Task_MP_JobStandar_Task		
70	Table	MP_Maintain_MP_Maintain		
80	Table	MP_Maintain_Task_MP_Maintain_Task		
81	Table	MP_OT_Request_MP_OT_Request		
82	Table	MP_OT_MP_OT		
83	Table	MP_OT_Task_MP_OT_Task		
85	Table	MP_OT_Resource_MP_OT_Resource		
105	Table	MP_Prognosis_MP_Prognosis		
111	Application or Module			Maintenance Prevention (MP)
112	Window		Asset	

Follow strictly the arrangement of tables as there is dependency constraint due to AD_Ref_Table_Display that may call a column from a prior table that has to be there. Some days before such an issue, there was no need for the extra tables and just a single Application Module was all there is! I tried to debug in the code but it was too complex to rearrange the writing of the XML lines to ensure such dependencies are dealt with in right order. The above arrangement is also more future proof as tables are spelt out clearly first.

You can test the 2Pack output by restoring back a clean data dump in your iDempiere and Pack In your 2Pack. If there is an error, examine the log and error stack which will tell you what is the dependency constraint. For deployment to the MANIFEST folder of your plugin to work it must work at one go. Otherwise you can ask your users to deploy the 2Pack the conventional way where running double time may help resolve the dependency.

Deploying to P2 Site

For the feature, open up feature.xml and from the editor panel in Eclipse click on Export Wizard.



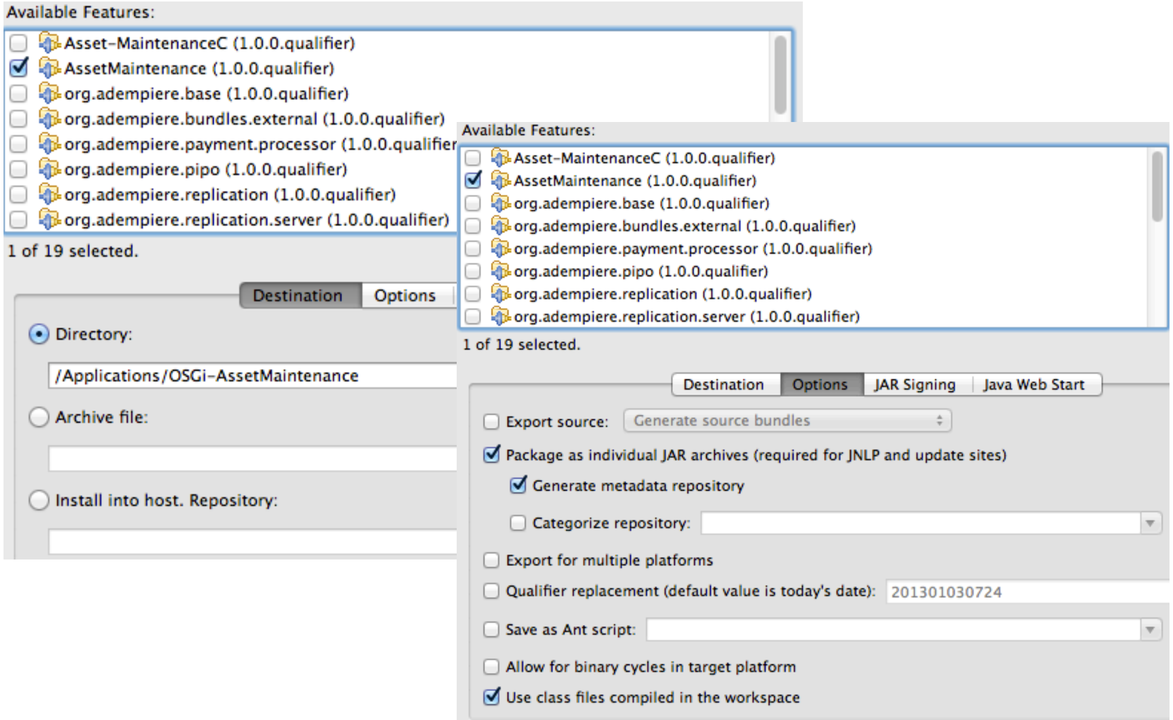
ID: AssetMaintenance
Version: 1.0.0.qualifier
Name: AssetMaintenance
Provider:
Branding Plug-in:
Update Site URL:
Update Site Name:

Information: holds information about feature, such as description and licen
Plug-ins: lists the plug-ins that make feature.
Included Features: lists the features t included in this feature.
Dependencies: lists other features an ins required by this feature when inst
Installation: sets advanced installatio declares an optional install handler ar plugin data in a feature.

Supported Environments
Specify environment combinations in which this feature can be installed. Leave blank if the feature does not contain platform-specific code.
Operating Systems:
Window Systems:
Languages:

Exporting
To export the feature:
1. Synchronize versions of contained plu and fragments with their version in th workspace
2. Specify what needs to be packaged in feature archive on the Build Configur
3. Export the feature in a format suitabl deployment using the Export Wizard

Specify your output folder and click on Finish. The resulting output is a P2 site format. Corelate that with the installer script given early on and also pasted right below. Then modify the script accordingly and deploy that to your end user to get to your p2 site which you can upload anywhere on the Internet.



Available Features:
- □ Asset-MaintenanceC (1.0.0.qualifier)
- ☑ AssetMaintenance (1.0.0.qualifier)
- □ org.adempiere.base (1.0.0.qualifier)
- □ org.adempiere.bundles.external (1.0.0.qualifier)
- □ org.adempiere.payment.processor (1.0.0.qualifier
- □ org.adempiere.pipo (1.0.0.qualifier)
- □ org.adempiere.replication (1.0.0.qualifier)
- □ org.adempiere.replication.server (1.0.0.qualifier)

1 of 19 selected.

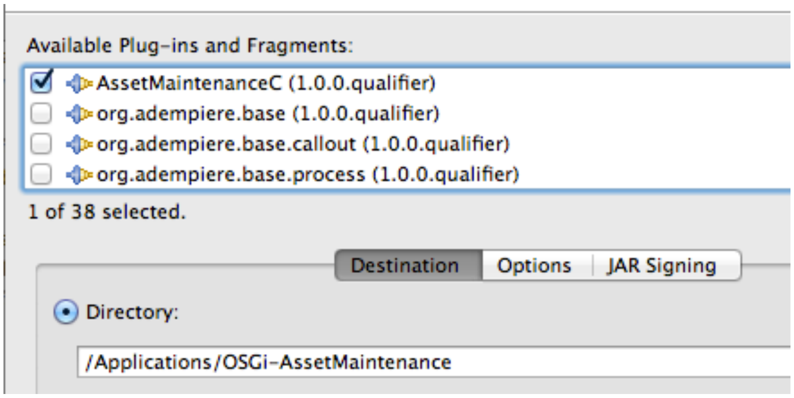
● Directory: /Applications/OSGi-AssetMaintenance
○ Archive file:
○ Install into host. Repository:

Available Features:
- □ Asset-MaintenanceC (1.0.0.qualifier)
- ☑ AssetMaintenance (1.0.0.qualifier)
- □ org.adempiere.base (1.0.0.qualifier)
- □ org.adempiere.bundles.external (1.0.0.qualifier)
- □ org.adempiere.payment.processor (1.0.0.qualifier)
- □ org.adempiere.pipo (1.0.0.qualifier)
- □ org.adempiere.replication (1.0.0.qualifier)
- □ org.adempiere.replication.server (1.0.0.qualifier)

1 of 19 selected.

□ Export source: Generate source bundles
☑ Package as individual JAR archives (required for JNLP and update sites)
☑ Generate metadata repository
□ Categorize repository:
□ Export for multiple platforms
□ Qualifier replacement (default value is today's date): 201301030724
□ Save as Ant script:
□ Allow for binary cycles in target platform
☑ Use class files compiled in the workspace

For the Swing client, open up the **AssetMaintenanceC** project, and from its MANIFEST panel in Eclipse, click on **Export Wizard**. But it will only deploy its single jar. That has to be installed together with the base jar as shown early on.


Available Plug-ins and Fragments:

- ☑ AssetMaintenanceC (1.0.0.qualifier)
- □ org.adempiere.base (1.0.0.qualifier)
- □ org.adempiere.base.callout (1.0.0.qualifier)
- □ org.adempiere.base.process (1.0.0.qualifier)

1 of 38 selected.

● Directory:

/Applications/OSGi-AssetMaintenance

Installer Script

Modify from the script here (thanks to Heng Sin for the sample reference which he did for updating iDempiere)

```
#!/bin/sh

#

cd $(dirname "${0}")

DESTINATION=$(pwd)

java -Dosgi.noShutdown=false -Dosgi.compatibility.bootdelegation=true -Dosgi.install.area=director -jar
plugins/org.eclipse.osgi_3.7.*.jar -application org.eclipse.equinox.p2.director -consoleLog -profileProperties
org.eclipse.update.install.features=true -destination $DESTINATION -repository
http://downloads.sourceforge.net/project/red1/p2/Maintenance/ -u AssetMaintenance.feature.group

java -Dosgi.noShutdown=false -Dosgi.compatibility.bootdelegation=true -Dosgi.install.area=director -jar
plugins/org.eclipse.osgi_3.7.*.jar -application org.eclipse.equinox.p2.director -consoleLog -profileProperties
org.eclipse.update.install.features=true -destination $DESTINATION -repository
http://downloads.sourceforge.net/project/red1/p2/Maintenance/ -i AssetMaintenance.feature.group
```