Newbie's Questions

Readme

This document contains 'a few' questions asked by Jiefei Cai and answered by his kind co-workers. Please note no one is responsible for the correctness of anything in the document. Feel free to correct any mistake found in this document as you read it.

Questions are listed in different sections. You can jump to a section by clicking its link.

Perl

Beanstalk

Working at Homes.com

Introduction from Fincher (for the new folks)

First, welcome aboard! It'll be a little slow at first as you get to know all the systems and how everything works together, but we'll get you involved in no time. This is obviously just a suggestion, so feel free to approach things different if you feel more comfortable.

To begin with, I think I would make sure you are familiar with git/github. We currently have a bit of holdover svn in use svn, but primarily use git/github. If you aren't set up on there yet, you can start a free account and start learning about how everything works or ping Brian about getting you an account.

The main tools I use on a daily basis are SQL Developer and RoboMongo. You will use some other tools, but those are definitely the ones used the most. If you aren't familiar with them, that is something else you can start looking at.

If you get through all that and are still itching for more, you can start looking through the codebase. You'll definitely need to have a Github account set up for you for this. You can start wherever, but I would probably start by working backwards.

Solr is our "last stop" for data. Anything on Solr is what is actually live data. So the Solr refresh process is the last step to get things pushed to solr. On GitHub, if you look in Pipeline-Solr/HC_Slprod_Refresh.pm, you will see a good portion of the relevant code to this process. The getSQL method pulls together all of the data from the various oracle db tables that we'll need, so if you aren't very familiar with oracle queries (I wasn't), then that can be helpful to look through. Most of the types of queries you will use will be represented in that set of very, very large queries. Additional parts of the code massage the data from oracle to make sure everything is representative of what should be displayed, so feel free to peruse that as well.

I think that should get you well on your way. Feel free to email any of us if you have any questions. Welcome to the team!

Business logic

Who are we?

We are the data team of Homes.com. Yeah!!! Homes.com was found by a local magazine, and then it was acquired by Dominion Enterprise.

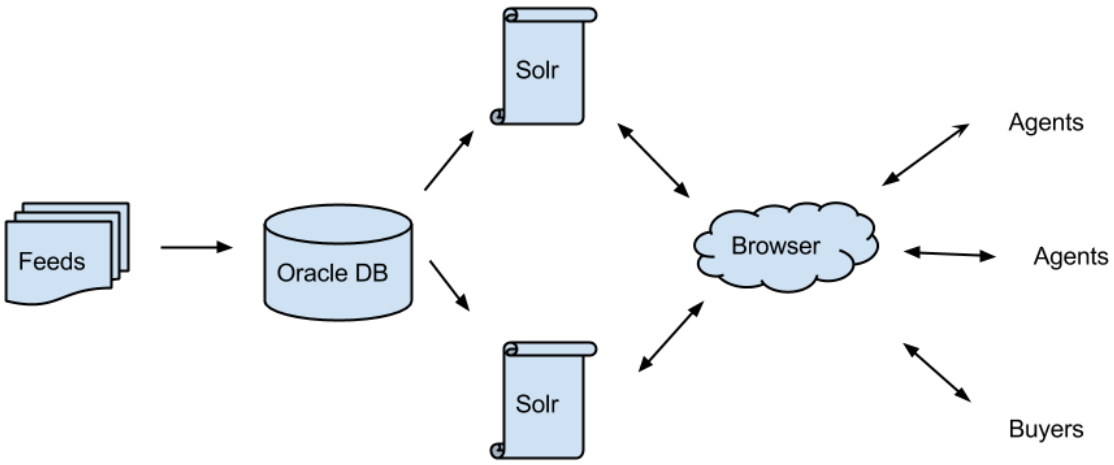
What's our duty?

To minimize the time gap between new data (feeds) becoming available and that data being available to the end user (i.e. minimize post time).

Homes.com's clients are agents, who sell properties. They pay us for posting their listing on our website. There are companies that collect this listing data from agents and put them into files so that real estate related companies like us can buy them. In Homes.com we call these files feeds. As long as agents still want to do business, they will keep generating new listings, and will keep sending new feeds to Homes.com. New feeds arrives regularly, say every 4 or 6 hours. They are XML files and the size varies, depending on how many new listing are generated during that 4 or 6 hours.

To put it simply, new coming feeds are parsed and put into our Oracle DB, our main database. It is possible (in fact, easier) for us to let agents or buyers to access Oracle DB via homes.com website. However, executing SQL statements are slow operation and it gets worse when large number of user access at the same time. To solve this problem, we use Solr, an enterprise search engine. Solr can be viewed as an open sourced version of Google search engine that makes use of inverted index.

We post listing records from Oracle DB to Solr and let end user access from Solr. Solr is a distributed search engine, which means it is deployed on multiple machines. These architecture allows more users to access our site at the same time. At 2014 there are virtual machines running as Solr server.



What is KPI?

KPI stands for key performance indicator. Currently, it is a measure of the time taken to have data posted after becoming available.

What is an agent?

Agents are people who sell properties. They are our clients we need to make them happy. These agents post their listing on Homes.com so their customers can browse.

What is an advertiser?

An “advertiser” is a historical term for an “office”, which lives in the ols_advertiser table.

What is a property? What is a listing?

A property represents a physical location, like a house. A listing represents a property for sale. In our system, property and listing is one to one relationship, they are created and deleted together, and both are required to create a Homes.com “listing”.

Who keep sending feeds to us?

We get data from Listhub/3YD, NRT, DIDX web service (Homes Connect/Boca office), MLS feeds, Broker feeds. Essentially, any listing sources that wish to be displayed on Homes.com.

What is DIDX?

DIDX stands for Dominion IDX. IDX is the Internet Data Exchange, which is a real estate search format.

But when we talk about DIDX, it usually refers to our Boca Raton office (means "mouth mouse" in Spanish), which collects data and combines them into a file and sends to us regularly. It used to be a different company. Instead of buying their data every month, we decided to buy the whole company (formerly RPIS and eNeighborhoods).

What is an aggregator?

An aggregator is a 3rd party service that combines MLS data into a standard format, for example Listhub. For us, this requires separate rules and handling as duplicates can be sent by the same supplier ID.

What is an non-aggregator?

This is a standard feed that would be delivered from 1 underlying source. For example, an MLS.

What is SmartZip?

It is a company that provides us with historical property data for approximately 90 million residential properties, including sold data. They also generate an "AVM" which is an estimated value for a home.

Who's using our data?

The Homes.com portal is the primary user of our data, which is accessed by consumers and home buyers, as well as real estate agents.

What is MLS?

Multiple Listing Service. They own and provide listing data for geographical areas. For example, the Tallahassee Board of Realtors.

What is NHC?

New Homes Channel. This is our product focused on new home builders. These are different from real estate listings, as they can be model homes, or not yet built.

What is OLS?

This is a historical term that stands for “online listing service”. It prefaces many of our database tables.

What is the pipeline?

Our data sources (DIDX webservice, Listhub) generate new data regularly. We need to 1) run loader to download new listings 2) Geocode each listing 3) run SmartZip match on each listing 4) identify duplicate listings by sorting them and 5) post listing on Solr. Pipeline is the a sequential workflow that link these five tasks together. It is achieved by using a message queue tool Beanstalk.

The programs of pipeline are deployed in

How to load a feed?

run loader on development environment, type on loaderdev12 (sudo required)

```
/code/loaders/dds/bulkload.pl -v -s -2 -c -l DEBUG > output 2> errors &
```

mlstransfer@loaderdev12:/code/loaders/dds$ /code/loaders/dds/bulkload.test.pl -v -s -2 -c -l DEBUG > output 2> errors &
[1] 1414
mlstransfer@loaderdev12:/code/loaders/dds$ tail -f output
Tue Nov 11 12:40:38 2014 - DIDX Data Loader v1.0
Tue Nov 11 12:40:38 2014 - Connecting to databases.
Connected to MongoDB: host = mongodb://mongoloaderdev.dc3.homes.com:27017
Tue Nov 11 12:40:38 2014 - Processing listing source ID -2 [Homes Connect].
Tue Nov 11 12:40:38 2014 - Creating storage schema.
Tue Nov 11 12:41:39 2014 - Retrieving source files.
Tue Nov 11 12:41:39 2014 - Checking for webservice errors.
Tue Nov 11 12:41:39 2014 - Initializing Beanstalk job distribution platform.
Tue Nov 11 12:41:39 2014 - Processing listing images...
Tue Nov 11 12:41:39 2014 - Finished processing listing images!
Tue Nov 11 12:41:39 2014 - Processing agents...
Tue Nov 11 12:41:39 2014 - Finished processing agents!
Tue Nov 11 12:41:39 2014 - Processing offices...
Tue Nov 11 12:41:39 2014 - Finished processing offices!
Tue Nov 11 12:41:39 2014 - Processing listing inserts and updates...
Tue Nov 11 12:41:39 2014 - Processing listing (11915194).
Tue Nov 11 12:41:39 2014 - Processing listing (2939747).
Tue Nov 11 12:41:40 2014 - Processing listing (5861125).
Tue Nov 11 12:41:40 2014 - Processing listing (20736843).
Tue Nov 11 12:41:40 2014 - Processing listing (20736845).
Tue Nov 11 12:41:41 2014 - Finished processing listing inserts and updates!
Tue Nov 11 12:41:41 2014 - Processing listing deletes...

Tue Nov 11 12:41:41 2014 - Finished processing listing deletes!
Tue Nov 11 12:41:41 2014 - Processing normalized listing features...
Tue Nov 11 12:41:41 2014 - Finished processing normalized listing features!
Tue Nov 11 12:41:41 2014 - Processing listing features...
Tue Nov 11 12:41:41 2014 - Finished processing listing features!
Tue Nov 11 12:41:41 2014 - Processing listing extras...
Tue Nov 11 12:41:41 2014 - Finished processing listing extras!
Tue Nov 11 12:41:41 2014 - Processing listing open houses...
Tue Nov 11 12:41:41 2014 - Finished processing listing open houses!

What is Homesconnect?

Homesconnect is our main product.

What is supplier_id?

What is souce_id?

what is Maponics ?

Maponics builds and defines geographic boundaries that are meaningful at the local level, such as neighborhood boundaries, ZIP Codes, and school attendance zones. Because locally relevant boundaries define where people spend their time and money, our products are critical for national companies looking to attract and retain new business.

How does the DIDX loader work?

The script bulkloader.pl calls DIDX web services and save the returned XML data into a folder on our server. It also parses the XML, produces a job for each listing, and dispatch them into beanstalk tube.

The loader uses a third-party tool XML::twig to parse the data. When parsing an XML, the tool calls several subroutines (in bulkloader.pl) named dispatch_*_msg to handle some specific tags. In the `loader_home/type/*/def.pm` module, we define which function handles which tag. For example, when the parser sees a 'Listings' tag, it calls `dispatch_listing_msg` subroutine to build a job and send to the tube.

```
dispatch_listing_msg
|- build_msg
|- if ( !(dispatch_msg($msg)) ) { … }
   |- $main::client->put( {data => freeze($msg), ..} )
```

bulkloader.pl

```
|- process( $main::rules )
   |- foreach my $key ( @main::rule_keys ) { process_file }
      |- eval {$entity_r->{data_tag}} => $entity_r->{data_coderef}
      |- if ( !($xml->safe_parsefile("...")) ) { .. }
      |- get_job_feedback
         |- msg_complete
         |- $job->delete()
      |- finish_remaining_job_feedback
      |- update_history
```

A number of gbulk_worker processes consume and execute the jobs. Their work is to get a job, unpack and convert it to the right format, and insert it into collection of MongoDB (aka the heap).

gbulk_worker

```
|- $job = $worker->reserve(60)
|- process_message($job)
|- if ( !(process_data($mongo_dbh, $type, $entity_id, ...)) ) { .. }
   |- $coll = $mongo_dbh->get_collection("${type}_fields${source}_${history}")
   |- $coll->insert($href, {safe => 1})
|- $job->delete()
```

gdidx_client.pl

```
|- foreach my $type qw(office agent listing )
   { $doc->{type} = $type;
     push @{$main::itrs{$iter_name}}, $doc; }
|- dispatch_task
```

gdidx_worker

```
|- $job = $worker->reserve(10)
|- run_listing ($job) if $job->data eq 'run_listing'
   |- run_listing_ols
|- drop_listing ($job) if $job->data eq 'drop_listing'
|- process_profile ($job) if $job->data eq 'process_profile'
|- delete_profile ($job) if $job->data eq 'delete_profile'
|- run_agent ($job) if $job->data eq 'run_agent'
|- run_office ($job) if $job->data eq 'run_office'
|- process_message
```

ols_listings.pm

```
run_listing_ols
|-  process_openhouses
```

database.pm

```
process_openhouses
```

How does the Listhub loader work?

Listhub loader is similar to the DIDX loader, lh_heap_client.pl produces jobs and lh_heap_worker consumes them.

lh_heap_client.pl

|- get_source_files
  |- get_webservice_data
    |- exec_http_request
|- get_keys_to_process
|- execute_xml_parser
  |- threads->new(parse_xml_files_thread_internals)
    |- $parser = XML2_Pull_Parser->new(..)
    |- $parser->parse_job()
  |- complete_all_jobs
    |- check_threads

lh_heap_workers calls `unpack_job_data` to handle Listhub jobs. If there is no Listhub jobs available, the worker will take jobs from DIDX tube instead. The subroutine `process_message` is the same as in gbulk_worker.

lh_heap_worker

|- $job = $worker->reserve(60)
|- unpack_job_data
  |- process_nrt_openhouse

How does the Geo coding work?

How does the SmartZip matching work?

How does the dedupe work?

How to run dedupe test cases?

Dedupe test cases are used to test our dedupe code and make sure it works as expected. There are 16 test cases in total and each of them is a listings.xml file. More detail can be found in Deduplication 2.0 Test Cases .
To run them (take test case 7 for example), you need to:

1) login to loader11.dev.homes.com with mlstransfer account
2) `cd /code/loaders/source_data/dds/-2`
3) `cp tc7/listings.xml ./`
4) run bulkload.test.pl to load -2 feed (make sure -2 feed is not being loaded by other at the moment)

5) find out listing ids from the output message of bulkload.test.pl

mlstransfer@loaderdev12:`/code/loaders/dds$ /code/loaders/dds/bulkload.test.pl -v -s -2 -c -l DEBUG > output 2> errors &`

[1] 1414

mlstransfer@loaderdev12:/code/loaders/dds$ tail -f output

Tue Nov 11 12:40:38 2014 - DIDX Data Loader v1.0

Tue Nov 11 12:40:38 2014 - Connecting to databases.

Connected to MongoDB: host = mongodb://mongoloaderdev.dc3.homes.com:27017

Tue Nov 11 12:40:38 2014 - Processing listing source ID -2 [Homes Connect].

Tue Nov 11 12:40:38 2014 - Creating storage schema.

Tue Nov 11 12:41:39 2014 - Retrieving source files.

Tue Nov 11 12:41:39 2014 - Checking for webservice errors.

Tue Nov 11 12:41:39 2014 - Initializing Beanstalk job distribution platform.

Tue Nov 11 12:41:39 2014 - Processing listing images...

Tue Nov 11 12:41:39 2014 - Finished processing listing images!

Tue Nov 11 12:41:39 2014 - Processing agents...

Tue Nov 11 12:41:39 2014 - Finished processing agents!

Tue Nov 11 12:41:39 2014 - Processing offices...

Tue Nov 11 12:41:39 2014 - Finished processing offices!

Tue Nov 11 12:41:39 2014 - Processing listing inserts and updates...

Tue Nov 11 12:41:39 2014 - Processing listing (11915194).

Tue Nov 11 12:41:39 2014 - Processing listing (2939747).

Tue Nov 11 12:41:40 2014 - Processing listing (5861125).

Tue Nov 11 12:41:40 2014 - Processing listing (20736843).

Tue Nov 11 12:41:40 2014 - Processing listing (20736845).

Tue Nov 11 12:41:41 2014 - Finished processing listing inserts and updates!

Tue Nov 11 12:41:41 2014 - Processing listing deletes...

Tue Nov 11 12:41:41 2014 - Finished processing listing deletes!

Tue Nov 11 12:41:41 2014 - Processing normalized listing features...

Tue Nov 11 12:41:41 2014 - Finished processing normalized listing features!

Tue Nov 11 12:41:41 2014 - Processing listing features...

Tue Nov 11 12:41:41 2014 - Finished processing listing features!

Tue Nov 11 12:41:41 2014 - Processing listing extras...

Tue Nov 11 12:41:41 2014 - Finished processing listing extras!

Tue Nov 11 12:41:41 2014 - Processing listing open houses...

Tue Nov 11 12:41:41 2014 - Finished processing listing open houses!

6) find out prop ids of those listings from EWS log file `/code/loaders/logs/dds/EWSsummary.log.loaderdev12`. You could use command like `tail - n 10000 EWSsummary.log.loaderdev12 | grep '2939747] Property ID'`. The result may look like:

[32143] 2014/11/11 11:05:04,280> [DEBUG] (Worker) [S: -2 Listing ID: 2939747] Property ID: 215358132 (601)

7) check beanstalk tube on `loaderdev11` to see if the loader has pushed jobs on dedupe tube.

8) run assign_szid.pl, which changes these listings' smartzip id to be the same, so they all become duplicates.
9) run `dedupe_worker.pl`
10) check Oracle database to see if the result of deduplication is correct. The query may look like: `select ol.fk_propid, ol.TRUMP_FK_PROPID, ol.TRUMP_SEQ, ol.dchange from ols_listing ol where ol.fk_propid in (178504409,178506758,188268259,155108882,215358132); -- Test case 7`

How does the Solr refresh working?

go back

Oracle Database

What version are we using?

11g

Where is Oracle DB located?

How can I access Oracle DB?

Which database do we use most often?

ols

What is Oracle PL/SQL?

This is a database programming language where the bulk of the business logic exists. Often called the "packages".

What is a parameterized query?

A parameterized query is something like this:

We use it for two reasons:
1. The overhead of compiling and optimizing the statement is incurred only once, although the statement is executed multiple times.
2. Prepared statements are resilient against SQL injection.

Tell me about tables used most often in Oracle DB?

ols_property (op) details about properties like address geo-code, smartzip id,

ols_listing (ol)
ols_mls_supplier (oms)
ols_olcp_account
portal_mh_account
portal_mh_account_ls_xref
ads_supplier_xref

Tell me about fields in ols_property (op)?

aggregator_listing_id
floorplan_id
geo_long, geo_lat
LONGITUDE, LATITUDE

Tell me about fields in ols_listing (ol)?

Tell me about fields in ols_mls_suppiler (oms)?
hc_hide

go back

MongoDB

Why do we use MongoDB while we already have Oracle DB installed?

What do we use MongoDB for?

How to connect to MongoDB using Perl?

```
#!/usr/bin/perl

use MongoDB;
use MongoDB::OID;

my $conn = MongoDB::Connection->new(
    host => "mongodb://mongodbdev01.dc3.homes.com");
```

Note that Mongodb does not require authentication for access in default.

```
my $id2 = $coll->insert({ name => 'mongo', type => 'database' }, {safe => 1});
```

What is a collection?

A collection is a data table.

How to create a collection?

You don't need to create them. If the collection specified in your query does not exist, MongoDB will create it automatically. Note that typo will not be catched because of this feature.

How to connect to mongodb with in terminal?

```
mongo --host mongodbdev01.dc3.homes.com
```

What are the basic commands in mongo shell?

```
show dbs
use homes_didx_cb_1
db.mls_updates.find()
db.mls_updates.find({fk_supplier_id : '589'})
db.mls_updates.update({fk_supplier_id : '2859'}, {$set : {disclaimer : 'test test'}})
```

Is there a mongo client end?

Yes, http://robomongo.org/. though it currently is only useful for the mongobatch mongo server. For accessing mongoheap, you will still need to use command line tools until the update supporting Mongo 3.0 has been released.

go back

Solr

What is Solr?

Solr is an open source enterprise search platform from the Apache Lucene project.

What is SolrCloud?

SolrCloud is a later version of Solr, which offers fault tolerance and high availability.

How to install and run Solr?

Installing Solr is relatively easy.
Download the distribution.
Save it in a convenient location on your file system.

Unpack/uncompress it.
Change directories to the example directory, and fire up Solr by typing `java -jar start.jar` at the command line.

How to access Solr via browser?

As long as you have not made any configuration changes, you should now be able to connect to your locally hosted Solr administrative interface through your favorite Web browser. Try, `http://localhost:8983/solr/`.

What is slprod?

This is a Solr index that contains all active listing data. It is used to power the search results and detail pages on Homes.com
slin11vld.dev.homes.com:8080/solr/slprod/select?q=*:*

What is sldir ?

This is a Solr index that contains agents and other service provider (mortgage, etc) account information. It is used for the “Local Pros” section of Homes.com as well as lead routing.
http://slin11vld.dc3.homes.com:8080/solr/sldir/select?q=*:*

What is slqa?

This is a Solr index that contains questions and answers, and powers the “Q&A” section of Homes.com

What is slszip?

This is a Solr index that contains historical listing data, provided by Smartzip Analytics. It powers the “Home Values” section of Homes.com, and is also used on the detail page.

How to **ADD** fields to Slprod schema.xml in dev and production?

This is just for adding fields. Ask others if you want to perform other operations.

Email changelog@homes.com to inform DBA about this change

check out the schema.xml from SVN (`/system/Configuration/trunk/slprod11/solr47/schema.xml`)

Update the header of schema.xml

Make changes in schema.xml locally

login to slin11vld using command like `ssh solr@slin11vld.dc3.homes.com` password required

change dir to `./4.7/slprod/conf/`

make a backup file for schema.xml before updating it. Name the backup file `schema.xml.yyyymmddhhmmss`

replace the schema.xml with your local version using scp, it should look like:
`scp ~/localpath/Configuration/trunk/slprod11/solr47/schema.xml solr@slin11vld.dc3.homes.com:4.7/slprod/conf/jiefei.xml`

Use `diff` command to compare the schema.xml with its backup to make sure that's what you want.

commit shcema.xml to SVN with comment on what was changed.

Comment out three slprod related cron jobs using `crontab -e`
(As to which three, you need to figure out yourself. When I changed it, they were the first and last two. Do not rely on the order!)

```
#0 0 * * * /home/solr/scripts/delete_propid.pl 2>&1 | /usr/bin/mail -s 'SolrDev Delete Propid' fang.lin@homes.com,bide.xu@homes.com


0 * * * * /home/solr/scripts/update_sldir.pl -ad 2>&1 | /usr/bin/mail -s 'SolrDev Sldir Update' fang.lin@homes.com,bide.xu@homes.com

* * * * * /usr/bin/curl "http://localhost:8080/solr/slqa/dataimport?command=delta-import&clean=false&optimize=false" >/dev/null 2>&1
```

```
* * * * * /usr/bin/curl
"http://localhost:8080/solr/sldir/dataimport?command=delta-import&optimize=fals
e" >/dev/null 2>&1


##solr slprod refresh processes
#*/5 0-2,4-23 * * * /home/solr/scripts/start_solr_refresh.pl 10 >
/home/solr/scripts/result1.txt 2>&1

#0,5,10,15,20,25,30,35,40 3 * * * /home/solr/scripts/start_solr_refresh.pl 10 >
/home/solr/scripts/result1.txt 2>&1
```

~~restart Tomcat, the Solr container, using `sudo service tomcat7 restart`.~~

Don’t restart Tomcat.

Uncomment those three cron jobs

Check if the changes have been taken by Solr in your browser, such as slin11vld.dc3.homes.com:8080/solr/slprod/select?q=*:*

if it's been more than 20 minutes, email systems@homes.com - there could be a problem with replication, also check the last replication time

How to update sldir index?

The sldir index is updated using the delta import feature. The schema is similar to the slprod schema so changes to it should be done in a similar manner. Don't forget to restart the tomcat server after you make changes to the schema.xml for the new field changes to take effect.

SVN Repo Folder:
http://svn.homes.com/repos2/System/Configuration/trunk/sldir11/solr47
File Path:/home/solr/4.7/sldir/conf/schema.xml
Cmd: `sudo service tomcat7 restart`

For generating the values for the fields the queries are in data-config.xml.
(Read the DELTA IMPORT section before working on it to understand the file better)
There are different types of agent information we store:non-agents, agents-non account and agents.

Make the query changes accordingly. If it is part of the select in the query or the deltaImportQuery then the corresponding field must be present in the schema.xml. The import will fail otherwise.

If the field value you are adding is dependent on a value fetched in the main query or depends on a completely different table which you don't want to join into the main query then you add it as a separate entity query below.

If this entity value is to be added for all types then you need to replicate it for all types.
Once you make the changes follow these steps to deploy the changes:
-Turn off the cron jobs on the Solr servers: especially
…
`http://localhost:8080/solr/sldir/dataimport?command=delta-import&optimize=false" >/dev/null 2>&1`
- Check if current delta import is running or no by using a browser and typing in `http://solrsrvr.dc3.homes.com:8080/solr/sldir/dataimport`
If the status is idle, it means you can proceed. Otherwise wait until the current procedure is completed.
-Restart the tomcat server ONLY if you have made changes to the schema.xml
- Reload config using
`http://solrsrvr.dc3.homes.com:8080/solr/sldir/dataimport?command=reload-config`
This should reload successfully.
-Now start a new delta import by running this in the browser
`http://solrsrvr.dc3.homes.com:8080/solr/sldir/dataimport?command=delta-import&optimize=false`
This will run the deltaImportQuery which picks up all records that have been changed since the last index run time. You can check the status of the delta import and the

number of records changed by running the
`http://solrsrvr.dc3.homes.com:8080/solr/sldir/dataimport?command=reload-config`
to monitor the status

In order to test, look up which date field of the entity that you are testing, that is being queried to look for changed records. Modify that date field `pma.date_modified` in Oracle to a future timestamp. So that when you run the delta-import that record will be picked up. Now query the index to look for changes.

What is DELTA IMPORT?

Delta import is a data import command that is provided by Solr. It can be used to do both an incremental and full import of data to Solr index. We use this feature for our **sldir** and **slqa** solr indexes.

- The *query* gives the data needed to populate fields of the Solr document in full-import
- The *deltaImportQuery* gives the data needed to populate fields when running a delta-import
- The *deltaQuery* gives the primary keys of the current entity which have changes since the last index time

Note: It takes a really long time for the full import to complete. We try to avoid that by using the rolling refresh script

How to run a long time update on Solr?

batchsrv11 /web/SHS/rolling_sldir_refresh

What is FACET QUERY?

How to interact with Solr using perl

http://search.cpan.org/~bricas/WebService-Solr-0.21/lib/WebService/Solr/Query.pm
link

go back

Beanstalk

What is Beanstalk?

A job queue

What is a tube?

A tube is essentially an organized “bin” of jobs

What is a worker?

A worker is a consumer of jobs from the queue, though workers can also generate new jobs.

How do we create a worker?

How do we put a job into a tube?

How do we check the status of a job?

Specifics can be tricky, but http://pipemon.dc3.homes.com/loader-cgi/PL/Tools/monitor.cgi will be your best friend.

go back

Perl

What version do we use?

```
perl -v
```

```
This is perl, v5.10.1 (*) built for x86_64-linux-gnu-thread-multi
Copyright 1987-2009, Larry Wall
```

How do we avoid namespace pollution?

Namespace pollutions means variable name collisions between packages. To prevent it, we tend to give modules their own namespaces and have shared resources also named appropriately.

Why some perl files have no extension?

Perl files are usually with .pl extension and Perl module are with .pm extension. But this is not required. As long as the perl file has a shebang in the first line (such as `#!/usr/bin/perl`), the OS knows where to find an interpreter to run it.

What modules do we used in our Perl code?

JSON
DBI
Beanstalk::Client
Log::Log4perl
Data::Dumper
WebService::Solr
XML::Twig

What is Data::Dumper?

It is a module from CPAN. “Given a list of scalars or reference variables, writes out their contents in perl syntax. The references can also be objects. The content of each variable is output in a single Perl statement. Handles self-referential structures correctly.”

What is DBI?

“The DBI is a database access module for the Perl programming language. It defines a set of methods, variables, and conventions that provide a consistent database interface, independent of the actual database being used.”

How to connect to Oracle DB?

see Oracle DB section.

How to connect to mongoDB?

see mongoDB section.

How to read a file line by line?

There is an example in this link

```
use strict;
use warnings;

my $file = 'SnPmaster.txt';
open my $info, $file or die "Could not open $file: $!";
while( my $line = <$info>) {
print $line; last if $. == 2;
}
close $info;
```

How do we process XML documents?

It depends. We use a command line tool xmllint to handle some easy tasks, such as formatting an XML file. `xmllint --format origin.xml > new.xml`

For heavy duty, we use XML::Twig.

How to parse XML documents with XML::Twig?

How to create an XML document with XML::Twig?

Here is an example code that creates the following XML document.

```
<root>
    <elem>
        <sup_id>2001</sup_id>
        <pk_acctid>32</pk_acctid>
    </elem>
</root>
```

```
use XML::Twig;

my $twig= new XML::Twig;
my $in_file = 'head.xml';
my $out_file = 'out.xml';
$twig->parsefile( $in_file );     # build the twig
my $root= $twig->root;            # get the root of the twig (stats)

my $eblg= new XML::Twig::Elt( 'elem', '');   # create the element
$eblg->paste( 'last_child', $root);

my $epk_sup_id= new XML::Twig::Elt( 'sup_id', 2001);
$epk_sup_id->paste( 'last_child', $eblg);

my $epk_acctid= new XML::Twig::Elt( 'pk_acctid', 32);
$epk_acctid->paste( 'last_child', $eblg);

open (my $fh_out, '>', $out_file) or die "unable to open '$out_file' for
writing: $!";
$twig->print($fh_out); # this prints to the filehandle
```

go back

MySQL

Why do we use MySQL if we already have Oracle and MongoDB?

How to access MySQL?

mysqldev.dc3.homes.com

Documents

Are there any other documents available?

Yes. Most official documents are in Google drive.

How to capture a snapshot?

If you use Mac like I do, check this link from Apple.

Press `Command (⌘)-Shift-3`. The screenshot is added to your desktop.

Press `Command (⌘)-Shift-4`, and then drag the crosshair pointer to select the area. Hold Shift, Option, or the Space bar while you drag to resize the selection area. To cancel, press Escape (esc) before you release the mouse button.

go back

Source code management

What version control software are we using?

GitHub, though there is still a bit of code waiting for migration from SVN.

go back

Work at Homes.com

What is ontimenow aka Axosoft OnTime?

OnTime is a 3rd party issue tracker and project management system. Recently is has been replaced by JIRA.

What is a ticket/issue?

A ticket or issue is represents a task to be done or a bug to be fixed. It is tracked in JIRA.

When is lunch time?

The lunch window is from 11:00 am to 2:00 pm. You can go for lunch within this time frame.

How to use a timesheet?

Available in Novatime via ask4hr.com Submitted every 2 weeks.

go back

Hardwares

What machines do we have?

Development machines

loader11.dev.homes.com
slin11vld.dev.homes.com (solr)
mongoheap.dev.homes.com
mongobatch.dev.homes.com
pipeline11.dev.homes.com (anything pipeline-related)

Production machines

loader11.dc3.homes.com
loader11.dc3.homes.com
mongoheap.dc3.homes.com
mongobatch.dc3.homes.com
solr@solr11old.dc3.homes.com
Great many pipeline servers
web@batchsrv11.dc3.homes.com
web@batchsrv13.dc3.homes.com

Pipeline machines

mlstransfer@loaderdev12.dc3.homes.com
dedupe11.dc3.homes.com

Where are these machines located?

Who's maintaining these machines?

Why I don't have write permission when login with my own account?

In loaderdev12 and batchsrc13, we need to use mlstransfer account to gain write permission. `sudo su - mlstransfer`

go back

Software

What tools are used in Homes.com?

> Beanstalk for message queue,
> Solrcloud for search engine,
> MongoDB for logging both temporary and historical,
> Oracle for main database,
> MySQL for storing information about sources of data,
> Hadoop for log file processing,
> crontab for scheduling executables,
> github for version control.

How do we run long term jobs on remote servers via terminal?

> When you execute a Unix job in the background (using &, bg command), and logout from the session, your process will get killed. You can avoid this using the nohup command. `# nohup command-with-options &`
>
> Or you can use screen.
>
> To get started, simply type `screen`.
> While in screen, type the command `Ctrl-a "` (ctrl-a followed by a double quotation mark) to see a list of open terminals.
> To rename this terminal, type `Ctrl-a A`.
> One way to create a new terminal is by typing `Ctrl-a c`.
> You can use the up and down arrow keys or j/k to move between two open terminals.
> You can also create new terminal by typing `screen -t "Terminal 3"`.
> To switch between terminals, use `Ctrl-a n` and `Ctrl-a p`.
>
> Some helpful commands

go back

Pipeline

How do we create jobs from gdidx_loader.pl given a feed file?

```
/code/loaders/dds/bulkload.pl -v -s 2411 -c -l DEBUG > output 2> errors &
```