

通达应用自建平台

(2016 版)

帮助文档

(脚本编写指南)

通达应用自建平台组

2016 年 6 月 30 日

重要声明

- 一、感谢您购买中国兵器工业信息中心通达科技出品的通达应用自建平台产品，请在使用之前认真阅读本说明书。
- 二、用户帮助文档分为《通达应用自建平台帮助文档(初级用户)》、《通达应用自建平台帮助文档(高级用户)》以及《通达应用自建平台帮助文档（脚本编写指南）》，请用户根据具体情况进行阅读。本文档为《通达应用自建平台帮助文档(脚本编写指南)》。
- 三、本说明书随软件升级而升级，内容如有改动，恕不另行通知，请从指定的网站下载本说明书的最新版。
- 四、注册时所需要的软件注册码将另行提供。
- 五、请勿将本文档用于未经授权的出版、宣传等用途。
- 六、如有疑问请联系 619103385@qq.com，最新联系方式请参见 <http://www.tongda2000.com/bi/service/>

目 录

脚本编写必备基础.....	1
1.1 什么是脚本.....	1
1.2 编写脚本需要具备哪些基础.....	1
1.3 应用自建平台库表结构.....	1
自定义自动编号.....	3
1.4 概述.....	3
1.5 输入参数与输出结果.....	3
1.6 举例.....	3
自定义函数.....	8
1.7 概述.....	8
1.8 输入参数与输出结果.....	8
1.9 举例.....	8
自定义提取或回写引擎.....	12
1.10 概述.....	12
1.11 输入参数与输出结果.....	12
1.12 举例.....	13
公共函数.....	19
1.13 公共函数.....	19
自定义业务管理.....	21
1.14 概述.....	21
1.15 定义业务管理.....	21
客户服务.....	23
1.16 版本升级.....	23
1.17 远程咨询.....	23
1.18 定制服务.....	23
1.19 培训讲座.....	23
1.20 联系方式.....	23

脚本编写必备基础

1.1 什么是脚本

脚本是一段 PHP 代码，可以被应用自建平台执行以实现特定功能。应用自建平台支持通过脚本实现以下功能：

- 自定义自动编码规则。
- 自定义计算函数。
- 自定义提取公式。
- 自定义回写公式。

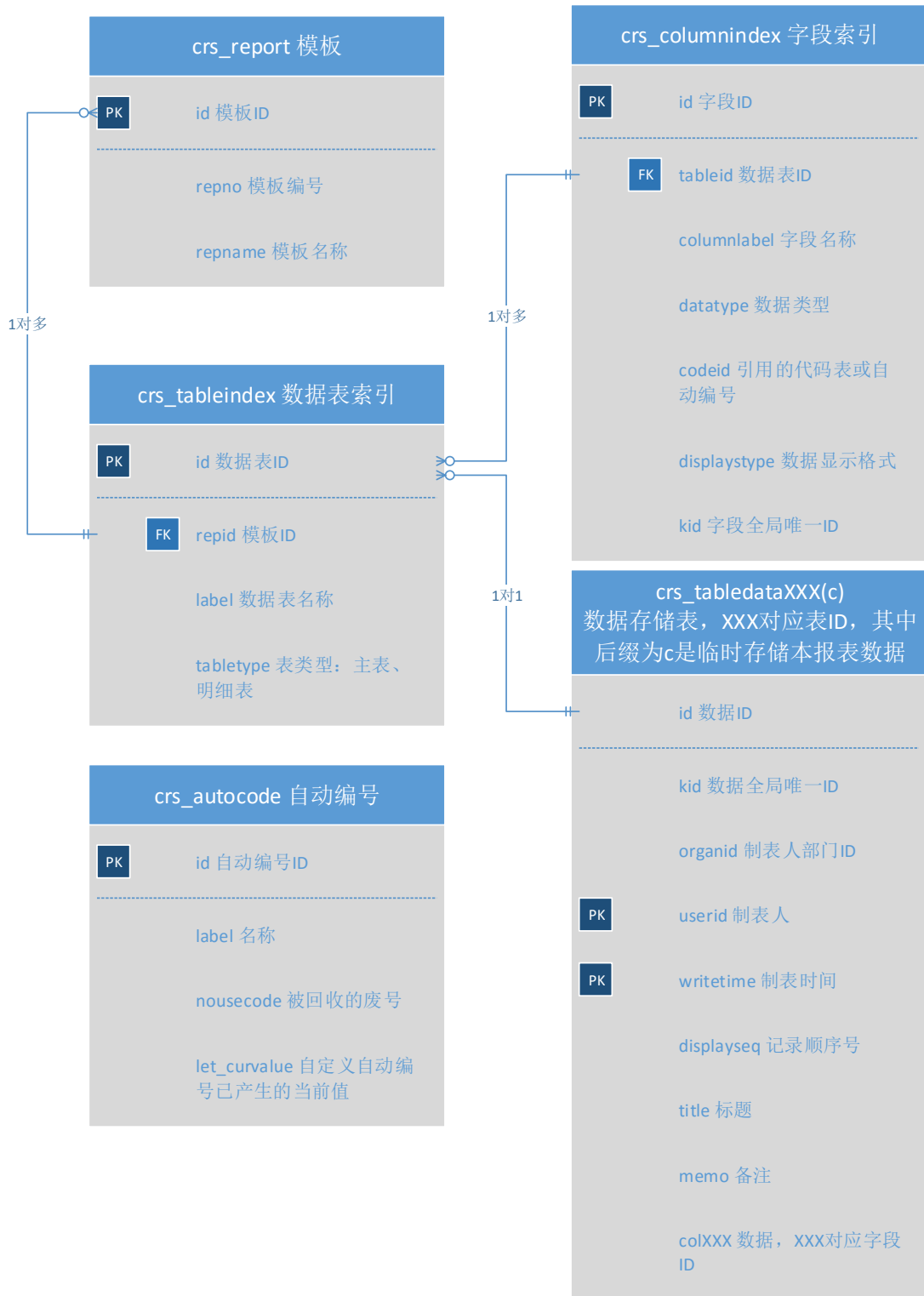
编写脚本对于应用自建平台来说不是必须的，但通过编写脚本却可以极大地增强应用自建平台的处理能力，这对于实施复杂度高、个别功能应用自建平台标准化功能满足不了的项目是一个很好的选择。

1.2 编写脚本需要具备哪些基础

除了要熟悉应用自建平台以及相关数据库表结构之外，编写脚本还需要有 PHP 编程基础，写脚本的过程本质上写 PHP 代码的过程。因此，应用自建平台的实施人员需要补充些 PHP 编程知识，这对编写脚本是非常有帮助的。

1.3 应用自建平台库表结构

应用自建平台主要库表结构如下图所属：



自定义自动编号

1.4 概述

定义自动编号时选择自定义选项，可实现自定义编号规则。定义自定义自动编号时需要遵循以下规则：

- 自定义规则引擎要以英文字段、数字以及下划线组合命名。
- 实现规则脚本文件为：引擎名称.php。
- 规则脚本文件存储路径为：webroot/general/reportshop/utils/autocode/。

1.5 输入参数与输出结果

应用自建平台传递给规则脚本文件的参数，即实现自定义编码规则的脚本文件输入参数包括：

\$action: 自动编号操作动作。动作为"generate"时表明要产生下一个自动编号；动作为"rollback"表明要回收一个自动编号。

\$row: 自动编号信息数组，包括 id（自动编号 ID）、nousecode（已产生但未用的编号，多个值用分号分隔）、let_curvalue（已产生的自动编号最大值，多个值逗号分隔）。

\$cur_rep: 本报表标识。

脚本文件传递给应用自建平台的处理结果，即实现自定义编码规则的脚本文件输出结果为：下一个自动编号，通过 echo 或 print 返回。

1.6 举例

以往来单位管理为例，举例说明如何通过脚本方式实现自定义自动编码规则。往来单位管理的单位包括供应商、代理商以及客户，为了使自动编号更具含义，我们需要根据不同单位类型分别编码，供应商以“GYS”为前缀，代理商以“DLS”为前缀，客户以“KEH”为前缀，后接同一单位类型 9 位流水号，即第一个供应商编号为 GYS000000001，第二个供应商编号为 GYS000000002；第一个代理商编号为 DLS000000001，第二个代理商编号为 DLS000000002，依次类推。

定义往来单位自动编号如下图：

名称：	往来单位编号	<input checked="" type="checkbox"/> 自定义
自定义规则引擎：	wldw	
新编号产生于：	保存报表	<input type="checkbox"/> 自动复位
重用原则：	不可重用	
<input type="button" value="保存修改"/> <input type="button" value="回收废号"/> <input type="button" value="重置"/>		

在 MYOA/webroot/general/reportshop/utils/autocode/目录下创建规则脚本文件 wldw.php，
如下图：



wldw.php 完整代码如下：

```
<?php
/**
 * 自定义引擎规则
 * 自动编号信息数组$row 包含: id(自动编号 ID)、nousecode(已产生但未用的编号, 多个值分号分隔)、
let_curvalue(当前值, 多个值逗号分隔)
 */
/**
 * 以下包含文件请不要更改
 */
if (!isset($PASSWORD)) {
    if (!empty($P)) {
        include_once('mobile/auth_mobi.php');
```

```

    } else {
        include_once("inc/auth.inc.php");
    }
} else {
    $PWD = md5(md5($PASSWORD));
    include_once("../task_auth.php");
}
/**
 * 以下代码负责生成自动编码
 */
if ($action == "generate") { /*自动编码的动作为生成*/
    /*根据客户类型这个字段的全局唯一 ID 获取对应表 ID 以及字段 ID。字段全局唯一 ID 请到对应模板
    的管理数据表》字段属性中查看*/
    get_tabid_and_colid_by_kid($i_tabid, $i_colid, "E0705CA9031549E3A0D38C50A259DC88");
    /*从本报表中取得指定字段的数据，这里是获取客户类型，客户类型为代码型，1 为供应商、2 为代
    理商、3 为客户*/
    $s_type = get_col_value($cur_rep, $i_tabid, $i_colid);
    /*从自动编号信息中判断是否有被回收的废号，如果有的话，可以重用之，多个废号之间用分号分隔
    */
    if (!empty($row["nousecode"])) {
        $s_curvalue = $row["nousecode"];
        $arr_curvalue = explode(";", $s_curvalue);
        $s_curvalue = "";
        foreach ($arr_curvalue as $s_v) {
            $s_prefix = substr($s_v, 0, 3);
            if ($s_type == 1 && $s_prefix == "GYS") {
                $s_curvalue = $s_v;
            } elseif ($s_type == 2 && $s_prefix == "DLS") {
                $s_curvalue = $s_v;
            } elseif ($s_type == 3 && $s_prefix == "KEH") {

```



```

        $s_curvalue = $s_v;
    }

    /*找到了可被重复使用的废号，更新下废号列表(nousecode)后，返回这个废号*/

    if (!empty($s_curvalue)) {

        $s_nousecode = $row["nousecode"];

        $s_nousecode = str_replace("$s_curvalue;", "", $s_nousecode);

        $s_nousecode = str_replace("$s_curvalue", "", $s_nousecode);

        $sql = "update crscell.crs_autocode set nousecode='$s_nousecode' where
id=$row[id]";

        exequery(TD::conn(), $sql);

        echo $s_curvalue;

        exit;

    }

}

}

/*如果没有可被重复利用的废号，那么我们分配一个自动编号*/

/*从 let_curvalue 中取得之前的已分配过的最大号*/

$s_curvalue = $row["let_curvalue"];

$i_curvalue = 0;

if (!empty($s_curvalue)) {

    $arr_curvalue = explode(",", $s_curvalue);

    foreach ($arr_curvalue as $s_v) {

        $arr_v = explode("_", $s_v);

        if ($arr_v[0] == $s_type) {

            $i_curvalue = $arr_v[1];

            break;

        }

    }

}

if ($i_curvalue > 0) { /*从 let_curvalue 中获取到之前已分配过的最大号*/

```

```

        $s_curvalue = str_replace($s_type . "_" . $i_curvalue, $s_type . "_" . ($i_curvalue +
1), $s_curvalue);
    } else { /*首次分配*/
        $s_curvalue .= $s_type . "_" . ($i_curvalue + 1) . ",";
    }
    /*更新下 let_curvalue*/
    $sql = "update crs_cell.crs_autocode set let_curvalue=' $s_curvalue' where id=$row[id]";
    exequery(TD::conn(), $sql);
    /*构造自动编号并返回*/
    if ($s_type == 1) {
        echo "GYS" . sprintf("%09d", ($i_curvalue + 1));
    } elseif ($s_type == 2) {
        echo "DLS" . sprintf("%09d", ($i_curvalue + 1));
    } elseif ($s_type == 3) {
        echo "KEH" . sprintf("%09d", ($i_curvalue + 1));
    }
}
?>

```

自定义函数

1.7 概述

应用自建平台提供了大量的标准函数,通过组合这些标准函数可以实现各种复杂度的数据处理。此外,应用自建平台还提供了自定义函数功能,方便用户自行扩充数据处理功能。

自定义函数语法如下:

自定义(<函数名>): 返回由<函数名>指定的自定义函数处理后的结果。例如: 自定义 (“get_aux_quantity1”)。

处理引擎命名规则: XXX 名称.func, 对应的实现文件为: XXX 名称.func.php, 例如: 自定义 (“get_aux_quantity1”)指定处理引擎为 get_aux_quantity.func, 对应的实现文件为: get_aux_quantity.func.php。

引擎实现文件存放路径为: webroot\general\reportshop\utils\plugins\模板编号, 例如: webroot\general\reportshop\utils\plugins\JH02。

注意: 自定义函数只能用在填充表达式中, 不能用在筛选条件中。

1.8 输入参数与输出结果

应用自建平台调用自定义函数时, 将给自定义函数传递以下参数(即自定义函数的输入参数):

`$_arr_row`: 公式执行结果集中的当前记录;

自定义函数输出结果为计算结果值。

1.9 举例

以单位换算为例, 举例说明如何使用自定义函数。首先, 定义如下提取公式:

[制单]时提取<JH01.基本信息,JH01.订货信息>中满足下列条件的数据:

<筛选条件:JH01.基本信息.单据编号=本报表.关联单据.进货订单 与 (JH01.订货信息.数量-JH01.订货信息.到货数量)>0>

<字段排序:JH01.订货信息.商品编号 升序>

按以下方式填入<本报表.进货信息>:

JH01.订货信息.商品编号 -(填入值) 本报表.进货信息.商品编号

JH01.订货信息.商品全名 -(填入值) 本报表.进货信息.商品全名

JH01.订货信息.单位 -(填入值) 本报表.进货信息.单位

JH01.订货信息.生产日期 -(填入值) 本报表.进货信息.生产日期

JH01.订货信息.批号 -(填入值) 本报表.进货信息.批号

自定义("get_aux_quantity1") -(填入值) 本报表.进货信息.辅助数量

自定义("get_aux_quantity2") -(填入值) 本报表.进货信息.辅助数量 1

自定义("get_aux_quantity3") -(填入值) 本报表.进货信息.辅助数量 2

JH01.订货信息.数量-JH01.订货信息.到货数量 -(填入值) 本报表.进货信息.数量

JH01.订货信息.单价 -(填入值) 本报表.进货信息.单价

JH01.订货信息.折扣 -(填入值) 本报表.进货信息.折扣

JH01.订货信息.税率 -(填入值) 本报表.进货信息.税率

JH01.订货信息.条码 -(填入值) 本报表.进货信息.条码

JH01.订货信息.辅助单位 -(填入值) 本报表.进货信息.辅助单位

自定义("get_basic_quantity") -(填入值) 本报表.进货信息.基本数量

其中，自定义("get_aux_quantity1")、自定义("get_aux_quantity2")、自定义("get_aux_quantity3")、自定义("get_basic_quantity")均为自定义函数。处理引擎定义为：

定义取数公式
✕

基本信息

公式名称: 任务阶段:

应用方式: 筛选条件改变后自动执行 保存时自动执行

初始填报时自动执行 手动执行

修改报表时自动执行

查阅报表时自动执行

显示方式: 显示全部查询数据 重复数据仅显示一次 显示 记录

仅提取已办结数据 处理引擎:

公式内容:

[制单]时提取<JH01.基本信息,JH01.订货信息>中满足下列条件的数据:
 <筛选条件:JH01.基本信息.单据编号=本报表.关联单据.进货订单 与 (JH01.订货信息.数量-JH01.订货信息.到货数)>0>
 <字段排序:JH01.订货信息.商品编号 升序>
 按以下方式填入<本报表.进货信息>:
 JH01.订货信息.商品编号->(填入值) 本报表.进货信息.商品编号
 JH01.订货信息.商品全名->(填入值) 本报表.进货信息.商品全名
 JH01.订货信息.单位->(填入值) 本报表.进货信息.单位
 JH01.订货信息.生产日期->(填入值) 本报表.进货信息.生产日期
 JH01.订货信息.批号->(填入值) 本报表.进货信息.批号
 自定义("get_aux_quantity1")->(填入值) 本报表.进货信息.辅助数量
 自定义("get_aux_quantity2")->(填入值) 本报表.进货信息.辅助数量1
 自定义("get_aux_quantity3")->(填入值) 本报表.进货信息.辅助数量2

引擎实现文件 `get_aux_quantity.func.php` 包含自定义 `get_aux_quantity1`，其脚本如下：

```

/**
 * 把基本数量转换为第一个单位所表示的数量
 * @param type $arr_row 计算结果当前记录
 * @return type 结果值
 */
function get_aux_quantity1($arr_row)
{
    /*从当前记录中获得相关数据*/

    $i_no_col = 0; $i_unit_col = 2;$i_qantity_col = 8;/*数组访问下标*/
    $s_no = $arr_row["$i_no_col"];
    $s_unit = $arr_row["$i_unit_col"];
    $f_quantity = $arr_row["$i_qantity_col"];

    /*根据商品编号获得度量单位以及对应的换算关系*/

    $arr_units = get_aux_unit_matrix($s_no);

```

```
/*转换数量*/  
$f_quantity = calculate_basic_quantity2($f_quantity, $s_unit, $arr_units);  
/*返回转换结果与单位名称组合*/  
return $f_quantity.$arr_units[0]["unit"];  
}
```

本公式涉及的其他自定义函数实现脚本从略。

自定义提取或回写引擎

1.10 概述

自定义函数仅在公式的局部实现了自定义处理逻辑。除此之外，应用自建平台还支持在整个公式范围内自定义处理逻辑。要实现整个公式范围内的自定义处理，需要满足以下条件：

如果是提取公式，填充方式里除了匹配之外，其他填充方式不得存在；如果需要用到其他填充方式，那么至少有一个填充表达式必须含有”PHPLET_”这个常量。

如果是回写公式，则无需定义回写字段。

自定义提取或者回写处理引擎命名规则为：XXX 名称，对应的实现文件为：XXX 名称.php，例如：处理引擎为 get_goods_lot，对应的实现文件为：get_goods_lot.php。

引擎实现文件存放路径为：webroot\general\reportshop\utils\plugins\模板编号，例如：webroot\general\reportshop\utils\plugins\JH02。

1.11 输入参数与输出结果

应用自建平台传递给规则引擎脚本文件的参数，即引擎文件输入参数包括：

- \$PLUGIN_CURSOR：数据集
- \$PLUGIN_REPID：模板 ID
- \$PLUGIN_APPTIME：应用方式
- \$PLUGIN_FUNC_POSITION：公式顺序号（从 1 开始）
- \$cur_rep：本报表标识
- \$cur_seq：当前记录顺序号

提取数据的引擎处理输出结果规则为：XXX 值->表名称#_#字段名称，不同字段用英文逗号风格，不同记录用英文分号风格，以下为有效返回结果：“12->明细表#_#单价,”，” 12->明细表#_#单价,13->明细表#_#数量,”，“12->明细表#_#单价,13->明细表#_#数量,; 22->明细表#_#单价,23->明细表#_#数量,;”。回写数据的引擎处理文件不输出结果。输出的结果拼接到 \$res 变量中，如：\$res :=“12->明细表#_#单价,13->明细表#_#数量,; 22->明细表#_#单价,23->明细表#_#数量,”。

注意：由于自定义提取或处理引擎是被嵌入到应用自建平台里一起执行的，因此引擎里使用的变量尽可能加前缀“plugin_”并采用小写形式，例如\$plugin_i、\$plugin_str、\$plugin_arr_datas等，以区别于应用自建平台里所使用的变量。

1.12 举例

以获取产品批次列表为例，举例说明如何使用引擎实现自定义提取逻辑，提取产品批次公式定义如下：

定义取数公式

基本信息

公式名称: 任务阶段:

应用方式: 筛选条件改变后自动执行 保存时自动执行
 初始填报时自动执行 手动执行
 修改报表时自动执行
 查阅报表时自动执行

显示方式: 显示全部查询数据 重复数据仅显示一次 显示 记录
 仅提取已办结数据 处理引擎

公式内容:

<筛选条件:本报表.基本信息.收货仓库 有值>
按以下方式填入<本报表.进货信息>:
本报表.进货信息.商品编号 ->(匹配) 本报表.进货信息.商品编号
列表(PHPLET_批号, 本报表.进货信息.批号, PHPLET_生产日期, 本报表.进货信息.生产日期, PHPLET_库存里) ->(构造下拉列表(允许其他值)) 本报表.进货信息.批号

1.数据源
2.筛选条件
3.填充方式

确认
关闭

处理引擎定义为“get_goods_lot”，填充方式里由于用到了“构造下拉列表（允许其他值）”，因此填充表达式包含了常量“PHPLET_”。

引擎脚本文件 get_goods_lot.php 包含以下内容：

<?

/**

* 获得商品批次信息


```

* @param type $PLUGIN_CURSOR 计算结果集
* @param type $s_tabname 待填充的表名称
* @param type $_i_tabid 待填充的表 id
* @param type $i_no_col 商品编号字段 id
* @param type $i_mtabid 仓库字段所在表 id
* @param type $i_warecol 仓库字段 id
* @param type $cur_rep 本报表标识
* @param type $cur_seq 当前记录顺序号
* @return string
*/

function get_goods_lots($PLUGIN_CURSOR, $s_tabname, $_i_tabid, $i_no_col, $i_mtabid, $i_warecol,
$cur_rep, $cur_seq)
{
    /*取得需要用到的字段所对应的表 ID 和字段 ID*/

    $s_rksl = "{674A91EE-8F01-44A5-B3E1-E3F1BCD8E2B4}"; //入库数量
    get_tabid_and_colid_by_kid($_i_tabid, $i_rksl, $s_rksl);

    $s_rkcb = "{CAEE856E-E164-4AE5-B3B5-B1AF61F7996C}"; //入库成本
    get_tabid_and_colid_by_kid($_i_tabid, $i_rkcb, $s_rkcb);

    $s_cks1 = "{979BAA40-6138-4D3E-A2A1-B1D882C93E1D}"; //出库数量
    get_tabid_and_colid_by_kid($_i_tabid, $i_cks1, $s_cks1);

    $s_ckcb = "{5B105508-65DE-4A78-8594-CFBFA56AD32E}"; //出库成本
    get_tabid_and_colid_by_kid($_i_tabid, $i_ckcb, $s_ckcb);

    $s_ck = "{F7A1A24F-D3D9-4475-8B4A-BB601457FB13}"; //仓库
    get_tabid_and_colid_by_kid($_i_tabid, $i_ck, $s_ck);

    $s_spbh = "{28F2BBB2-FCB8-4C8F-846F-0F15799386A2}"; //商品编号
    get_tabid_and_colid_by_kid($_i_tabid, $i_spbh, $s_spbh);

    $s_scrq = "{664BE396-2BC3-4089-BC98-FD10FA98A15D}"; //生产日期
    get_tabid_and_colid_by_kid($_i_tabid, $i_scrq, $s_scrq);

    $s_pc = "{60E369BA-5220-4851-AB61-236B79A15E4D}"; //批号
    get_tabid_and_colid_by_kid($_i_tabid, $i_pc, $s_pc);

```

```

$s_ywsj = "{298D97C6-AAD6-43A4-9767-050F5ED25D84}"; //业务时间

get_tabid_and_colid_by_kid($i_tabid, $i_ywsj, $s_ywsj);

$PLUGIN_RES = "";

/*根据计算结果集取得计算结果数组*/

if($PLUGIN_ROW = mysql_fetch_array($PLUGIN_CURSOR))
{
    $s_plugin_val = $PLUGIN_ROW[0]; /*含有 PHPLET_字样的批次信息*/

    $s_warehouse = get_col_value($cur_rep, $i_mtabid, $i_warecol); /*取得仓库信息*/

    $s_goods = get_col_value($cur_rep, $i_i_tabid, $i_no_col, $cur_seq); /*取得商品编号*/

    $plugin_sql = "select col$i_pc as 'pc', if(col$i_scrq='0000-00-00','',col$i_scrq) as
'rq', sum(col$i_rksl -col$i_cks1) as kc from crscell.crs_tabledata$i_tabid
        where col$i_ck=' $s_warehouse' and col$i_spbh=' $s_goods' and (if(col$i_pc is
null,'',col$i_pc)<>' ' or (if(col$i_scrq is null,'',col$i_scrq)<>' ' and
col$i_scrq<>'0000-00-00')) group by pc, rq order by pc, rq";

    $plugin_cursor = exequery(TD::conn(), $plugin_sql); /*提取当前商品的批次信息*/

    while ($plugin_row = mysql_fetch_array($plugin_cursor))
    {
        /*把含有 PHPLET_字样的批次信息，转换为实际的批次信息*/

        $s_plugin_val2 = str_replace("PHPLET_批号", "$plugin_row[pc]", $s_plugin_val);

        $s_plugin_val2 = str_replace("PHPLET_生产日期", "$plugin_row[rq]",
$s_plugin_val2);

        $s_plugin_val2 = str_replace("PHPLET_库存量", $plugin_row["kc"], $s_plugin_val2);

        /*结果值如果不是简单的字面量，需要用 HandleMemoStr 进行编码处理。一个商品的批次信
息可能很多，因此用分号分隔，每个批次信息对应一条记录*/

        $PLUGIN_RES .= HandleMemoStr($s_plugin_val2) . "->$s_tabname#_#批号,;";

    }
}

return $PLUGIN_RES;
}

```

```

/*根据表名取得表 ID*/

$i_plugin_mtabid = get_tabid_by_name($PLUGIN_REPID, "基本信息");

$i_plugin_tabid = get_tabid_by_name($PLUGIN_REPID, "进货信息");

/*根据字段名取得字段 ID*/

$i_plugin_goods_no_colid = get_colid_by_name($i_plugin_tabid, "商品编号");

$i_plugin_warehouse_colid = get_colid_by_name($i_plugin_mtabid, "收货仓库");

$res .= get_goods_lots($PLUGIN_CURSOR, "进货信息", $i_plugin_tabid, $i_plugin_goods_no_colid,
$i_plugin_mtabid, $i_plugin_warehouse_colid, $cur_rep, $cur_seq);

?>

```

以回写进货订单为例，举例说明如何使用引擎实现自定义回写逻辑，自定义回写进货订单公式定义如下：

上述公式使用了处理引擎 update_jh01，且回写表达式都空白（未定义）。

引擎脚本文件 update_jh01.php 包含的内容如下：

```
<?php
```

```

/*获得需要用到的字段所对应的表 ID 和字段 ID*/
get_tabid_and_colid_by_kid($i_tabid, $i_42932, "{233032E1-5FF7-4E07-8B2B-C52F8B331BAD}");
get_tabid_and_colid_by_kid($i_tabid, $i_43369, "{8F007245-248B-4DD0-860A-4D2FB38DCA65}");
get_tabid_and_colid_by_kid($i_tabid, $i_45314, "{FD550EB9-491A-4412-93DD-359810CCE81D}");
get_tabid_and_colid_by_kid($i_tabid2, $i_41991, "{76C979D7-7226-4CFE-9748-D99CD3192456}");
get_tabid_and_colid_by_kid($i_tabid2, $i_42001, "{D3EED76F-12F0-464C-9DBB-6BD84F2DD41E}");

/*打开结果集*/

if($PLUGIN_ROW = mysql_fetch_array($PLUGIN_CURSOR))
{
    /*取得进货订单编号*/

    $plugin_book_sn = $PLUGIN_ROW["$i_col42932"];

    /*取得进货订单中待回写记录（某个进货订单）的制单人和制单时间，注：制单人和制单时间是记录
的主键信息*/

    $plugin_userid = $PLUGIN_ROW[3];

    $plugin_writetime = $PLUGIN_ROW[4];

    /*计算待回写数据的订货量和完成量*/

    $plugin_sql = "select sum(col$i_41991) as a, sum(col$i_42001) as b from
crscell.crs_tabledata$i_tabid2 where userid=' $plugin_userid' and
writetime=' $plugin_writetime'";

    $plugin_cursor = exequery(TD::conn(), $plugin_sql);

    if($plugin_row = mysql_fetch_array($plugin_cursor))
    {

        $plugin_finished = "";/*完成标识*/

        /*进货量不超过完成量时，表明该订单已经完成，完成标识打勾*/

        if($plugin_row["a"] <= $plugin_row["b"])
        {

            $plugin_finished = "√";

        }

        /*设置进货订单完成标识*/

        $plugin_sql = "update crscell.crs_tabledata$i_tabid set col$i_43369=' $plugin_finished',

```

```
col$i_45314=' $plugin_row[b]' where userid=' $plugin_userid' and writetime=' $plugin_writetime'";  
    exequery(TD::conn(), $plugin_sql);  
  }  
}  
?>
```

公共函数

1.13 公共函数

在编写脚本时，应用自建平台提供了一些可供直接调用的公共函数：

```
/**
 *作用：根据表 ID 以及字段 ID，返回本报表指定字段数据。
 * @para $cur_rep: 本报表标识
 * @para $i_tabid: 表 ID
 * @para $i_colid: 字段 ID
 * @para $i_seq: 第几条记录（从 1 开始）
 * @return mixed 字段数据
 **/

function get_col_value($cur_rep, $i_tabid, $i_colid, $i_seq = 1)

/**
 *作用：根据模板编号返回模板 ID
 * @para $RepNo: 模板编号
 * @return String 模板 ID
 **/

function GetRepIdByNo($RepNo)

/**
 *作用：根据模板 ID 返回模板编号
 * @para $RepId: 模板 ID
 * @return String 模板编号
 **/

function GetRepNoById($RepId)

/**
 * 根据字段 KID 获得表 ID 和字段 ID
 * @param type $i_tabid 表 ID（输出）
```

```

* @param type $i_colid 字段 ID (输出)
* @param type $s_kid 字段 KID (字段全局唯一 ID)
* @return boolean
*/

function get_tabid_and_colid_by_kid(&$i_tabid, &$i_colid, $s_kid)
/**
* 根据模板 ID 以及表名称返回表 ID
* @param type $i_repid 模板 ID
* @param type $s_name 表名称
* @return int 表 ID
*/

function get_tabid_by_name($i_repid, $s_name)
/**
* 根据表 ID 以及字段名称返回字段 ID
* @param type $i_tabid 表 ID
* @param type $s_name 字段名称
* @return int 字段 ID
*/

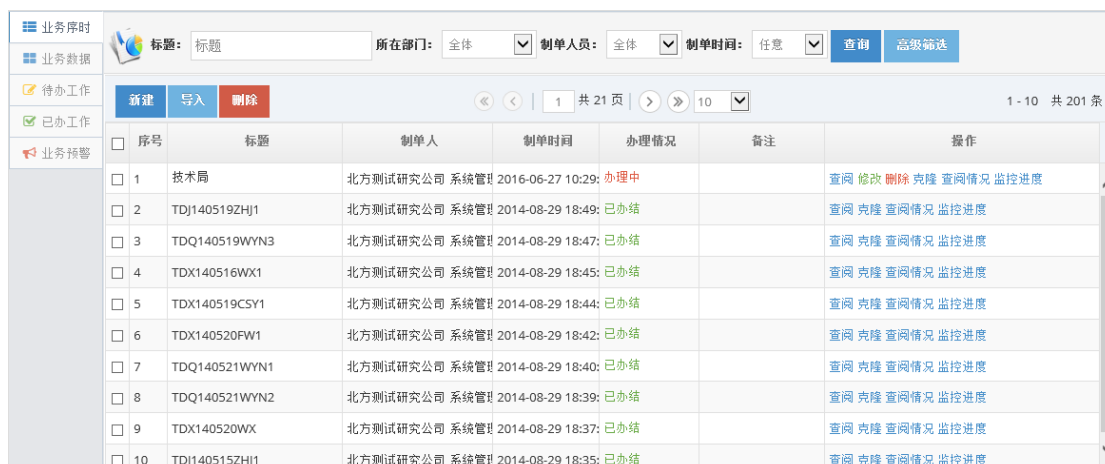
function get_colid_by_name($i_tabid, $s_name)

```

自定义业务管理

1.14 概述

应用自建平台提供了标准化的业务管理模式，如下图所示：



| 序号 | 标题 | 制单人 | 制单时间 | 办理情况 | 备注 | 操作 |
|----|---------------|--------------|------------------|------|----|-----------------------|
| 1 | 技术局 | 北方测试研究公司 系统管 | 2016-06-27 10:29 | 办理中 | | 查阅 修改 删除 克隆 查阅情况 监控进度 |
| 2 | TDJ140519ZHJ1 | 北方测试研究公司 系统管 | 2014-08-29 18:49 | 已办结 | | 查阅 克隆 查阅情况 监控进度 |
| 3 | TDQ140519WYN3 | 北方测试研究公司 系统管 | 2014-08-29 18:47 | 已办结 | | 查阅 克隆 查阅情况 监控进度 |
| 4 | TDX140516WX1 | 北方测试研究公司 系统管 | 2014-08-29 18:45 | 已办结 | | 查阅 克隆 查阅情况 监控进度 |
| 5 | TDX140519CSY1 | 北方测试研究公司 系统管 | 2014-08-29 18:44 | 已办结 | | 查阅 克隆 查阅情况 监控进度 |
| 6 | TDX140520FW1 | 北方测试研究公司 系统管 | 2014-08-29 18:42 | 已办结 | | 查阅 克隆 查阅情况 监控进度 |
| 7 | TDQ140521WYN1 | 北方测试研究公司 系统管 | 2014-08-29 18:40 | 已办结 | | 查阅 克隆 查阅情况 监控进度 |
| 8 | TDQ140521WYN2 | 北方测试研究公司 系统管 | 2014-08-29 18:39 | 已办结 | | 查阅 克隆 查阅情况 监控进度 |
| 9 | TDX140520WX | 北方测试研究公司 系统管 | 2014-08-29 18:37 | 已办结 | | 查阅 克隆 查阅情况 监控进度 |
| 10 | TDJ140515ZHJ1 | 北方测试研究公司 系统管 | 2014-08-29 18:35 | 已办结 | | 查阅 克隆 查阅情况 监控进度 |

如果标准化管理模式满足不了个性化的管理需求，那么我们可以通过自定义业务管理来实现个性化管理。


1.15 定义业务管理

在应用管理》设计中心》管理，找到具体业务，然后点击“设置”，如下图：



在设置界面里可定义实现业务管理的页面路径，如下图：

×

| | |
|-------|--|
| 视图模式： | 表格 表单 控件 |
| 旧密码： | <input type="password"/> |
| 新密码： | <input type="password"/> |
| 再次输入： | <input type="password"/> |
| 业务路径： | <code>/general/reportshop/workshop/modules/jxcdj/manage.php</code> |
| 图标： |  |

确认

在应用中心访问该业务时将打开业务路径所指定页面，并给该页面传递以下参数：

- repid, 即模板 ID。

注意: 为了同一权限管理的需求, 自定义业务最好放在 `/general/reportshop/workshop/` 目录下。

客户服务

1.16 版本升级

通达应用自建平台自购买之日起,根据购买版本的不同提供一年或者终身免费服务与升级。

1.17 远程咨询

提供电话、Email 等方式的远程咨询,咨询联系方式见网站:

<http://www.tongda2000.com/bi/service/>

1.18 定制服务

可以根据客户需求定制业务管理系统,收费标准请以电话、Email 等方式垂询,具体开发费用视开发周期和难度决定。

1.19 培训讲座

我们可以根据用户要求,为用户提供有偿培训和讲座,具体事宜面议。

1.20 联系方式

通讯地址:北京市海淀区紫竹院路 69 号中国兵器大厦 508 室

单位名称:北京通达信科科技有限公司

邮编: 100089

Email: 619103385@qq.com

QQ: 619103385

QQ 群: 40181741、124796420

官方网址: <http://www.tongda2000.com/bi>