Installation and Dev Setup

Prerequisites

- MySql installed
- Tomcat 6 installed
- Eclipse along with the WPT if you want to use eclipse and the built-in server, or use IntilliJ or a vanilla text editor, ant and version control to build and deploy instead
- Java 8 (should work with Java 6 or 7 too)

First steps

1. Checkout the harvest_manager_code at http://subversion.ucar.edu/DLS/harvest_manager_project_tree/ (during development stage the branch we are using new_ingest_code/). In the future it will be trunk.
2. Create 3 new schemas in your MySql database. you can also grab these from dev or prod if you want to start with data. See DB Schemas section for the create statements and schema names
3. Create a folder somewhere on your file system that you want to house the harvest files. Remember that path, you will need later on during deployment
4. Either follow the steps labeled Deploy for Basic Tomcat or Deploy for Eclipse Tomcat. Word of caution getting this to run in eclipse is somewhat more difficult and requires quite a bit a monkeying around with the the eclipse files.

Deploy for Eclipse Tomcat

1. Within eclipse go to File->New->Java project. For the source directory browse to harvest-manager-project so it imports the code into eclipse. Note you want the project not the tree. For consistency throughout these instructions it will be called harvest-manager-project
2. Doing this will create a .classpath and .project files. To view these files in eclipse you will have to unselect the filter that hides hidden files.It is found in the navigator bar, click on the down arrow and then filters. Then unselect .* files.
3. Make sure that your .project is exactly like this .project
4. Make sure that you add these lines into the.classpath file. added_lines.txt
5. Extract this zip folder and put the .settings into you project foldler settings.zip
6. If you installed WPT, you should be able to go File->new ->other / Server Server and select tomcat. Follow the instructions and point the path to tomcat to the installation that you have on your box.
7. (Not sure if this is necessary but I had to do it to make it work correctly) Create a folder somewhere on your C drive, call it ie tomcat_deploys. Within the Servers tab(Should be a tab that is next to Problems and Console) double click on Tomcat. Should see an overview . For server location select Use custom location and browse to the folder you created.
8. While your in this window. Click over to the modules section of the config and add a web module. Select harvest-manager-project, and make the path /hm
9. Within the navigator browser in eclipse find the Servers project folder. Then expand the directory Tomcat v...(depends which version you have). Open up server.xml and search for your project name. harveset-manager-project. That should find it as a context element. Right below there add these lines
   <Context path="/records" docBase="$path that you created in first steps, step 3" debug="0" reloadable="false">
   </Context> . These enables your to his the url /records and view all files listed in docBase, which we need for the View records functionality of Harvest Manager
10. If its not already present within the context element harveset-manager-project add attribute crossContext="true"
11. Now open up web.xml in the same folder. Within the element <servlet><servlet-name>default</servlet-name> a few things need to change. There should be a <init-param called listings. Make the value true.
12. Also below that init-param add a new one
    <init-param>
    <param-name>globalXsltFile</param-name>
    <param-value>$(path to harvest-manager-project\web\records_tab_directory_listing.xsl</param-value>
    </init-param>
13. Now to the settings part. I could not get eclipse server to load any .xml files that would override the web.xml so I just change harvest-manager-project/web/WEB-INF/web.xml to the settings that I want. The ones you will have to change are listed in the Configuration Parameters section below
14. Right click on the server and start it. The path is http://localhost:8080/hm as long as you made a point of changing the path when configuring the server per step 8 and 9.

Deploy for Basic Tomcat

These instructions are taken from the project config files at harvest-manager-project/web/doc/INSTALL.TXT and BUILD_INSTRUCTIONS.txt.

1. Create a new file in the parent folder of Harvest-manager-project called build.properties. Change the catalina.home to the path to your installation of tomcat.
2. Make sure your path has JAVA_HOME pointing to a Java JDK. You might have to export Variable this in your bashrc
3. navigate in command line to your harvest-manager-project and type ant deploy. This should compile the project and deploy it to your catalina.home directory
4. Navigate to your tomcat folder on your file system. Then /conf. Open up server.xml and add these lines right before the end tag </Host>
   <Context path="/records" docBase="$path to your harvested files from line 3 in first steps" reloadable="false" debug="0">
   </Context>

5. Also in server xml add the attribute crossContext="true" to the context path for /hm. So that context should look like
   <Context path="/hm" docBase="/hm" debug="0" reloadable="true" crossContext="true">
   </Context>
6. Copy the file $tomcat/webapps/hm/records_tab_directory_listing.xsl to the conf directory $tomcat/conf/records_tab_directory_listing.xsl
7. Open up default conf/web.xml (in the same folder server.xml is in). For the first servlet element. It contains a init-param called listings, change the value to be true
8. Add another init param below it
   <init-param>
   <param-name>globalXsltFile</param-name>
   <param-value>$(path to harvest manager project\conf\records_tab_directory_listing.xsl</param-value>
   </init-param>
9. Place this file hm.xml in your tomcat folder in /conf/Catalina/localhost. This file will override the parameters in web.xml
10. Change the the parameters in hm.xml to ones that reflect your system, see the section Configuration parameters for the list of parameters that need to be changed.

Configuration Parameters

For both types of deployments you must change some parameters either in the harvest-manager-project web.xml. Or the hm.xml if you are doing the basic tomcat deploy. These are the parameters that muse be changed.

- harvestDbUrl - point this to your MySQL database. Will probably be just jdbc:mysql://localhost
- harvestDbUser - the user that has privileges to write to the schema
- harvestDbPwd - the password for the user
- ingestorBaseFilePathStorage - $"absolute path that you created in first steps, step 3. You also put the same path in the server.xml
- ingestorConfigsURI - Depending if there is a URL somewhere that is hosting these, The default in web.xml will work. So you don't have to change this
- mailServer, toEmails, mailType. If you have mail server running on your machine use that otherwise set mailServer to dev. Then go to harvestManager and put your external exchange server information into src\org\nsdl\harvestmanager\HarvestManager.java method sendDevEmail

These are optional but worth noting

- harvestTriggerFileDir and harvestManagerPersistentDataDir. These can be specified but by default the locations will be in the WEB-INF folder which works great because you don't have to do anything. But if you want to make sure that your trigger files are maintained you might want to point it to a place outside the deployed directory. Just in case you remove the project and deploy it again on tomcat from scratch
- harvestManagerUrl - This URL is used for nothing besides being used within the emails. This is to make sure that the email links point to the harvest manager instance that created them. If so desired make sure this URL is the one that points to your instance when tomcat is running
- handleServiceURL - This points to the handle service that you want to use to retrieve the metadatahandle and resourcehandle from. Note for either one, your ingest process will only be able to retrieve ones that are currently created. If its a new record and doesn't have handles yet, you will get record errors for those records because your local IP is not listed as a trusted IP for creating handles.The options are
  - http://nsdldev.org/hrs/service.do - for the dev handle service
  - http://nsdl.org/hrs/service.do - for the prod handle service

DB Schemas

hm_repository

create schema called hm_repository and run these three commands (note: previous versions used the name 'nsdl_repository' for this schema)

1. CREATE TABLE `metadata` (
   `metadatahandle` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
   `setspec` varchar(50) COLLATE utf8_unicode_ci NOT NULL,
   `partnerid` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
   `nativeformat` varchar(50) COLLATE utf8_unicode_ci NOT NULL,
   `native_xml` blob NOT NULL,
   `sessionid` varchar(70) COLLATE utf8_unicode_ci NOT NULL,
   `agent` varchar(23) COLLATE utf8_unicode_ci NOT NULL,
   `created_date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
   `target_xml` blob NOT NULL,
   `targetformat` varchar(45) COLLATE utf8_unicode_ci NOT NULL,
   PRIMARY KEY (`metadatahandle`,`sessionid`),
   KEY `setspec` (`setspec`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci COMMENT='nsdl harvest ingest data'
2. CREATE TABLE `resource` (
   `resourcehandle` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
   `setspec` varchar(128) COLLATE utf8_unicode_ci NOT NULL,
   `metadatahandle` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
   `URL` varchar(4096) COLLATE utf8_unicode_ci NOT NULL,
   `sessionid` varchar(100) COLLATE utf8_unicode_ci NOT NULL,

```
  `agent` varchar(100) COLLATE utf8_unicode_ci NOT NULL,
  `created_date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (`resourcehandle`,`setspec`,`metadatahandle`,`sessionid`),
  KEY `setspec` (`setspec`),
  KEY `metadatahandle` (`metadatahandle`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
```

3.
```
CREATE TABLE `session` (
  `setspec` varchar(128) COLLATE utf8_unicode_ci NOT NULL,
  `sessionid` varchar(100) COLLATE utf8_unicode_ci NOT NULL,
  `last_update_ts` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (`setspec`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
```

harvest_workspace

create new schema called harvest_workspace. This schema will have no default tables. Tables are created and deleted on the fly during the ingest process

harvest

create new schema called harvest and run 2 create table statements

```
CREATE TABLE `ingest_log` (
  `log_id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `uuid` varchar(50) NOT NULL DEFAULT '',
  `program_name` varchar(50) NOT NULL DEFAULT '',
  `message_context` varchar(100) DEFAULT NULL,
  `message_id` varchar(50) DEFAULT NULL,
  `message_level` varchar(100) DEFAULT '50',
  `log_timestamp` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `message_text` text,
  `records_in` int(11) unsigned DEFAULT '0',
  `records_out` int(11) unsigned DEFAULT '0',
  `notification_required` tinyint(1) DEFAULT '0',
  `notification_completed` datetime DEFAULT '0000-00-00 00:00:00',
  `harvest_timestamp` varchar(50) DEFAULT '[NULL]',
  `role` varchar(50) NOT NULL DEFAULT '',
  PRIMARY KEY (`log_id`),
  KEY `uuid` (`uuid`)
) ENGINE=MyISAM AUTO_INCREMENT=36688247 DEFAULT CHARSET=latin1 MAX_ROWS=4294967295 AVG_ROW_LENGTH=500


delimiter $$


CREATE TABLE `harvested_records` (

  `setspec` varchar(50) collate utf8_unicode_ci NOT NULL,

  `identifier` varchar(255) collate utf8_unicode_ci NOT NULL,

  `sessionid` varchar(70) collate utf8_unicode_ci NOT NULL,

  `data` blob,

  `format` varchar(45) collate utf8_unicode_ci NOT NULL,

  `repository_record` int(11) NOT NULL,

  `url` varchar(4096) collate utf8_unicode_ci default NULL,

  `working_date` timestamp NULL default NULL,

  `id` int(11) unsigned NOT NULL auto_increment,

  PRIMARY KEY  (`id`),

  KEY `setspec_sessionid` (`setspec`,`sessionid`)

) ENGINE=InnoDB AUTO_INCREMENT=2376671 DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci COMMENT='harvested records data'
```