道生一，一生二，二生三，三生万物。

——《道德经》

general Management Information System

*g*MIS

通用管理信息系统

Zhenxing Liu

wadelau@{ufqi, gmail, hotmail}.com

(Working with ufqi.com, ippjj.com)

Update Dec. 2016

(This page is intentionally left blank)

(This document contains three sections: A, B, C)

（本手册分为 A、B、C 三个部分）

Contents/目录

"In a demand-driven opinion, we faced increasing requests of creating enormous table-based management tools for operation teams in years of 2005-2010 at ChinaM, an affiliate of Telstra[1] in Beijing. These shared some common functions and most of them just needed to achieve basic goals (CURDLS) for a table. So we conducted many practices to find one to meet this kind of demand, for all, forever."

[1] http://www.telstra.com.au

A. 系统设计说明/Design Manual Sections

1. MIS, GWA2 and gMIS

According to OCC's MIS guidelines[2], a Management Information System (MIS) is defined as "a system or process that provides the information necessary to manage an organization effectively. MIS and the information it generates are generally considered essential components of prudent and reasonable business decisions". A MIS has these goals:

- Enhance communication among employees.
- Deliver complex material throughout the institution.
- Provide an objective system for recording and aggregating information.
- Reduce expenses related to labour-intensive manual activities.
- Support the organization's strategic goals and direction.

Regarding to general purposes, especially with MySQL as back-end, we have tools like phpMyAdmin[3] and its descendant, Chive [4](Web-based MySQL Admin Interface). These systems have three common merits:

- No-targeting user data.
- On-demand and instant deployment without 2nd-round development needed.
- Basic data and structures manipulations: **c**reate, **u**pdate, **r**etrieve, **d**elete, **l**ist and **s**earch (**CURDLS**).

General Web Application Architecture[5] (GWA2) is a framework for web applications development. Its feathers include MVC-based, core-shared, clear and clean structures, quite fast new instances deployment and easy kick-start. GWA2 uses Smarty[6] as its default template engine.

[2] http://www.occ.gov/publications/publications-by-type/comptrollers-handbook/mis.pdf
[3] http://www.phpmyadmin.net/home_page/index.php
[4] http://www.chive-project.com
[5] http://ufqi.com/dev/gwa2
[6] http://smarty.net

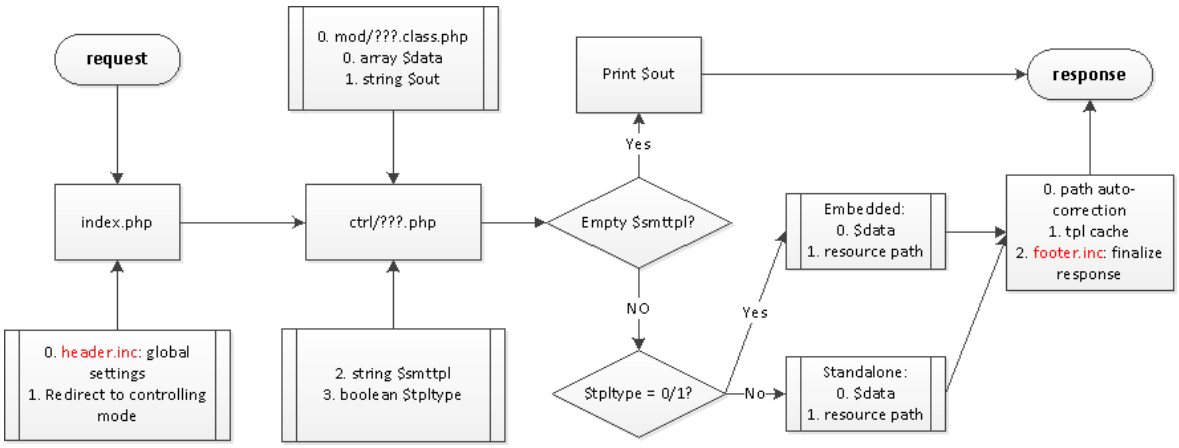

Figure 1 GWA2

*g*MIS is a new and generally-targeted kind of MIS. It is a GWA2-based web application and has following characteristics:

- For general purpose, no user data involved in codes.
- XML configurations.
- Run immediately on connection to databases.
- Ajax-enabled, i.e. GTAjax[7].
- Inline editing of user data.
- Customized input and output, e.g. WYSWYG.
- Strong searching, i.e. innovative page navigator.

We explore gMIS in next sections in detail and explain what these items represent.

2. gTbl Class

gTbl is the core component of gMIS. It locates in mod/gtbl.class.php in an instance of GWA2 point of view. gTbl stands for “general table” as this idea comes from an intention to find a tool to manage all tables without coding each function of **CURDLS** for every single table.

In a demand-driven opinion, we faced increasing requests of creating enormous table-based management tools for operation teams in years of 2005-2010 in ChinaM, an affiliate of Telstra in Beijing. These shared some common functions and most of them just needed to achieve basic goals (CURDLS) for a table. So we had conducted many practices to find one to meet this kind of demand, for all, forever.

We had ever tested phpMyAdmin and some other similar toolsets, but all of them failed to satisfy us. Then gTbl came out with the goal of general purpose, in the meantime, providing a customized

[7] http://ufqi.com/dev/gtajax

configuration for each table. Gradually gTbl evolved itself to gMIS. That is to say, it is a pre-condition that a phpMyAdmin-like tool with more powerful and manageable functionalities.

Nowadays, there are still many engineers in various organizations to create different "management console" for their applications. It is therefore necessary to continuously improve gMIS and advocate it to public for wide usages.

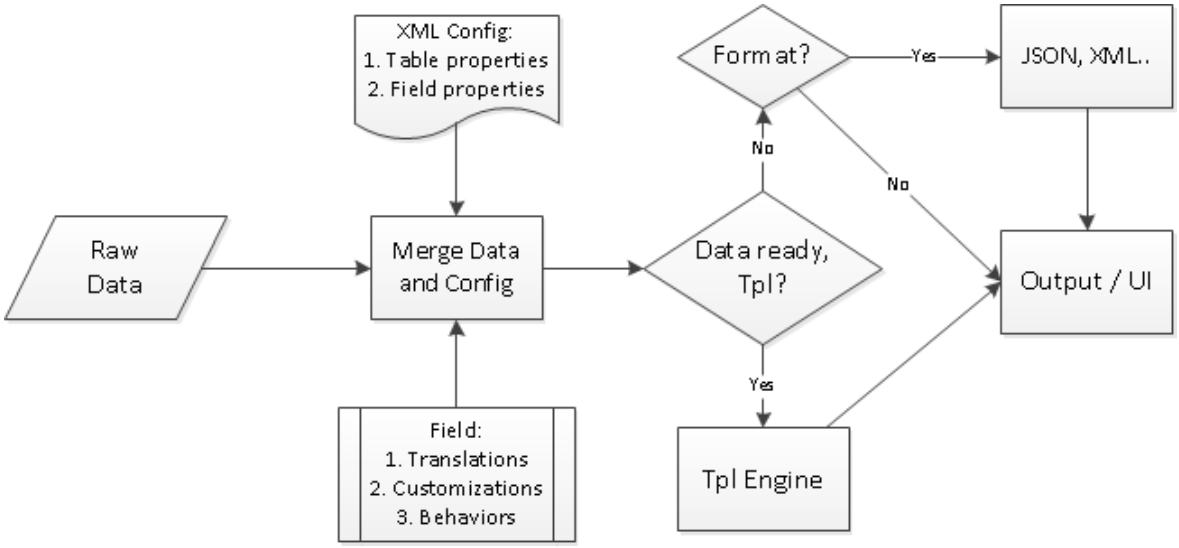

Figure 2, gMIS' flowchart

As an instance of and expanding from inc/webapp.class.php, the mission of gTbl is to read table configurations and apply those properties and behaviours to the table. Figure 2 shows the processing of gMIS. Its first step is to retrieve raw data from a database connection. Then gMIS merges data with predefined configurations. These configurations include translations, customizations, behaviours and constraints, all of which differentiate one table from the others. At its 3rd step, gMIS directs itself by whether there is a template file to be specified. If yes, it transfers the data to template engine for further output procedures; otherwise, it comes to test whether a kind of output format is needed, if one kind of data format is specified, e.g. JSON or XML, then do as request and output the formatted data.

In general, gTbl can be seen as a bridge between tables' configurations and manipulations. gTbl contains these components at present:

- **tags**

  Tags are defined to provide consistency for each property which may be used across all the application. Tags are used in describing both tables and fields. E.g. "table", "field", "chnname", "inputtype", "selectoption" and so on. All tags will be explained again in next sections.
- **set/get**

gTbl is a subclass of inc/webapp.class.php and instantiates some methods of its parent class. Most of methods are to get something from the configurations hash map which is created and filled in runtime, e.g.

a) getTblCHN, read a table's Chinese name from XML configurations via a hash map.
b) getTblCHK, read a table's validating rules.
c) getTblCharset
d) getOrderBy, read override settings for default order clauses in list view.
e) getMode, read table's access mode, usually a table can be accessed in full privileges, but in some cases, a table may be locked in read mode only.
f) getJsAction, read some JavaScript settings for a table in general.
g) getRelatedRef, read related functions and/or modules to current table.
h) getCHN, read its Chinese name of a field in current table.
i) getInputType, read a field's input type in html form.
j) getListView, read settings of whether to display current field in list view, value codes are set to be:
    i. '0': not show,
    ii. '1' or '': show,
    iii. '2': force to show.
k) getSingleRow, whether a field should be displayed in a single row.
l) getAccept, validating rules on current field. i.e. what kind of values can be input and saved in the field. This functions works with GTAjax or extra JavaScript-related html form validations.
m) setTrigger/getTrigger, set/get some trigger(s) on a field or table.
n) Other methods….

These methods can be read by source in class/gtb.class.php.

- **xml2hash**

  This function reads table configurations from an xml file and generates a key for each item of settings and save it in a hash container.

  When one table is specified in runtime, its configurations can be read from the hash container through an instance of class/gtbl.class.php.

  The hash container also includes some settings from global configurations which would be read from inc/config.class.php. The latter holds more items for the whole application.

- **I/O**

  Fields input and output are defined in xml configurations, e.g.

```
<field name="brandid">
      <chnname>商品品牌</chnname>

<selectoption>fromtable::brandtbl::chnname</selectoption>
<!—options read from another table -->
</field>
<field name="img">
  <chnname>配图1</chnname>
  <inputtype>file</inputtype> <!—upload an file(image) -->
</field>
<field name="beauty_words">
  <chnname>美丽心语</chnname>
  <inputtype>textarea</inputtype> <!—input many paragraphs a
time, e.g. an article -->
  <singlerow>1</singlerow>
  <listview>0</listview>
</field>
```

Accordingly, output of these fields will be showed in different style, e.g. a selector keep the same way of input and output; an image will show itself directly in browsers; a text area also lists all of its contents after same security validations.

More customized output tags will be needed with specified demands increasing, e.g. WYSIWYG html editor, multiple checkbox, embedded audio/video and so on.

- **ido.php/jdo.php**

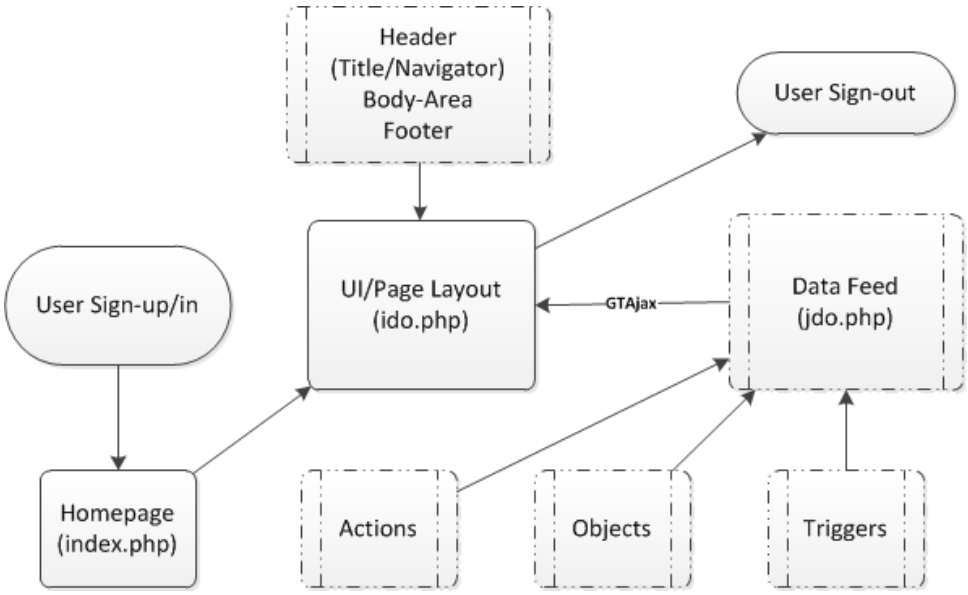

Figure 3, UI/Page Layout

gMIS has two main scripts: ido.php is designed for page layout and jdo.php is employed to manipulate data in background. These two parts are bridged with GTAjax, which triggered by user's mouse click or keyboard events.

As Figure 3 shows, the procedure can be seen that a user will be provided with a sign-in or sign-up page if he/she has not logged in. After the sign-in got done, a page or an application environment is given. All of tasks will be done within this page of the application. Behind it Data-Feed (jdo.php) continuously read or write data from/to backend services.

It is clear that gMIS just has only a PAGE in which most of other frames or popup windows/dialogs would be called for different requests and interactions.

3. Extra IO

In practice, the main purpose of a MIS is to input some and get others from the system. gMIS supports some basic input and output as text, select, file, textarea. For other uncommon field type, they might need more technical codes to be embedded. The following are some ideas on various extra input and output.

Extra IO is located in /extra/.

3.1. Inline editing

Normal listing

14 / 45284	0	呼和浩特房地产资讯	news.nm.fang.com	资讯站	自动抓取	2014-10-20 09:02:11
15 / 45283	0	中国警察网消防频道	xf119.cpd.com.cn	资讯站	手工	2014-10-20 08:58:13
16 / 45282	0	长江商学院	www.ckgsb.edu.cn	资讯站	自动抓取	2014-10-20 08:41:10
17 / 45281	厦门网- (7130)	厦门人居网	rjz.xmnn.cn	资讯站	手工	2014-10-20 08:41:09
18 / 45280	0	松原新闻网	www.0438news.com	资讯站	手工	2014-10-20 08:38:53
19 / 45279	机构或官网- (24369)	博罗县文体旅游局	wtlyj.boluo.gov.cn	资讯站	手工	2014-10-20 08:31:17
20 / 45278	0	江西消防在线	xf.jxnews.com.cn	资讯站	自动抓取	2014-10-20 08:25:51
21 / 45277	南阳网- (6768)	宛城网	wancheng.01ny.cn	资讯站	手工	2014-10-20 08:23:00
22 / 45276	0	西城区人民政府网站		资讯站	自动抓取	2014-10-20 08:17:05
23 / 45275	0	厦门业主网	yezhu.xmnn.cn	资讯站	手工	2014-10-20 08:16:52

Figure 4 Inline Editing

Double click to make it editable.

Figure 5 Inline Editing (2)

3.2. Input2Select

Figure 6 Input2Select

-R/l2Sn, -R/r2Sl

3.3. Wysiwyg

By iframe and other JavaScript-based tips to embedded a live html editor in the system.

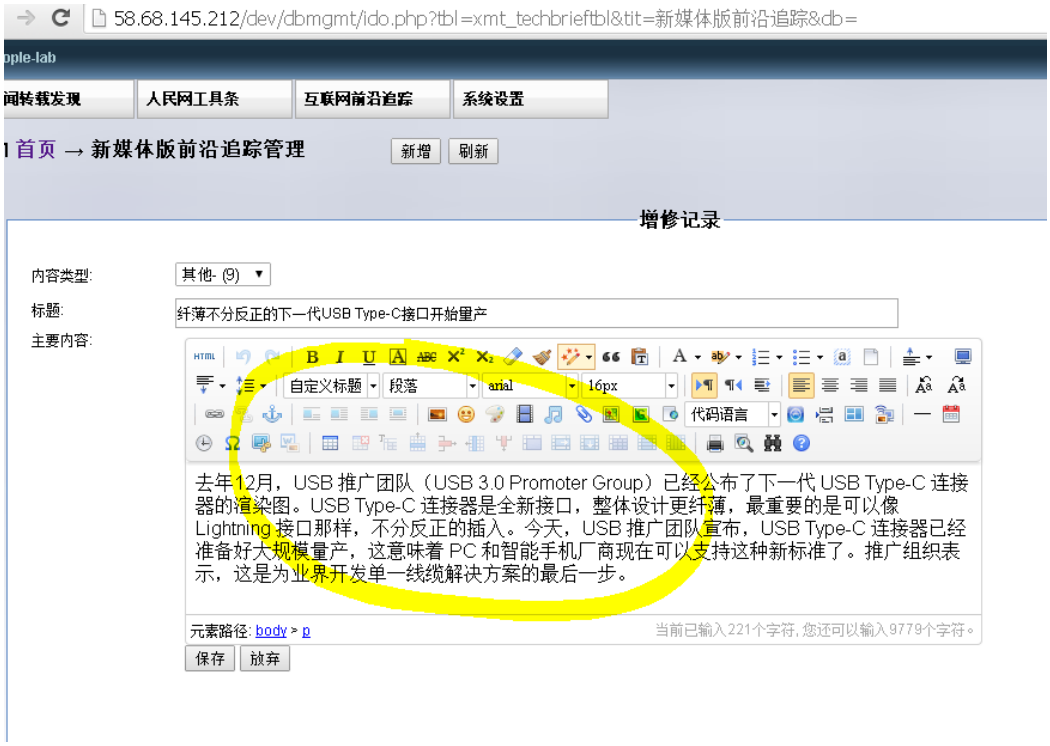

Figure 7 Live HTML Editor

3.4. Dialogs

Multiple Options List

Hierarchy Directory

3.5. TO-DO

4. Tags

Tags are used for two purposes: 1) to define a function across the whole application; 2) to be used as keys in xml2hash.

Tags are also xml node names and carry information from xml files to hash containers via xml2hash in class/gtbl.class.php, e.g.

```
<field name="sex">
    <chnname>适用人群</chnname> <!—"sex" will be showed as "适宜人群" in front page.  -->
```

<selectoption>0:女性|1:男性|2:Both</selectoption> <!—the input type would be set as "select" with three options. -->

</field>

"field", "chnname", "selectoption" are all tags, which will be referred in gtbl class as:

```
public function getCHN($field){
    $tmpstr = $this->hmconf[$this->taglist['field'].$this->sep.$field.$this->sep.$this->taglist['chnname']];
    return $tmpstr = $tmpstr==null?$field:$tmpstr;
  }
```

There are already more than 30 tags defined by now in class/gtbl.class.php. Here is the full list.

1) 'table'=>'table', # to mark a node of a table
2) 'field'=>'field', # to mark a node of a field
3) 'chnname'=>'chnname', # Chinese name of a field
4) 'inputtype'=>'inputtype', # input type of a field, available for text | textarea | select | file
5) 'selectoption'=>'selectoption', # select options, two types:
   - <selectoption>0:女性|1:男性|2:Both</selectoption> # definite options
   - <selectoption>fromtable::brandtbl::chnname</selectoption> # dynamically read from another table
6) 'selectmultiple' => 'selectmultiple', # is multiple of a select or not
7) 'extrainput'=>'extrainput', # other input types which cannot be showed as common, usually
   - read/write to another table via extra/linktbl.php or extra/OTHER
   - other special I/O control, e.g. multiple checkbox
8) 'memo'=>'memo', # memo message for a field or table
9) 'charset'=>'charset', # character charset
10) 'dbname'=>'dbname', # database name
11) 'relatedref'=>'relatedref', # related functions of a table
12) 'listfieldcount'=>'listfieldcount', # how many fields to display in list view within a single row
13) 'listview'=>'listview', # hide in list view or not, '0': not disp, '1' or '': disp, '2': force to disp
14) 'singlerow'=>'singlerow', # display in a single row
15) 'printref'=>'printref', # for a table in print layout
16) 'reftable'=>'reftable', # related tables to current table
17) 'jsaction'=>'jsaction', # JavaScript binding to a table or field
18) 'delayjsaction'=>'delayjsaction', # JavaScript for a table or field, delay its start time in runtime
19) 'check'=>'check', # process business check logic...
20) 'orderby'=>'orderby', # explicitly specify a field for ordering...
21) 'defaultvalue'=>'defaultvalue', # default value during add/modify…
22) 'managemode'=>'managemode', # managemode for a table, r(read),w(write),d(delete)

23) 'accept'=>'accept', # front-end validator, e.g. "lt=100,gt=1000"
This requires GTAjax loaded firstly.
24) 'trigger'=>'trigger', # trigger sth when meeting some requesters
The tag makes it come true that a static project goes alive and health with the concept of work flow.
25) 'readonly' => 'readonly', # there are two kinds of read-only data:
- some fields do not need input by users, but by programs;
- for privileges reason, some fields are denied to modify by some users.

26) 'href' => 'href', # href of a field
27) 'hidesk' => 'hidesk', # default search key, e.g. environment variables like time, IP, user ID and so on, and with these clauses the query can be limited in a small scope.
28) 'css' => 'css', # css of a field or table
29) 'superaccess' => 'superaccess', # access control over system settings
30) 'stat' => 'stat', # methods used when making statistics, sum | count | average

Some of tags share a few default settings, e.g.

0 or '': always means "No", "Not", "Negative", "Empty";

1: stands for "True", "Positive", and "Not Empty".

A brief of a configuration xml looks like:

```
<?xml version="1.0" standalone="yes"?>
<tablecfg>
    <table name="producttbl">
        <chnname>商品详情表</chnname>
        <listfieldcount>8</listfieldcount> <!--max_disp_cols-->
        <since>20121015</since>
        <creator>Wadelau</creator>
        <superaccess>inherit::id=USER_ID::rw</superaccess>
    </table>
    <field name="productno">
            <chnname>商品编码</chnname>
      </field>
    <field name="chnname">
      <chnname>中文名</chnname>
      </field>
...
...
</tablecfg>
```

5. Actions

Actions include standard tasks of what have been discussed above, i.e. CURDLS, and non-standard jobs which generated by specific work of a project. Each of standard actions needs at least an html form and a corresponding do-form action. For non-standard jobs, there are some tailored pairs.

There are therefore the following mandatory actions located in act/:

1) addmodi.php, to present creating form and/or updating form
2) doaddmodi.php, to receive and save data filled by user in addmodi.php
3) dodelete.php, to function a backend service for dropping records
4) jdo.php(in /), to function as a combination of LIST and SEARCH
   Page navigator[8] in section 9 of GWA2describes the powerful SEARCH in great detail.
5) log.php(writelog.php), to log each step in backend, provide traceable history
6) print.php, to display an html form in printable style
7) view.php, to present a retrieving form with stored data
8) updatefield.php, to edit any single field of a row separately, i.e. inline editing of user data

Other non-standard jobs include (in act/):

9) checkaccess.php(and tblcheck.php), for security-check jobs
10) sendbulkmail.php, for mails sending
11) toexcel.php, for downloading data in Microsoft Office Excel compatible format
12) trigger.php, for dynamically invoking other actions, which make work flow possible

These actions listed below are used to carry out dynamic modules:

13) synctblfield.php
14) synctblindexkey.php
15) updateobjectfieldtbl.php
16) updateobjecttbl.php

Actions can be launched by either users' clicks or being triggered by other actions.

All actions are handling in jdo.php(see Figure 3), and they are provided in the parameters of each request as:

http://ufqi.com/dev/proj-a/console/?tbl=tbl-a&act=view&...

For various actions, jdo.php includes specified act scripts to make transactions. At present, there are a few of pre-defined acts in paraments:

&act=add | modify | list | list-addform | list-dodelete | view | print | updatefield

[8] http://ufqi.com/dev/gwa2/General.Web.Application.Architecture.201301.v4.pdf

“list” is the default action. “updatefield” is worth being explained in more detail. As a distinctive point of Chive, “Inline editing of data” provide users a quiet convenient way of working with data input and output, so does gMIS. “updatefield” is designed to edit any single field of a row separately, e.g.

Figure 8, Inline editing of data in gMIS.

6. Triggers

Actions are routine jobs and triggers are dynamically-set work which keep the target project “alive and health”, e.g. procedures automation, data consistence, message notifications, auditing and monitoring and traceable history (log).

In the general view of MIS, triggers are set by the concept of these principles showed in the following figure (Figure 9).

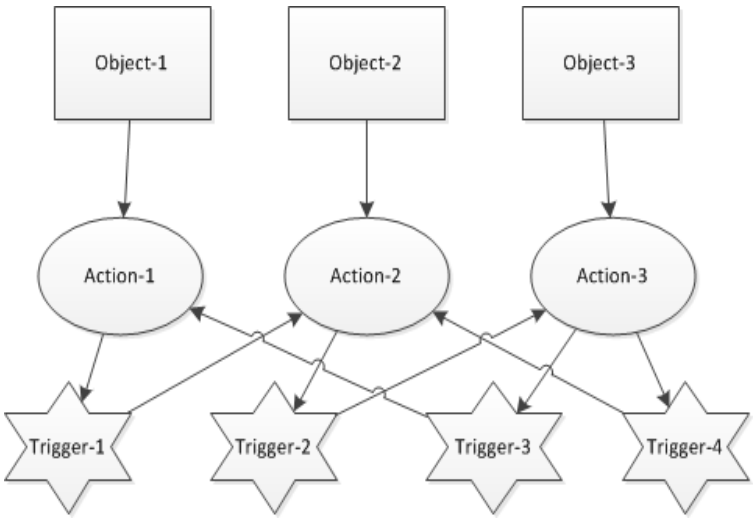

Figure 9, Triggers in gMIS

Suppose object-1 has an action-1. When a user launches the action-1, the action would do another action-2 by trigger-1. That is to say, a user’s request results in two actions at a single time. In the same way, object-2 gets two actions from action-2 with an accompanying action-3 via trigger-2. Object-3 gets three actions done in a single request with two triggers: trigger-3 for action-1 and trigger-4 for action-2.

An example demonstrates this clearer.

In an office daily work, a clerk raises an application for leave. When the clerk clicks the submit button on an application form after filling some information, gMIS will do at least two jobs: 1) one is to save the data of this application form; 2)forward the application to the clerk's manager to approve this application. The second action is set by a trigger looking as:

```
<field name="state">
      <selectoption>0:applying|1:manager-
auditing|2:passed|3:done</selectoption>
      <trigger>0::notify::manager_todo_tbl::….
            |2::notify::clerk_leave_tbl::…
      </trigger>
      ….
</field>
```

It is easy to see that triggers simulate the mode of work flow in the scenario. In general, gMIS splits each step of a complex job into many pieces of work. For each piece of work, a trigger and a corresponding action are defined and every one of them can be invoked or triggered in a loop-like application environment. Another benefit from this design is that each action can be reused again for other objects. It is therefore to realize that write once for all, forever.

Triggers are interpreted in act/trigger.php. Triggers are bound to updating actions, e.g. act/doaddmod.php, act/dodelete.php.

7. Dynamic Modules

Adding or removing modules or objects could be done dynamically in gMIS because that gMIS inherits some gene of Chive and/or phpMyAdmin. Just like manipulating another new table in Chive, managing a new object/module is easy to achieve by creating an xml file for it.

Through an xml configuration is optional, it is recommended to generate a customized xml configuration file for each table. The xml file usually contains lots of information on what the table/object/module looks like and how the data of the module will be transacted.

Generally, adding a module is to create a link to a URL like:

./?tbl=tbl-new&act=list&…

Then, a web-based management console is showed in a modern browser. It is just a finger-snap. But there is a long way to improve the basic CURDLS function into a customized MIS. To map the module into a position (link) in the menu of this MIS is the first step. This step can be done in default gMIS menu bar and its path is Settings → Menu Adjustment.

The second step it to create the module in the path of Settings → Modules. This function looks much like phpMyAdmin or Chive. These kinds of tools provide users lots of convenience that instead of creating tables in command-line console via a remote connection to server, such web-based admin interfaces allow user to manipulate tables directly, e.g. create a table, drop a table, add/remove/update table fields, add/remove/modify table indexes and keys. All in one, this function is a web-based console for tables' management for a database.

For an instance, in a running application administrators want to add a module to support a new kind of product which has its own properties and actions to do.

Table 1, ways of adding new module

		Traditional MIS	**gMIS Dynamic Module**
Developing	Basic CURDLS	Dir-a/, add-a.php/add-a.html, do-add-a.php, edit-a.html/edit-a.php doedit-a.php list-a.php/list-a.html search-a.php/search-a.html do-search.php dodelete.php	(No script files needed, add new module in web-based console).
	Business-based	OTHER-BUSINESS_LOGIC.php ….	OTHER_BUSINESS_LOGIC.php ….
Running		modify menu, add new link manually in script files; Train operators for new functions.	Add new link in web-based console. Notify operators.
Time cost		3 plus days.	3 minus hours.

By saving time and work cost, dynamic modules make gMIS self-improve and self-evolve to meet new changes with business growing. It decrease the amount of coding work to a very lower level by reuse basic CURDLS functions and also decreases the work of training operators for new modules due to that modules added in this way share the same UI and logic.

8. Full-function Searching

Searching and Paging are two basic listing jobs in any MIS. These functions come from -GWA2 which is the mainframe of the management system.

Figure 10 Powerful Searching

For a given field, there are some detailed operators to be selected to mix the whole searching. E.g., omit the field, equals, not equals, contains, not contains, greater than, less than, in a range or in a list.

In addition, there are two options two concatenate or join fields with different fields and their operators: meet all of these clauses (and), meet any of those clauses (or).

For a detailed manual reference, please go to -GWA2.

9. Security, Firewall-style Auditing

It is no more careful than us to strengthen the securities with web-based MIS since our servers ever caused severe data leak caused by one issue with phpMyAdmin in around 2000 when we provided maintenances for those servers.

Based on a patent solution[9] and the theory of access control[10], we design a new mode of access control mechanism. The design works like firewall for network packets flowing.

System securities can be set and adjusted in the path of Settings → Privileges Management.

The mode has three parts: objects and objects group, users and users group, access methods and their combinations. Matching any three of these parts can make a rule of access control, e.g.

User-a → object-a → read & write

User-group-b → object-b → read only

User-c → object-group-c → read only

[9] http://www.cnpatent.com/list_zhuanli.asp?id=200810171394&zt=
[10] http://infosec.pku.edu.cn/~wyz/course/8.pdf

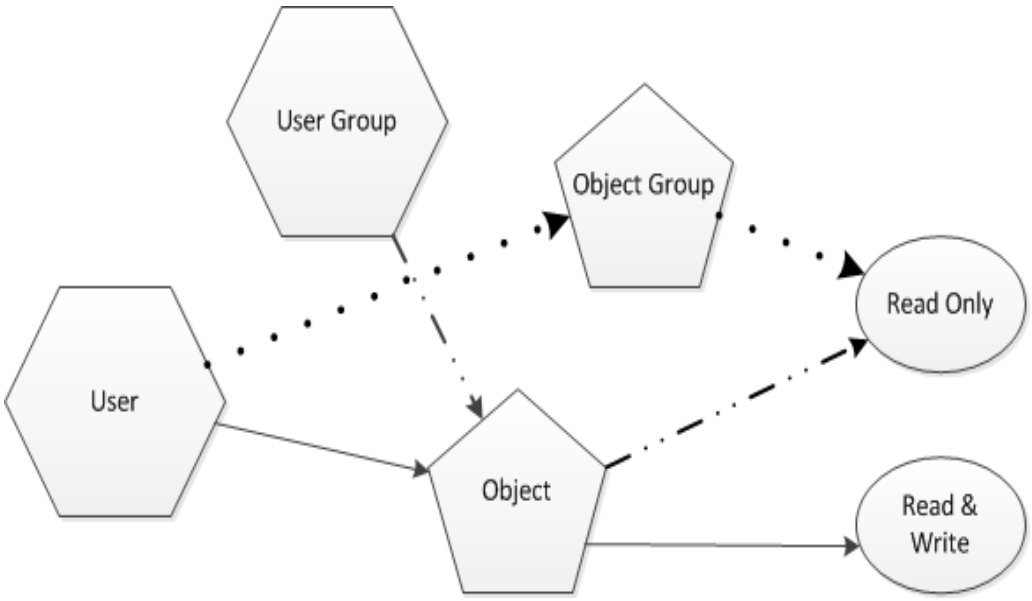

Figure 11, Securities Mapping in gMIS

Unlimited rules of access control could be made as wanted except that there are sometimes conflicts between two of these rules. For instance that, user-a could read object-b according to rule-a but could not read it by rule-b. Therefore one more field of priority is provided to make sure that when conflicts aroused, the rule with higher priority will win.

In gMIS, the access control goes further for a table-based record. It provides a tool that allows administrators to hide one column of a record but show other columns of that table, e.g. in table with four fields:

Field-a, field-b, field-c, field-d

A rule says user-a could read field-a, field-b and field-c but not field-d.

In runtime, for every request, there is a full-match checking by users' inputs, e.g. objects, actions and so on. If no access control rules are matched, it goes to the default. That is to say, one can set a system with open-door or close-door in default.

The procedure of checking access is at act/checkaccess.php which is included in comm/header.inc.

Figure 12, rules of access control in gMIS

10.To-do

10.1. Automatic Modules and Tags Setup

Instead of composing XML configurations manually, an automatic toolset will be made to provide this procedure of manipulating modules and tags in a web-based manners.

10.2. API

For a future deployment, API is selected for both mobile Internet and desktop applications.

10.3. i18n

Internalization for multiple languages support in UI.

11. Document history

Table 2 document history

No.	Version	Updates	Date	Author(s)	Inspector(s)
3	V0.3	Added Section B（End-user Manual）and Section C （Updates and Maintenances）, the document got much longer. -gMIS would be copyrighted.	2016-02-28	Zhenxing Liu, Shujuan Wang	Zhenxing Liu
2	V0.1	(Revised)	2013-03-09	Zhenxing Liu	
1	v0.1	Initial draft	2013-01-23	Zhenxing Liu	

B. 系统开发说明/Developers Manual Sections

12. 导航菜单动态调整

（1）选择主导航菜单——系统设置——菜单调整

Figure 13 Menu setting (1)

（2）进入下面的“菜单调整”页面

Figure 14 Menu setting (2)

- 其中“链接名”是指页面上显示的菜单名例如：【商品管理】【度假村管理】【公共信息维护】等等。
- 层代码代表的是菜单的层级：
  - 第一级菜单是两位数字的 例如 33,34,35，根据数字的大小菜单自动排序，不一定非从 00 开始；
  - 第二级菜单依附于第一级菜单，例如 33 下面的二级菜单即为：3301,3302，依次类推；
  - 同理三级菜单式依附于二级菜单的，例如：330101,330102。
- 模块名字：通常可以为数据表名，也可以不一致，来源于 xml 目录下相应的文件名。
- 显示标题：为 title。

（3）增加完成后，退出重新登录即可，或者刷新主页面即可。

这样显示的原理：

在 admin/navimenu/navimenu.php 文件里面，对如何自动获取菜单进行了定义，就是从 info_menulist 模块中获取各个字段，然后填充到菜单中。

13. 后台挂表操作

（1）将相应的数据表，通过 xml 配置文件进行挂表，即在 admin/xml 目录下，把所有的数据表逐一生成对应的配置文件描述表。该 xml 文件描述数据表层面的描述，各个字段的个性配置信息。如下图所示。

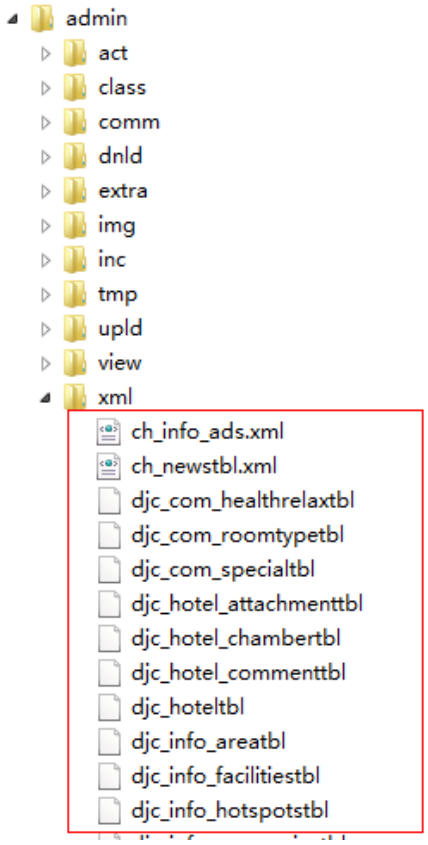

Figure 15 Table setting

模块名为不带前缀的数据表名，例如：djc_hoteltbl --> hoteltbl.xml。

（2）在每一个 xml 配置文件中，把数据表的相关字段挂进去，从而在页面进行展示，例如下面即是一个数据表的配置文件描述信息。

```
<?xml version="1.0" standalone="yes"?>
<tablecfg>
    <table name="fdk_ch_info_ads">
        <chnname>广告信息</chnname>
        <listfieldcount>12</listfieldcount> <!--max_disp_cols-->
        <since>20130718</since>
        <creator>Sunlit</creator>

    </table>

        <field name="adplace" type="text">
            <chnname>ad位置</chnname>
            <inputtype>select</inputtype>
            <selectoption>homepage:homepage|aboutus:aboutus|businessfield:businessfield|fdkproduct:fdkproduct|cont
        </field>
    <field name="starttime">
            <chnname>开始时间</chnname>

        </field>
        <field name="endtime" type="text">
            <chnname>结束时间</chnname>
        </field>
        <field name="adname">
            <chnname>名称</chnname>
        </field>
        <field name="content" type="text">
            <chnname>内容</chnname>
            <listview>0</listview>
        </field>
        <field name="info">
            <chnname>信息</chnname>
        </field>
```

Figure 16 Table setting (2)

14. 字段 JavaScript 和 CSS 修饰

在给定的字段上，可以通过 xml 设置对其进行绑定。目前提供了两种 JavaScript 程序的支持，一种是同步的，在<field>下面通过<jsaction>进行相关设置，如：

<jsaction>onchange::fillSubSelect('objectid','objectfield','xiane','./extra/readtblfield.php')</jsaction>

//- 在某个字段上绑定当状态值发生改变时，触发 fillSubSelect 动作

在某些字段的设置上，有时候需要延迟执行某一类动作，这种情况需要考虑使用异步的 Javascript 来绑定相关动作，如：

<delayjsaction>onload::3::oldv=document.getElementById('tblname').value;if(oldv!=''){document.getElementById('tblname').readOnly=true;}</delayjsaction>

//- 在加载的时候，延迟 3 秒执行后面的代码段

在给定字段或者表上的 CSS 修饰通过

<css>1::A|2::B|3::C</css>

来实现，其中的值在前端转换为 class=”A”, 上述设计表示，当字段值为 1 时显示为 class=”A”, 字段值为 2 时显示为 class=”B”, 3 为 class=“C”。

15. 公共设施

公共设施由 comm/tools.function.php 内的各类函数提供.

在这里定义的函数是全局适用的，除非另有 namespace 做命名空间或作用域定义的地方除外。

15.1. HTTP POST

封装了 PHP 内置的 curl 组件常用的命令，形成可调用的 curlPost 函数。

/**

* Send a POST requst using cURL, refer to http://www.php.net/manual/en/function.curl-exec.php

* @param string $url to request

* @param array $post values to send

* @param array $options for cURL

* @return string

*/

15.2. 邮件发送 Send Mail

邮件发送提供了两种方式，调用方式都是 sendMail. 发送邮件可以是本地的邮件希望的 sendmail, 还有一种方式是调用内置的基于 PHP socket 通信实现的邮件发送。

15.3. 字符串组件 String Utilities

inString($needle, $haystack)

inList

startsWith

endsWith

substr_unicode

shortenStr

15.4. 简单路由 Simplified-Routing

alert

生成一个浏览器的 window.alert 代码，终止当前页面输出，发送给客户端。

redirect

生成一个 HTTP 的 302 返回状态，终止当前页面输出，发送给客户端。

15.5. 调试 Debug

参考 升级维护日志“**一种 debug 方法的实现**”一节。

15.6. 图片组件 resize Image

实现对给定图片的扩展或压缩，可以指定目标宽度或者按一定百分比放大或缩小。

```
function resizeImage($srcFile, $toWidth, $percentNum=1, $destQuality=85)
# resize image by GD functions
# wadelau@ufqi.com, Thu Jan 28 22:04:14 CST 2016
# return resized image
# $toWidth: int, 10~10000 ?
# $percentNum: float, 0~1
# e.g. $destFile = resizeImage($srcFile, 1024);
# e.g. $destFile = resizeImage($srcFile, 0, 0.55);
```

15.7. 日期组件 Date Picker

对于绑定了 DatePicker 的字段，在点击是弹出日期选择器，方便用户挑选目标日期。

Figure 17 Date Picker

16. 异步删除演示删除/Delete delay

参考 升级维护日志“删除确认、删除异步及删除延时，delete confirm，aysnc and delay”一节。

17.To-do

C. 升级及维护记录/Updates and Maintenances

18. 两例 inline editing 的使用案例——未来可视化编辑的趋势

Posted on 2014 年 2 月 11 日 by Wadelau

Inline editing (in-place editing)，内联编辑 或者 行内编辑 或者 现场编辑

是籍着 div 的 editable 属性，contentEditable="true"; 可以让页面的一个区块文字变得可编写， inline editing 借着现代浏览器对 JavaScript 的强力支持，开始逐渐被用到可视化编辑器里，如下是两个例子。供参考。

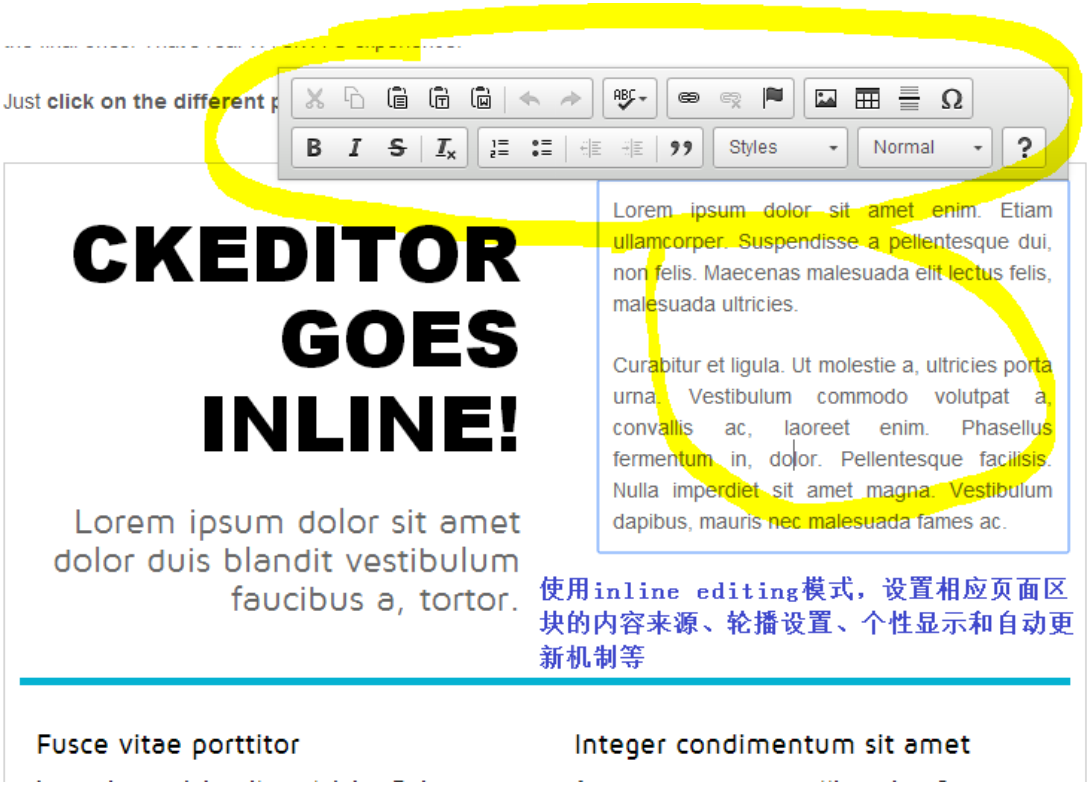

Figure 18 Inline editing (3)

有名的 CKEditor 的 Inline Editing 模式

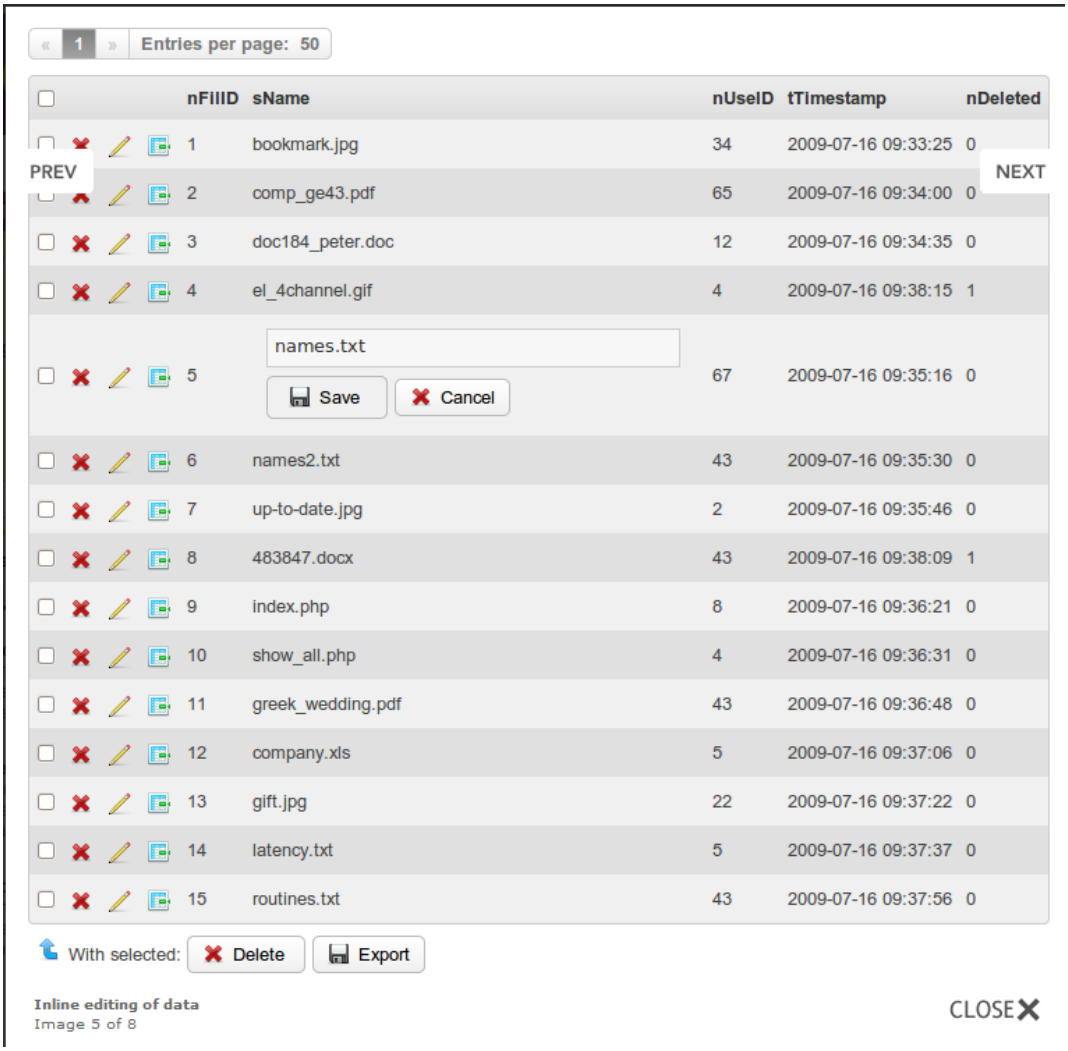

Figure 19 Inline Editting (4)

有望取代 phpMyAdmin 的 Chive 用上了 Inline Editing.

19. gMIS 更新–增强导航栏搜索，显示页面过滤脚本字符

Posted on 2014 年 3 月 20 日 by Wadelau

最近将 gMIS 部署到一个新应用中，一如既往的快捷、方便。几乎零代码实现一个新的管理后台。

改进了如下几项：

1. 增强了导航栏搜索功能，增加了忽略某个字段作为搜索条件的设置；

2. 在显示页面，list/view, 对输出的文本进行脚本字符的过滤， ‘<’ –>’<’ ;

3. 修正了 select 使用 fromtable 作为选项来源时的，条件语句；

20. gMIS 主要更新 201404

Posted on 2014 年 4 月 17 日 by Wadelau

gMIS 通用管理信息系统，最近进行了几个主要更新：删除记录后丢失 pntc 的导航页异常；初始化查询时的重复 SQL 查询问题；下拉框从表中选择 options 时的缓存优化等。

1. 删除记录后的 list 页面丢失 pntc，进而导航页异常

使用 gMIS 部署到一个新项目，频繁需要删除数据，主要异常行为是，在 list 页面时，如果删除掉一条记录，则当前页面的导航翻页等功能显示正常，但在进一步点击翻页时，翻页信息全部丢失。

经查，涉及到 class/pagenavi.class.php 中的

```
if(strpos($query, "act=list-") !== false){
      $query = preg_replace("/act=list\-([0-9a-
z]*)/","act=list",$_SERVER['QUERY_STRING']);
      $this->hmf['neednewpntc'] = 1 ; # newly-added
      $query = preg_replace("/&pntc=([0-9]*)/","", $query);
}
```

然后在 jdo.php 里，需要进一步的做处理：

```
if($_REQUEST['pntc'] == '' || $_REQUEST['pntc'] == '0'
            || $navi->get('neednewpntc') == 1){
      $pagenum = $gtbl->get('pagenum');
      $gtbl->set('pagenum', 1);
      $hm = $gtbl->getBy("count(*) as totalcount", $navi-
>getCondition($gtbl, $user));
      if($hm[0]){
            $hm = $hm[1][0];
            $navi->set('totalcount',$hm['totalcount']);
      }
      $gtbl->set('pagenum', $pagenum);
}
```

其中， $navi->get('neednewpntc') 的判断条件是新增的，原因是在前面的 class/pagenavi.class.php 中去掉了相应的参数，这里要重新算出。 由于$navi 对象此时持有的环境变量 $pagenum 和 $pagesize 是导航页的主对象 $obj 的，而 select count(*) as totalcount 的结果集，只会有一条记录，因此改之前会出现下面这样的查询 SQL:

mysql> select count(*) as totalcount from tbl where ..... limit 100,20

这样基本有结果集，但 limit 的太靠后导致读取不到结果，进而丢失了 pntc，导航页面也就失常了。

已修正。

2. 初始化查询时重复 SQL 问题

gMIS 里，在 inc/webapp.class.php 中，如果活跃对象 $obj 进行属性读取时，$obj->get('xxxx');时，系统会先读取运行时环境值，如果没有就触发去持久层读取相应数据，首选的是读取数据库。

```
# revised.

function get($field){
      if(array_key_exists($field,$this->hmf)){
            return $this->hmf[$field];
      }
      else if($field != 'id' && $field != 'er'){ #! Otherwise, this
will cause a dead loop with ._setAll.
            if($this->get('er') != 1){
                  if($this->_setAll()){
                        if(isset($this->hmf[$field])){
                              return $this->hmf[$field];
                        }
                        else{
                              return $this->hmf[$field]='';
                        }
                  }
                  else{
                        return '';
                  }
            }
            else{
                  return '';
            }
      }
      else{
            return '';
      }
}
```

其中 $this->get('er') 的判断是新增的，如果第一次读取已经知道是空记录，或者无记录，则在当前会话中，不再继续多次查询同一 SQL.

类似的，还新增了 isset($this->hmf[$field]) 的判断，这是针对某个运行时变量只在 app 中，而没有在持久层中，这样的也会导致因为一个变量的不确定值，而产生的多次同一 SQL 的查询，加入此判断，则可以减省掉这些不必要的开支。

3. 下拉框选择项的动态读取，重复 SQL 查询的问题

gMIS 中提供了动态下拉框的功能，也即一个下拉框的数据选择项，可以动态的从另外一个数据源（表）中读取。这为管理信息系统提供了丰富的功能。

之前的设计中，在 gMIS 的 class/gtbl.class.php 中，每次都来一个 getSelectOption 就读取一次数据库，改进后的做法，增加了缓存：

```
$hmoption = array();
if(isset($this->hmconf['selectoption_'.$field])){
	$hmoption = $this->hmconf['selectoption_'.$field];
}
else{
	$oldhmf = $this->hmf;
	$this->hmf = array();
	$this->setTbl($tbl);
	$hm = $this->getBy("id,$dispfield", $arr[3]);
	if($hm[0]){
	$hmoption = $hm[1]; $this->hmconf['selectoption_'.$field] =
$hmoption;
	}
	$this->hmf = $oldhmf;
}
foreach($hmoption as $k=>$rec){
	$optionlist .= "<option value=\"".$rec['id']."\"";
	if($defaultval != null){
		if($rec['id'] == $defaultval ||
strpos(",".$defaultval.",", ",".$rec['id'].",") !== false){
			$optionlist .= " selected";
			$selectval = $rec[$arr[2]].(isset($arr[3])?"-
".$rec[$arr[3]]:"").(".$rec['id'].")";
			if($needv == 1){
				$selectval_mul .=
$rec[$arr[2]].(isset($arr[3])?"-".$rec[$arr[3]]:"").("
(".$rec['id'].),";
			}
		}
	}
	$optionlist .=">".$rec[$arr[2]].(isset($arr[3])?"-
".$rec[$arr[3]]:"").(".$rec['id'].")</option>\n";
}
```

$hmoption 缓存到$this->hmconf 中是新增的功能，这样在同一个 list 页面中，每个 select 字段，只要一次 SQL 查询数据源即可。

—- —-

写这么多，发现之前的要优化的地方不少，改进无止境。

21.-gMIS 更新–增加“不等于”操作符

Posted on 2014 年 6 月 30 日 by Wadelau

-gMIS 更新，在对字段的操作符中，增加“不等于（!=)”项的操作符。

代码：

1. class/gtbl.class.php

```
public function getLogicOp($field){
$intop = array('—-'=>'忽略', '='=>'等于',
'!='=>'不等于',
'>'=>'大于',
'>='=>'大于等于',
'<'=>'小于',
'<='=>'小于等于',
'inlist'=>'等于列表中的一个,如: 1,2,3′,
'inrange'=>'在一个值域中,如: min,max');
$strop = array('—-'=>'忽略','='=>'等于',

'!='=>'不等于',
'contains'=>'包含',
'notcontains'=>'不包含',
'inlist'=>'等于列表中的一个,如: 1,2,3′,
'startswith'=>'以...开头',
```

2. class/pagenavi.class.php

```
if($fieldopv == 'endswith'){
	$condition .= " ".$pnsm." "."$field like ?";
	$gtbl->set($field, "%".$v);
}
else if($fieldopv == '!='){
	$condition .= " ".$pnsm." "."$field <> ?";
	$gtbl->set($field, $v);
}
else{
	$condition .= " ".$pnsm." $field $fieldopv ?"; # this should
be numeric only.
	$gtbl->set($field, $v);
}
```

22. 一例网络应用开发的 bug 分析

Posted on 2014 年 7 月 3 日 by Wadelau

作为 -GWA2 的一个实例， -gMIS 一直运行良好，今天在部署到一个新项目测试运行中，发现一个令人不安的 bug，虽很快找到并修复，但就像 OpenSSL 的 heartbleed 一样，需要事后反思。

事情缘起这样，新项目部署后，其中一个数据表的记录被不断的更新，数据被清空。由于是部署在一个“伪私有主机”上，无法登陆到服务器进行查询，所以加大了判断和 troubleshooting 的难度，好在对于 -gMIS 相当熟稔，同时对业务逻辑也很明细，再同时 -gMIS 里能够看到详细的 operatelog , 这样定位起来，即便在“虚拟主机”，看不到访问日志的情况下，基本推测出出错的流程。

错误点在于，由于未知的原因，远程访问的请求拼出类似下面这样的地址：

admin/jdo.php?act=list-addform&id=...

常规情况下，这一请求会被

admin/comm/header.inc

里的判断逻辑所阻隔，并跳转到登录页面，所使用的语句是

header(“Location: LOGIN_PAGE”);

问题恰出在此处，虽然这里设置了 header , 也即在 HTTP 的 response 头部增加了 302 的跳转导向，但！是！ header 之后的语句逻辑会照样被执行，也即符合相关条件的记录仍会被更新，由于这个请求是被恶意拼接的，所以相应的数据根本就是错误的，也因此出现了开头的一幕，数据被不断的重置为空，但记录数并没有变，只是项的值被不断的填入为“”。

经上分析测试，在测服务器上快速部署了代码

exit(0);

在发现未登录用户的情况下，随即结束当前页面的业务逻辑，终止进一步操作并将页面处理权由 PHP 交回到 Apache 那里去。

虽然是个小 bug，并且发现及时，但仍事后感到心有余悸。为何之前没有预料到这样的错误呢？如果没有

exit(0);

执行语句，程序页面在测试时并无任何异常，逻辑仍是按照预期的路线跳转到登录页，只是这种依赖终端浏览器的测试，不能察觉出，实际的 mod 和 act 仍旧被执行了。这里只测试了浮表，并未能使用 code review 的方式进行深入分析每一块逻辑和每一行语句。

软件仍是在不断使用和测试中成长的。

23. -gMIS 更新：增加 input2Select 功能

Posted on 2014 年 7 月 31 日 by Wadelau

所谓 input2Select 是将当下正在操作的 input 输入框变成 select 一样的功能，也即随着用户的输入，在下方会出现类似下拉列表的备选项或者提示项。在搜索引擎中又叫做 instant search.

Figure 20 Input2Select (3)

如上图所示，在 -bing 里输入“自然”两个字，下面会出现备选项的提示。

在 -gMIS 原本无此需求，但当某个字段的使用 select 输入方式时，如果选择项超过 50 或者 100 之后，用户检索、选择起来就比较困难。如下图所示。

Figure 21 Input2Select

这时如果能有个类似搜索引擎的那种先根据关键词过滤一下相关条目，然后再列出来就比较好。想到做到，经过一番分析和实验，实现如下效果：

Figure 22 Input2Select

主要逻辑：

1. 将原来的 select 使用 css 修饰为 style.display='none', 设置为隐藏；

2. 增加输入框，在输入框的附近设置一个隐藏的用于显示搜索结果的隐藏 div；

3. 在输入框上绑定 onkeyup 动作，用户每有输入即进行检索：1）可以在本地检索隐藏的 select 里的 options，也可以即时联网从服务器上获取检索结果（像通用搜索一样）。

下面的代码中使用的是第一种方式。

涉及到的程序：

```
function input2Select(inputx,obj){
	var lastSearchTime = userinfo.lastInput2SearchTime;
	var nowTime = (new Date()).getTime();
	var balaTime = nowTime - lastSearchTime;
	var inputVal = inputx.value;
	var obj1737 =
document.getElementById('pnsk_'+obj+'_sele_div');
	if(inputVal.length < 2 || balaTime < 1000){
		console.log('input-length:'+inputVal.length+',
balaTime:'+balaTime);
		//obj1737.innerHTML = '';
		return 0;
	}
	else{
		var iInputX = inputx.value.toLowerCase();
		var hidesele = document.getElementById('pnsk_'+obj+'');
		var odata = '';
		var dataarr = []; var j = 0;
		if(hidesele != null){
			for(var i=0; i < hidesele.length; i++){ //- 复制搜
索栏里的select选项到当前
				var seleText = hidesele.options[i].text;
				if(seleText.toLowerCase().indexOf(iInputX) >
-1){
					//-
					dataarr[j++] = '<span
onmouseover=parent.changeBGC(this,1);
onmouseout=parent.changeBGC(this,0);
onclick=parent.makeSelect(\'input2sele_'+obj+'\',this.innerText,\'pn
sk_'+obj+'_sele_div\',\'pnsk_'+obj+'\');>'+seleText+'</span>';
					if(j>30){
						dataarr[j++] = '更多……';
						break;
					}
				}
			}
		}
		if(dataarr.length == 0){
			dataarr[0] = '……Oops! 没有找到/Not Found….';
		}
		odata = dataarr.join('<br/>');
		//console.log(odata);
		obj1737.innerHTML = odata;
		userinfo.lastInput2SearchTime = (new Date()).getTime();
	}
}
```

在自己实现上述功能之前，找了网上一些类似的功能代码，都感觉不够简洁。自己写起来相对还不算太难。

24. -gMIS 更新兼容 Strict SQL Mode

Posted on 2014 年 8 月 18 日 by Wadelau

PHP5.4x 和 MySQL5.5+ 会导致错误信息：

Incorrect integer value: ” for column ‘ID’ at row 1 ”

主要是因为给定的字段值不符合给定的要求，尤其是在建表的时候指定了 NOT NULL，而在 insert/update 操作时，又没有指定相应的值，就会报错。一种解决办法是修改 MySQL 的配置参数，`STRICT_TRANS_TABLES`，`http://dev.mysql.com/doc/refman/5.6/en/sql-mode.html#sqlmode_strict_trans_tables` 。
另外一种方式是按规范给相应的字段默认值。-gMIS 的更新使用的是后一种方式，主要修改如下：
1) comm/tblconf.php

```
$hmfield[$field] = $fieldv;
$hmfield[$field."_default"] = $v['Default']; # 新增，读取字段的默认值
$tmpsort = $hmfieldsortinxml[$field];

.....
```

2) act/doaddmodi.php

```
$fieldv = $_REQUEST[$field];
if($fieldv == ”){
    $fieldv = $hmfield[$field."_default"]; # 新增，如果数值为空则读取默认值
}
else {

.....
```

25. -gMIS 更新：增加 input2Select 功能(2)

Posted on 2014 年 9 月 11 日 by Wadelau

上回根据需要实现了，

-gMIS 更新：增加 input2Select 功能

-R/l2Sn

该功能只是在 -gMIS 的搜索/导航栏内可以使用。如果要想在记录列表内使用，尤其是与 -gMIS 的另外一项功能“inline editing” 相结合的话，可能还要花一些功夫来折腾一下。

经过一天的努力，顺利实现了 input2Select 与 -gMIS 原有的 行内编辑（inline editing）完美的结合起来，也即针对输入方式是 select 的内容，也可以实现在行内编辑，在此前将可视区域的 div 转为 select 不同， 增强版的 input2Select 可以将可编辑的 div 随着输入的进行，即时呈现备选结果，如下图：

Figure 23 inline editing

双击启动行内编辑器，

Figure 24 Input2Select

解决的问题如前一篇，在面对较大量级的 select 选项时，人工翻阅是难以接受的。

增加了智能筛选就好方便了。

26. -gMIS 更新：增加默认主页和多层全路径

Posted on 2014 年 10 月 24 日 by Wadelau

-gMIS 通用信息管理系统 主要更新：

1. 增加默认主页，在该页面增加一些快捷链接实用信息。

Figure 25 Improved Homepage

主要包括常用的模块列表，根据调用频度排序，调用频次越高的排列往前。另外提供桌面列表，手工调整，增加一些快捷方式在桌面上。

2. 在导航栏提供全路径

Figure 26 Full Path

全路径可以更明确的告诉用户，当前所处的位置。

27. -gMIS 更新：getSelectOption, getTblRotateName, 附件管理, toExecl 等

Posted on 2015 年 1 月 7 日 by Wadelau

最近，为满足业务运营需求，又部署了几处 -gMIS 运行实例。
由于新的业务产生一些新的需求，对 -gMIS 进行了一些更新，记录如下。

add parentid in comm/tools.function.php
parentid 用于表述当前表的从属关系，在一些应用场景中，表与表之间，或者对象之间存在隶属关系时，会用到。当写完之后发现，parentid 这样的参数传递，其实可以使用 pnskid 即可，其中 pnsk 是在 PageNavigator 中定义的， page navigator searck key 的缩写。 pnskid 意思是当前请求中使用 id 作为条件之一. psnk 往往伴随着 oppnsk = operator of page navigator search key, 对检索条件的值的操作符，如大于、等于、小于、包含、不包含等等....
class/gtbl.class.php :
替换 THIS, THIS_由切割字符串换成 preg_match_all
简单的字符串拆解，为避免引入正则匹配引擎，一般采用 substr 等高效的计算，在复杂的环境中做字符串拆解，还是要 preg_match_all 这样的重型武器。
getSelectOption 增加对指定字段的支持，默认是 id
getSelectOption 允许在 -gMIS 中使用大量的动态字典表，通常情况下形成的列表都已 id 作为主键值，有些情况下可能需要其他字段的值，比如 name/code 等，案例：fwn_sitetbl, sitetype
getSelectOption 修正 dispname 可能以 – 开头
在导出到 Excel 时，如果当前 cell 的值以 – 开头，则容易出错
增加 getTblRotateName, 对滚动表名的支持
如 tbl_xxx_201412, tbl_xxx_201501 的共享 xml 的支持， 16:59 Monday, January 05, 2015

act/doaddmodi.php

增加使用 base62x 对文件名进行编码

为保留文件风貌，此前的附件处理时，一般保留文件的原名，增加一些防止重名的措施，但在一些非 UTF-8 运行环境，或者文件名中还有跨 OS 所致非法字符，使用 -base62x 对文件名进行编码，安全多了

增加对上传文件进行按月分目录存储

支持更大量级的附件存储

jdo.php

增加对 parent 参数转 pnsk 的代码

下载文件去掉路径的显示，使用 shortenStr 限定最长显示字符

act/toexecl.php

修正了输出格式，如果待输出的字段是字典表形式，则调用 getSelectOption 获取其表现值

class/pagenavi.class.php

修正 request 里，字段名称为 pnsk 情况，由 parentid 所致。

28. -gMIS 更新:字符串处理,op 字段及 github

Posted on 2015 年 2 月 13 日 by Wadelau

-gMIS 继续更新。

1. ./jdo.php

在操作列表增加 “打印”功能，可以从 list 页面直接呼出打印预览的页面；

修改 comm/ido.js 修正 doActSelect 函数使之配合完成上面的任务；

修改字符串处理函数，使之在 list 模式下对 HTML 的过滤更加智能，由替换 < 为 < 变成将 <.*?> 内容替换掉, str_replace 换成使用 preg_replace;

2. op 字段

修正 act/doaddmodi.php 中的 bug，使之能够记录当前操作人的身份到记录表；确认了 $opfield 的列表状态；

对应地，可以在 xml 中配置 op 字段为：

<field name=”op”>
<chnname>递交人</chnname>
<selectoption>fromtable::info_usertbl::email</selectoption>
<listview>1</listview>
<inputtype>select</inputtype>
</field>

3. 分层代码的使用

在新部署项目中，代码使用带有层级的风格，通用的字典表结构：

```
create table internet_typetbl(
  id int(11) not null auto_increment,
  icode char(16) not null default ”,
  iname char(32) not null default ”,
  iserial mediumint(4) not null default 0,
  op char(32) not null default ”,
  inserttime datetime not null default ’0000-00-00 00:00:00′,
  istate tinyint(1) not null default 1,
  primary key(id),
  unique index key2(iname)
);
```

其中 icode 是名称，iserial 是显示顺序。

在调用字典表的主表中，存储的值可能不再是值的 id，而是值的 icode，这样主要是为了能够方面的按父类检索、聚合（-R/k2Sl）。

4. -gMIS 迁入 -github

为便于管理，-gMIS 近日迁入 -github , 与此同时的还有 -GWA2 和 -GTAjax 也一并迁入 -github .

5. 在 -人民网 交流 -gMIS

2015 年 1 月，在人民网内部交流，以题目：-gmis 系统在业务运营中的应用

Figure 27 -gMIS in People.cn

29. -gmis 更新 201505，增加 linkfieldcopy

Posted on 2015 年 5 月 9 日 by Wadelau

linkfieldcopy 这一名词被引入是在 extra/linktbl.php 中需要将子表的一些字段抄送给父表。如在 xml/xxxx.tbl 中配置为：

<field name=”categorylist”>
<chnname>关联目录</chnname>
<extrainput>extra/linktbl.php?tbl=bxp_brandcategorytbl&linkfield=brandid
&pnskbrandid=THIS_id&pnsm=1&a
mp;linkfieldcopy=2</extrainput>
<!– linktbl_copyfield, 2 here means the second field in brandcategorytbl conf, ie. categoryid, refer to jdo.php
, 06:50 Saturday, May 09, 2015 –>
</field>

默认情况下，子表通过 comm/ido.js 中的 sendLinkInfo 调用，将列表页的第二列的值抄送到主表关联字段即 linkfield 所指的字段。但有些时候，问题可能需要抄送子表列表页的非第二个字段到 父表的 linkfield。

这时候就需要设置 linkfieldcopy 字段，能够指定一个顺序号，在 ido.php —> jdo.php 中调用，将相应的子表的列值抄送到父表的 linkfield 上。

如上的例子中，父表 categorylist 默认将抄送子表 brandid 这一列的值（顺序号为 1），如果指定了 linkfieldcopy=2，则子表的列表页 list 中的第二列的值将被送到到主表的 categorylist 中去.

涉及到的程序:
xml/xxxx.tbl
jdo.php
comm/tools.function.php

30. -gMIS 更新：增加 input2Select 功能(3)

Posted on 2015 年 10 月 14 日 by Wadelau

由于 -gMIS 中 input2Search / input2Select 的功能非常好用，此前先后进行了两次更新：

1. -gMIS 更新：增加 input2Select 功能， -R/l2Sn
2. -gMIS 更新：增加 input2Select 功能(2)，-R/s2Sx3

现在进行第三次更新，要解决的问题是，当待选/候选列表太长时，比如目前 -People-FwNews 中的站点名称列表，有 60K 记录，这样长的列表加载到 HTML 的 select 元素，用作 option 时，会影响到页面解析速度，在较好网络的情况下，需要 10s 左右的时间来处理。这需要优化。

改进的第一个尝试，是使用延时加载或异步加载，在 comm/ido.js 中增加 function lazyLoad, 当前主页面加载完毕之后，再加载这个超长的 60K 记录的 select 的 option list。

实测的结果是，当通过 -GTAjax 异步加载这 60K 的记录取回到页面前端，通过 JavaScript 写入 DOM 中的 select 的对应元素，仍需花费 6-8s 的时间。目前看来时间的花费主要在 DOM 中构建 select 的超长 option 元素。

继续改进，第二个尝试，将这个通过 lazyLoad, -GTAjax 异步加载到当前页面的内容，写入到 DOM 的 select 的 option 中去，修改为写入一个隐藏的 input 元素中，通过给 input.value 赋值的办法来实现将通过匿名回调函数取回的内容并入到当前 DOM 运行时环境。

同时继续修改，comm/ido.js 中的原来的 input2Search 函数，将新增的缓存 input.value 函数读取到进来，优先判断该值是否有用，如果有，则不再调用原来的 select 列表对象。

如此实现页面中运用 input2Search 的加速处理。

新增函数 lazyLoad:

```
//- lazy load long list, Wed Oct 14 09:08:51 CST 2015
function lazyLoad(myObj, myType, myUrl){
      window.console.log("lazyload is starting…. myurl:["+myUrl+"]
myobj:["+myObj+"]");
      if(true){
            //document.onreadystatechange = function(){
            //window.onload = function(){
            window.setTimeout(function(){
                  if(document.readyState == 'complete' ||
document.readyState == 'interactive'){
                        sendNotice(true, 'Lazy is loading in
process….. myobj:['+myObj+']');
                        var gta = new GTAjax();
                        gta.set('targetarea', 'addareaextradiv');
                        gta.set("callback", function(){

      //window.alert("getresult:["+this+"]");
                              var s = this;
                              var resultList = JSON.parse(s);
                              //var mySele =
document.getElementById(resultList.thefield);

      console.log("thefield:["+resultList.thefield+"] "+(new
Date()));
                              var optionList = {};
                              for(var
i=0;i<resultList.result_list.length;i++){
                                    var aresult =
resultList.result_list[i];
                                    //mySele.options[i] = new
Option(aresult.sitename+'('+aresult.id+')',aresult.id,
true,issel=false);
                                    optionList[aresult.id] =
aresult.sitename;
                              }
                              var myOptionList =
document.getElementById(resultList.thefield+'_optionlist');
                              myOptionList.value =
JSON.stringify(optionList);

      console.log("thefield:["+resultList.thefield+"]
completed……"+(new Date()));

      //console.log(JSON.stringify(myOptionList.value));
                              sendNotice(true, 'Lazy Load is done
successfully….. myobj:['+myObj+']');
                        });
                        gta.get(myUrl);
                  }
                  else{
                        sendNotice(false, 'Lazy Loading is
waiting…..');
                  }
            }, 3*1000);
      }
}
```

涉及到的其他内容:

1. 匿名函数传参数的问题

JavaScript 中匿名函数传参数是无法直接进行的，需要借助 bind 的机制，但测试未成功。实测的结果时，在第一层匿名函数，也即当前运行时环境中，首层设定匿名函数，可以直接调用其外部的变量，无需传递或声明；而二层匿名函数，也即在匿名函数中再设置一层匿名函数，则在其中无法直接使用其外部的变量。

2. JSON

借助 -GTAjax，设定了将当前页面异步读取一个接口文件，extra/readtblfield.php ，两者通过 JSON 格式的文件进行数据交换。JSON 在 -PHP 和 JavaScript 中的内置函数可以参考 -R/Y2Sx1 .

3. 局限

目前的优化，说针对的列表数据量级在 60K 左右，如果数据集继续增加到 100K，或者仍可以使用，但数据集增加到 1M 时，人需要考虑再次优化，再次优化的方向时，随着用户的输入即时调用后台查询接口，即时返回数据，然后再生成待选列表供用户点选，如同现在的搜索引擎的搜索提示一样。

之所以在这一步优化成将备选数据集一次性读取到前端，是因为这样在后续的处理中会更快，数据流量会更小，也减少了通信次数；这种优化主要针对有限数据集。

31.一种 debug 方法的实现

Posted on 2015 年 10 月 19 日 by Wadelau

据说程序的开发成本占 20%，维护成本占 80%，而维护的主要方式之一依靠的是 log 输出。

输出 log 有很多方式，各种开发语言都有不同的设施来满足这一功能。

在 PHP 中，可以这么实现一个 debug 功能。其主要功能是输出什么时间在什么位置（程序名、函数名）输出了标记为 xxx 的某个对象$obj 的打印形式的描述信息。

function debug($obj, $tag=null, $output=null);
# $obj 是待输出和跟踪的对象，可能是 string，也可能是 array、hash 等不同的数据结构；
# $tag 是用于标记待输出对象的字符串，一般用于过滤相关输出内容；默认 $tag 为 null;
# $output 类似 loglevel, 但又不同，通常 $output==null, 表示在后天输出相关日志，与 error_log 函数类似； $output==1 时，在后台和前端同事输出相关测试信息；$output==2 时，根据 PHP 中的 backtrace 信息，输出更多内容。
# 调用例子:
# debug($aHash); # 在后台输出一个 hash 变量的内容；使用 -GWA2 的 WebApp->toString 方法或者 -PHP 内置的 serialize ;
# debug($aHash, 'aHash'); # 在输出中增加识别标记字符 'aHash';
# debug($aHash, 'aHash', 2); # 同时在前端和后台输出相关内容，然后附带 -PHP 的 backtrace 信息.
源码 source：

```
# write log in a simple approach
# by wadelau@ufqi.com, Sat Oct 17 17:38:26 CST 2015
# e.g.
# debug($user);
# debug($user, 'userinfo');  # with tag 'userinfo'
# debug($user, 'userinfo', 1); # with tag 'userinfo' and in backend
and frontend
function debug($obj, $tag='', $output=null){
	$caller = debug_backtrace();
	if(is_array($obj) || is_object($obj)){
		if(isset($user)){
			$s .= $user->toString($obj);
		}
		else{
			$s .= serialize($obj);
		}
	}
	else{
		$s .= $obj;
	}

	if($tag != ''){
		$s = " $tag:[$s]";
	}
	$callidx = count($caller) - 2;
	$s .= ' func:['.$caller[$callidx]['function'].']
file:['.$caller[$callidx]['file'].']';

	if($output != null){
		if($output == 0){ # in backend only
			error_log($s);
		}
		else if($output == 1){ # in backend and frontend
			error_log($s);
			print $s;
		}
		else if($output == 2){ # in backend and frontend with
backtrace
			$s .= " backtrace:[".serialize($caller)."]";
			error_log($s);
			print $s;
		}
	}
	else{
		error_log($s); # default mode
	}

}
```

其他版本的语言也可参考此行为，自行实现。这一功能已经更新到新版的 -gMIS 和 -GWA2 的源码中.

32. 删除确认、删除异步及删除延时，delete confirm，aysnc and delay

Posted on 2015 年 11 月 27 日 by Wadelau

本文是对删除操作的一个优化的思考及实践的纪录。这里的删除也适合于更新等其他 update 操作。这里确认、异步和延时也适合其他非删除，但需要征询用户的情况。

首先要解决的问题是删除同步（sync），要变成删除异步（async）。当前在 -gMIS 中进行删除时，需要更新前端页面，办法时重新请求一下整个页面，这很不友好，也不节能，为了去掉一行，需要重新获取其他所有无关的数据再一次。这个可以使用异步的方法进行，就是在删除当前行时，不动其他的，在前端打个删去标记，通过 -GTAjax 给后台一个删除某行的命令即可。如果采用类似 JSON 的通信格式，则可以将通信成本控制到最低。
这里解决了两个问题，在服务器端减少了重新生成页面到各种计算，在通信层减少了传输信息量。在前端的影响几乎没有，已经通过 JavaScript 和 DOM 对待删除的纪录打了删除标记。在此之前 -gMIS 实现的 inline edit （行内编辑），就是出于这样的考虑, 现在终于可以爱删除这个动作上也应用上了。 扩大些，这个思维适合所有前端需要局部更新，单传统做法需要更新所有页面组件的情况。
这里的异步，更多的是从前端的视角，当前端还能够看到某条被删除的纪录时，后台已经没有了。

然后要对删除确认做点革命性的创新。其实，早就对删除确认（Confirm）不满了，所谓删除确认，是在操作系统、信息管理系统中，总是会在删除之前，要求用户进行确认才继续进行。偶尔遇见用一下还能够忍受。在使用 -gMIS 管理的 -people-fwnews 时，每天会遇到很多需要删除移除和修正的内容。觉得对于已经明确需要删除的内容再进行确认是某种浪费，不可忍。需要找到另外一种更人性，或者更舒服一些的做法。

传统的删除确认：

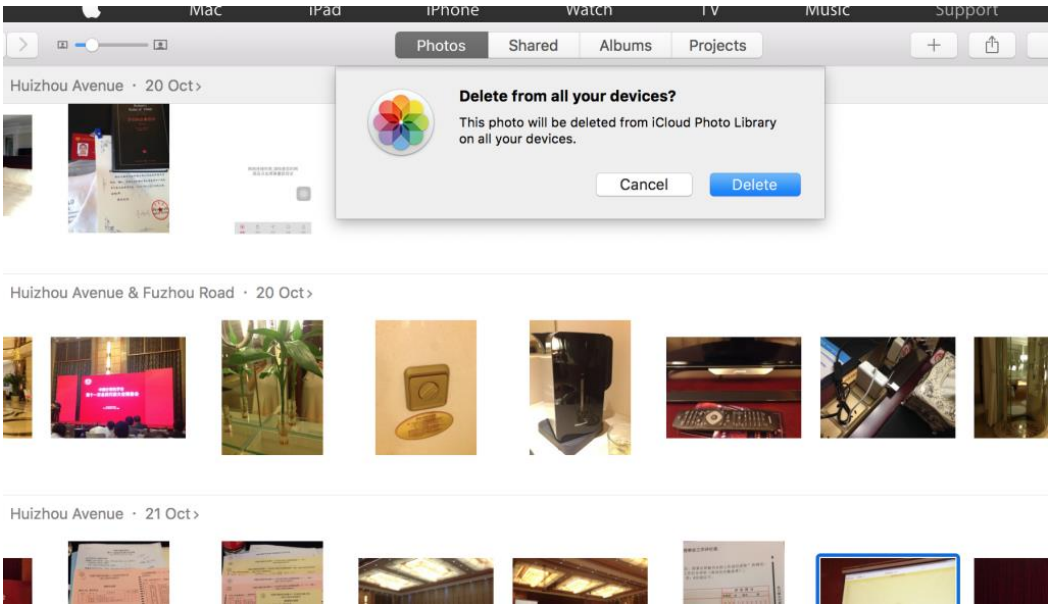

Figure 28 Delete in Mac OX

Mac OS X 的删除确认

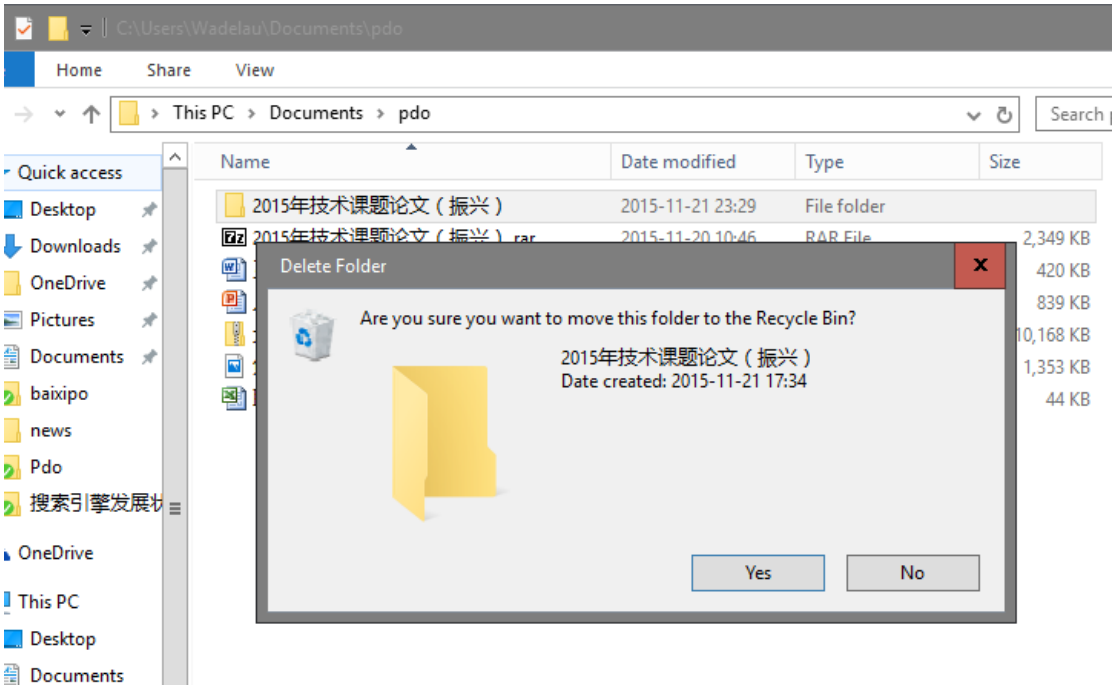

Figure 29 Delete in Windows 10

Windows 10 的删除确认

传统的删除确认在管理信息系统中的样子：

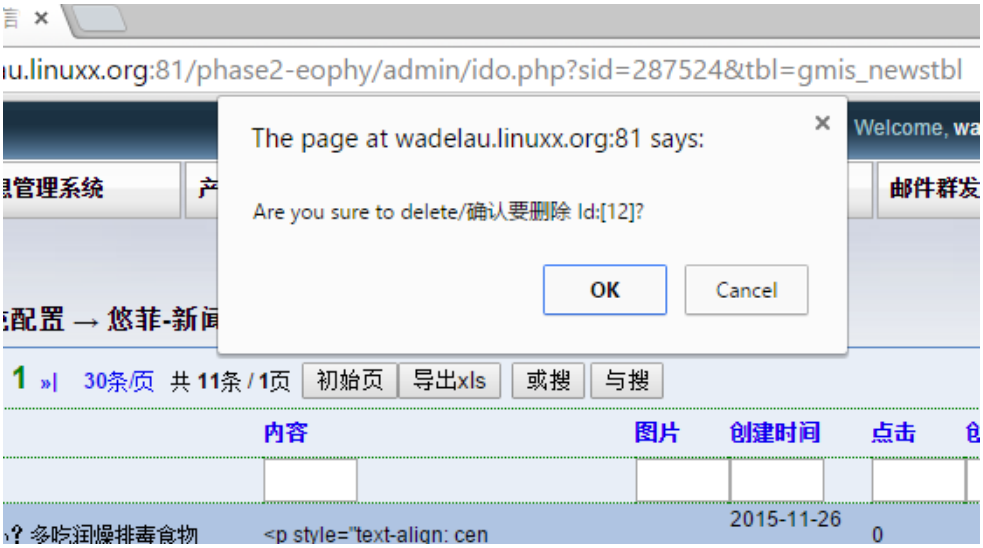

Figure 30 Conventional Delete

-gMIS 改造前的删除确认

通过前端或者后台生成一个确认对话框，用以对用户对提示，征询用户对资源纪录的移除动作的确认。这是常规的、靠谱地通常做法，偶尔用一下也不觉得有问题，用久了多了就会发现很费神。

但从另外一个角度说，如果不进行确认，删除之后无法进行找回，也是灾难。如何中庸地处理，有没有一种方法来进行中和？如果有新方法，应该能满足预期的三点：1）删除命令执行时，不需要用户二次交互，2）以某种方式警示用户进行删除，3）在第一步失误的情况下，可以回滚或终止。

在这种情况下，候选的方案是，变删除确（confirm）认为删除延时（delay），其步骤如下：

1. 删除命令（意愿）发出，
2. 给出提示，延迟一定时间后将实际执行删除操作，
3. 在延时期间，给出取消 / 终止删除操作的选项，
4. 当用户在延时倒计时期间，选择了取消 / 终止操作，则删除命令撤回，没有进一步地执行，
5. 当用户在延时倒计时期间，没有任何操作，则删除命令按预期进行。

这样，在删除一个对象时，用户只要交互一次即可完成最终删除，同时在发行删除有误时可以在延时期间随时取消删除行为。

对比如下两图，可以知道，传统的是阻塞模式，新的是非阻塞模式。

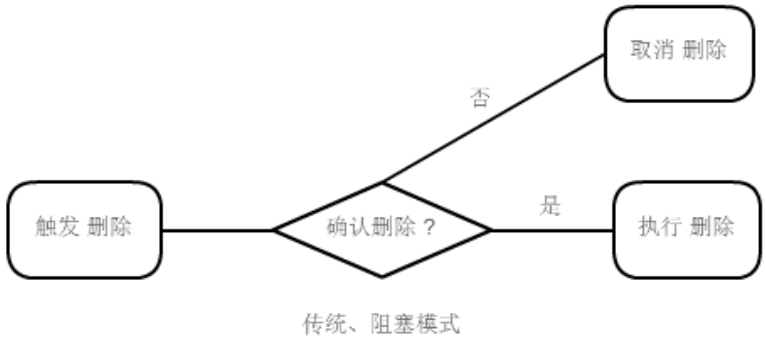

Figure 31 Delete routine

传统的阻塞模式的删除

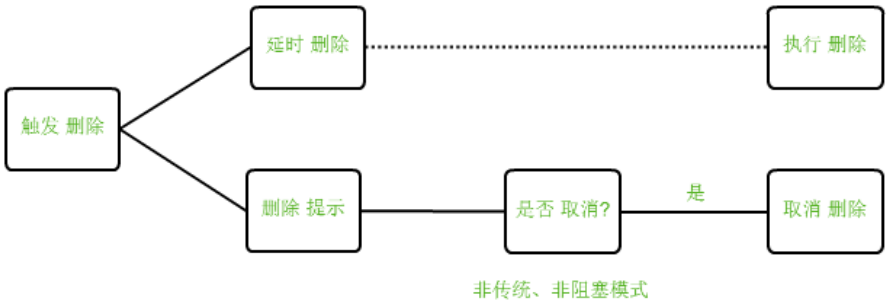

Figure 32 Non-block Delete

新的非阻塞模式的删除

改进后的删除延时使用实例：

Figure 33 Non-block delete instance

从此不用非得再点击一次，才能继续往下。

此想法已经汇报演示给技术大家 @ownhere，评价是更人性一些的改进，演示给人民网的同学 @wangchen，评价是 xxxx。
还需要一组数据来印证这种改进，在操作系统中，有多少次删除是进行了回滚？在管理信息系统，有多少次删除的确认中，用户选择了否？如果这两项的数据都表明，回滚和选择否的比例都很低很低，是否就不应该进行之前的设计，而应该应用新的删除延迟？

上文中，以 “-” 的拼写，都是 -NatureDNS ， -自然域名 。

33. -gMIS, -GWA2, -GTAjax 一并更新

Posted on 2016 年 1 月 24 日 by Wadelau

周末集中整块时间对 -gMIS ， -GWA2 和 -GTAjax 分别进行了小幅度的更新。

周末的北京是近年来少有的冷天。猫在家里写东西。

1. -gMIS
更新了所用组件 ido.js 在递送给 -GTAjax 的回调函数时，有未考虑对象为 null 的 JavaScript 片段，加上异常处理情况。
更新所用组件 -GTAjax。

2. -GTAjax
沉寂了好久的 -GTAjax 这个周末进行了小幅度更新，没有新增功能，而是增加了便于进行逐行调试的 track id。track id 的思路最早来自在 -ChinaM 时写 -Java 需要逐个模块的跟进在编译、运行和业务层面的程序分析，后来用在 -GWA2 和 -gMIS 等应用中。现在将这个也在 -GTAjax 中进行实现，有望在后续升级改进和 debug 时候能够进行快速而准确的定位。
这种定位，不单只是程序上的技术问题，有很多时候在业务层面的分析与判断也很有益于业务系统的改进与提高。

3. -GWA2

新功能增加，将原来默认不启用的 RESTful 地址功能改为默认启用。其具体实现是：

################## 关于 RESTful URL 地址风格的实现, Mon Oct 15 22:15:17 CST 2012, update Sun Jan 24 14:43:59 CST 2016

1. 资源访问路径中的？去掉

此前的: http://ufqi.com/dev/xxx/index.php?mod=web&act=preview&id=1234
RESTful 的: http://ufqi.com/dev/xxx/index.php/mod/web/act/preview/id/1234
RESTful 的: http://ufqi.com/dev/xxx/i/mod/web/act/preview/id/1234 ## use i as a soft link to index.php in server side

2. 规则，"?" 后面的参数，总是成对出现，奇数位的是参数名称，偶数位是参数值,

3. 实现：
在后台程序中，使用 $url."/para/value" 的样式拼合
在 Smarty 模板中，使用那个 {$url}/para/value 的样式拼合

在入口程序中，./index.php (i) 对 / 分割的参数重新转为 $_REQUEST 变量，同时重写

```
$_REQUEST['para'] = value;
$_SERVER['REQUEST_URI'];
$_SERVER['QUERY_STRING'];
```

4. 在其他程序中，与普通动态地址一样使用

5. TODO: 需要对 value 中的 "/" 做转义或者编码

6. 默认情况启用，since Sun Jan 24 14:00:51 CST 2016 ## !!!RESTful URL 地址风格在默认情况下不开启.

一个周末，同时更新三个软件。

34. Todo…