**Notes For Ranjit for the Jproj Project**

See attached for two sketches of my current and ideal setup regarding all my servers and databases and data flow. (jdata-servers-setup.pdf)

There are two parts to the software: The page creator (in the page-creator.zip and typicalPageCreationProject.zip for a working example) and the updater software (which is in the jdata3-1.zip). I think (but not entirely sure) that you’re mainly going to need to work with the page creator software to make the changes/updates I need.  
  
Basically I have 4 main ticket websites, all of which have ticketgrids on them that make calls to a database (see data-grids.zip). We have two databases set up now, and I would like to keep it that way. Originally, two DB's were set up for redundancy and load balancing.  
  
Currently, only one database is serving all 4 sites I think.  The sites are:  
  
www.clickitticket.com (Windows, grids in asp) - the IP is 204.12.25.164

tickets.oakwebworks.com (Windows, grids in asp) - the IP is 204.12.25.164

(this machine - IP 204.12.25.164 – is the one that has the MySQL database with the nightly dump of the TicketNetwork Web Services data.

bestshowticketslasvegas.com (Windows, grids in asp) - the IP is 70.101.240.88  
authoritytickets.com (Linus, grids in php) - the IP is 205.178.136.110  
  
Please refer to the attached sketches to see the set up and data flow (jdata-servers-setup.pdf).

Please Note: The TicketDataGrids in the zipped file called data-grids.zip are renamed so that you can tell which one goes with which website. In production, they are all called TicketDataGrid.asp or TicketDataGrid.php only

-------------------------------------------------------

**A typical page making project, the steps I take:**

See files in typicalPageCreationProject.zip.

1. To create a set of pages for each city the band Boyz ii Men will be in, I use the template file called templateConcerts-concerts-BoysMen.asp.

The template pages use variables, and there are generally two ways to set any one variable. They are:

{keyword} = event name plus the word ‘Tickets’

{keyword1} or {venue} = the venue the event is playing at

{keyword2} or {city} = the city that the event is in

{keyword3} or {state} = the state the event is in

{keyword4} or {performer} = the name of the event or performer

{keyword5} = the date of the event

{keyword6} = the time of the event

{keyword7} = the URL to the Buy Now Page, ResultsTicket.asp?evtid=

{keyword8} = the abbreviated state name

{keyword9} = link to the venue

1. I set the jproj.ini properly to use this template file
2. I set the event-query.ini to set the event, which in this case is Boyz ii Men.
3. I set the preprocess.ini to ‘c’ which stands for ‘concerts’ since these are going to be concert pages I make. The other two settings for preprocess.ini is ‘t’ for theater and ‘s’ for sports.
4. In a DOS command line prompt, I type in: java jproj after making all the necessary settings in each of the .ini files. This will create pages for each city Biys ii Men will be in, and place the newly created files in the designated output file.

-------------------------------------------------------

**Random notes/thoughts:**

This is about as simple as things get: data is pulled down nightly from the TN web service (I've been using the string version) and stashed as a local copy in a mysql db.  TN has a call in its API used to populate each of the major tables: events, venues, performers, sales, and categories.  the other tables, in the new, are derived from the main ones.  In most cases a little processing is done on the data and some extra columns of derived data may be added (especially for events).  

\*Note events is actually created by using the TN call several times to break up the amount of data, it is by far the largest table. 

Jason's data is read only except for the nightly updates.  If TN does not permit you to hit them directly (apart from the nightly updates, although a hybrid approach might be okay, grids direct, page creator via local db copy created by updater)  or if performance in doing so is not acceptable i recommend continuing with a fully denormalized data format and would advocate using an in memory db if server memory permits (and still allows for a reasonable cache for cache-able queries - some aren't due to rand order keyword use).

The data is used by the page creator portion of the jproj code and the ticket grids (Jason can explain these to you) in a read only fashion.  We have a few grids, asp, php, .net/c#

You should also set up a proxy sever on the new TN approved machine so that you can proxy requests through to TN from a local dev machine and so Jason can build a local copy of the DB using the updater on a local machine of his for page creation - page creation is MUCH faster if the db is local and down right painful when hitting TN directly.  Previously I used squid in linux for this for myself.

*Ranjit, this is already done, but I need a local copy on my new machine and I need this new machine configured to allow the page creating software (jproj) to work. Right now it can’t do it*

Here are the answers to questions from another developer who looked at this:

7 - no, but one could easily be put together - look in the updater code, all of the sql is there and could be used to make the script in a nearly cut and paste fashion.  Again, nothing fancy here.

8 - open the db and count them - the old version is under jproj the new jproj3.

I think the currently used (old) version has 5 tables and 5 backup tables (dumps of the 5 from the previous day)

9 – The Mysql database size is small, 10s of MBs. Maybe 100k-200k rows tops, most of them in events.

add indices - it needs them.  You might even consider looking at the query log and

determining the common fields queried and building the indices based on that info.

I've only causally examined this... so i have no real insight at present.

10 - the site needs to stay up while the updater is running.

jason has cases where mysql and sites are together and separate. **see attached sketches to see complete set up**

authority has them together clickit separate at present.  nothing prohibits either solution though, except possibly the number of distinct machines TN will allow one to register as permitted to hit their data.

13 - yes, but it's simple primary key stuff, eg. unique event ids as provided by TN are the events' primary keys.  no foreign, etc keys.

14 - yes, the grids, the site needs to stay up unless Jason approves down time. **never any down time is possible**

18 - the current working updater is on clickit runnning win, the working db is under linux on authority and sites are under various OSes - Jason can provide more info. **again, see sketch**

19 - Current hosting (IP is 204.12.25.164) - **Cloud - 1 vCPU, and Cloud - RAM 002 GB**, **let me know if you need more info**

21 - yes, all aspects of this can be tested.  TN has a test instance you can hit to test pulling down data, and you can test all aspects of its use on a local dev setup. **So I need the updater (called jdata) to update every night the clickitticket.com/oakwebworkstickets.com mysql DB, I need it to update the authoiryttickets.com DB every night, and I need it to update a version on my PC every night so I can make pages with it.** **So you can set up an instance of it on your dev env. and test from there. Other than it updating my local machine, I think it does the updating of the two databases now already.**