

通达应用中心

(2017 版)

使用手册

(脚本编写指南)

通达应用中心组

2017 年 11 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中心产品，请在使用之前认真阅读本说明书。
- 二、用户帮助文档分为《2017 版全新应用中心使用手册》以及《2017 版全新应用中心使用手册（脚本编写指南）》，请用户根据具体情况情况进行阅读。本文档为《2017 版全新应用中心使用手册（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、最新联系方式请参见 <http://www.tongda2000.com/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 应用中心库表结构..... | 1 |
| 1.4 应用中心控件数据结构..... | 2 |
| 1.5 公共函数..... | 4 |
| 1.5.1 获得当前表单 id..... | 4 |
| 1.5.2 获得当前表单 guid..... | 4 |
| 1.5.3 获得表单 guid..... | 5 |
| 1.5.4 获得字段 guid..... | 5 |
| 1.5.5 根据名称获得当前字段值..... | 5 |
| 1.5.6 获得当前字段值..... | 5 |
| 1.5.7 根据名称获得表单基本数据（非列表数据）或者列表数据..... | 5 |
| 1.5.8 获得表单基本数据（非列表数据）或者列表数据..... | 6 |
| 1.5.9 设置字段数据..... | 6 |
| 1.5.10 设置字段选项数据..... | 6 |
| 1.5.11 清除列表数据..... | 6 |
| 1.5.12 获得应用中心数据库连接..... | 6 |
| 1.5.13 获得数据源里配置的数据库连接..... | 7 |
| 1.5.14 根据表单名称获得表单 id..... | 7 |
| 1.5.15 根据表单 guid 获得表单 id..... | 7 |
| 1.5.16 根据字段 GUID 获得字段 ID..... | 8 |
| 1.5.17 根据字段名称获得字段 ID..... | 8 |
| 自定义函数..... | 9 |
| 1.6 概述..... | 9 |
| 1.7 输入参数与输出结果..... | 9 |
| 1.8 举例..... | 9 |
| 自定义提取或回填引擎..... | 12 |

| | | |
|-----------|----------------|----|
| 1.9 | 概述..... | 12 |
| 1.10 | 输入参数与输出结果..... | 12 |
| 1.11 | 举例..... | 12 |
| 客户服务..... | | 17 |
| 1.12 | 版本升级..... | 17 |
| 1.13 | 远程咨询..... | 17 |
| 1.14 | 定制服务..... | 17 |
| 1.15 | 培训讲座..... | 17 |
| 1.16 | 联系方式..... | 17 |

脚本编写必备基础

1.1 什么是脚本

脚本是一段 PHP 代码，可以被应用中心执行以实现特定功能。应用中心支持通过脚本实现以下功能：

- 自定义计算函数。
- 自定义提取触发器。
- 自定义回填触发器。

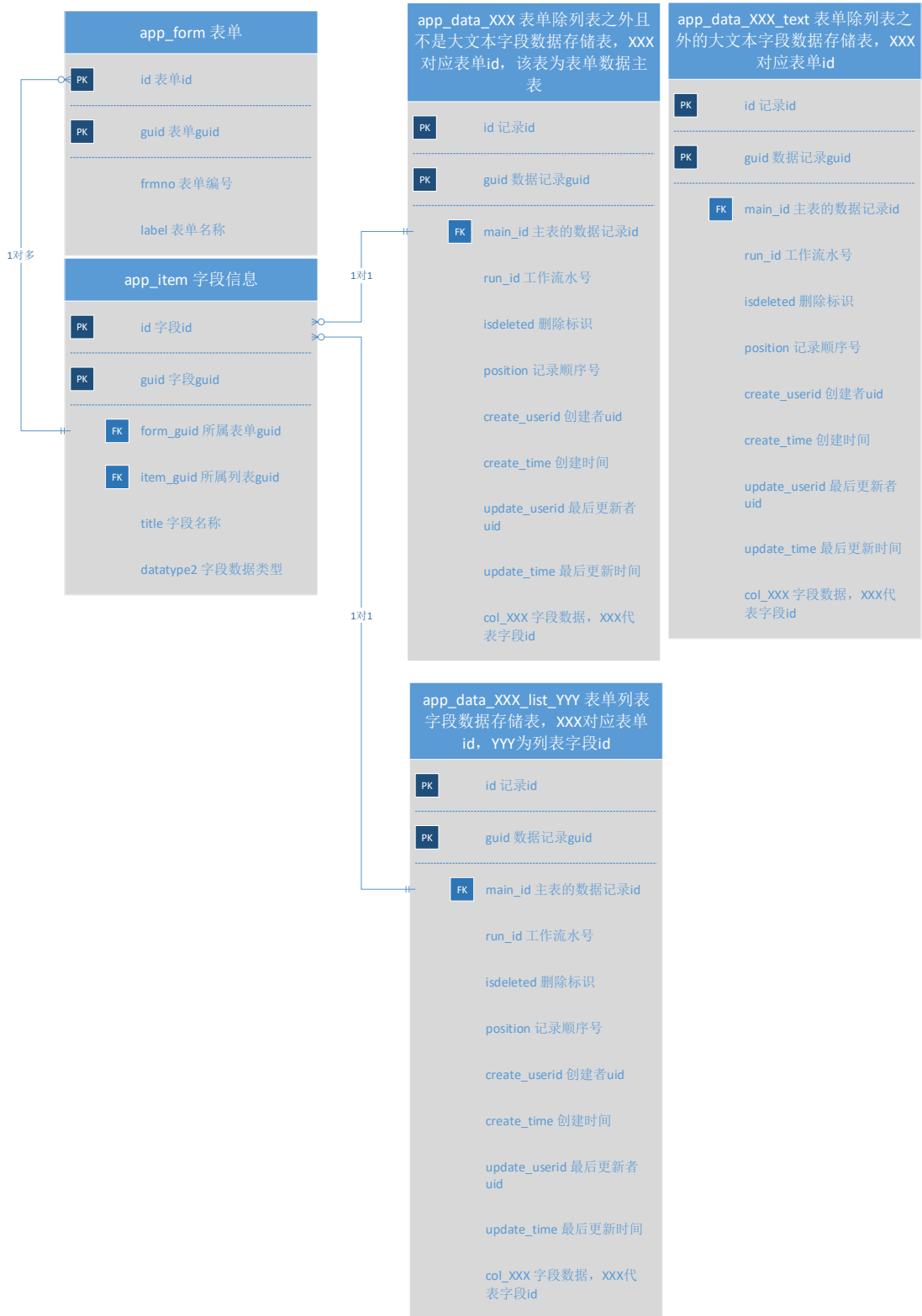
编写脚本对于应用中心来说不是必须的，但通过编写脚本却可以极大地增强应用中心的处理能力，这对于实施复杂度高、应用中心标准化功能满足不了的项目是一个很好的选择。

1.2 编写脚本需要具备哪些基础

除了要熟悉应用中心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，应用中心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 应用中心库表结构

应用中心主要库表结构如下图所属：



1.4 应用中心控件数据结构

新版应用中心采取前后端分离方式实现，前端和后端采取规定好的数据结构进行通信，因此用户自己编写脚本过程中读取和设置数据时要满足规定好的结构。应用中心各控件数据

结构说明如下：

单行文本、多行文本、数字文本、货币文本、自动编号：具体值，例如：“应用中心”，“2017版\n应用中心操作手册”，96，25.98，“2017110056”。

日期文本：类型为日期时，数据格式为 yyyy-M-d，例如：2017-11-20；类型为时间时，数据格式为 HH:mm:ss，例如：11:07:01；类型为日期+时间时，数据格式为 yyyy-M-d HH:mm:ss，例如：2017-11-20 11:07:01。

单选框：选项值结构为：[[“code”,“name”]]，code 为代码保存值，name 为代码显示值，例如：[[“code”=>“01”,“name”=>“男”],[“code”=>“02”,“name”=>“女”]]；选定值为：选择后的代码 code 值，例如：01。

多选框：选项值结构与单选框一样；选定值为：数组，数组里元素为选择后的代码 code 值，例如：[“01”,“02”]。

下拉选择：选项值结构与单选框一样；不支持多选的选定值结构与单选框一致，支持多选的选定值结构与多选框一致。

地 址：数 据 结 构 为：
[[“prov”=>[[“id”,“name”],“city”=>[[“id”,“name”],“country”=>[[“id”,“name”],“street”]], 分别对应省、市、县以及详细地址，例如：[[“prov”=>[[“id”=>“110000”,“name”=>“北京市”],“city”=>[[“id”=>“110100”,“name”=>“市辖区”],“country”=>[[“id”=>“110108”,“name”=>“海淀区”],“street”=>“紫竹院路 69 号”]]]]。

定位：数据结构为：[[“lat”,“lng”,“address”]]，分别对应纬度、经度以及具体地址，例如：[[“lat”=>39.9241,“lng”=>116.404,“address”=>“北京市东城区”]]。

附件、图片：数据结构为：[[“attach_id”,“attach_name”]]，分别对应附件 ID 以及附件名称，例如：[[“attach_id”=>“2873@1711_754692181”,“attach_name”=>“报表需求开发.docx”],[“attach_id”=>“2874@1711_303458136”,“name”=>“17 版应用中心建议.docx”]]。

部门选择：数据结构为：[[“dept_id”,“dept_name”]]，分别对应部门 ID 以及部门名称，例如：[[“dept_id”=>“ALL_DEPT”,“dept_name”=>“全体部门”]]，[[“dept_id”=>1,“dept_name”=>“总经理”],[“dept_id”=>2,“dept_name”=>“财务部”]]。

人员选择：数据结构为：[[“uid”,“username”]]，分别对应人员的 UID 以及人员真实姓名，例如：[[“uid”=>1,“username”=>“张三”],[“uid”=>2,“username”=>“李四”]]。

列表：数据结构为：[[“flag”,“index”,“data”=>[[“field_id”,“value”]]]]，解释如下：

flag: 新增行标识，新增行 flag 值为 new，非新增行时为空；

index: 行 id;

data: 记录数据, 是一个二维数组, 每列数据包含 field_id 和 value, field_id 为该列字段的 GUID, value 为该列的数值, 数值结构参考之前的描述。

列表数据例子如下:

```
[
["flag"=>"new","index"=>"
1511236728561","data"=>[[{"field_id"=>"{4C022D90-83C4-5EBA-EE7B-ED1A98AFD7ED}","value"=>
2],[{"field_id"=>"
{6FF2B352-A9D9-C1EC-414C-795016C2141D}","value"=>[11,12]]],
["flag"=>"new","index"=>"
1511236715995",
"data"=>
[[{"field_id"=>"{4C022D90-83C4-5EBA-EE7B-ED1A98AFD7ED}","value"=>1],[{"field_id"=>"
{6FF2B352-A9D9-C1EC-414C-795016C2141D}","value"=>[5,1]]]]
]
```

1.5 公共函数

系统内置了一系列公共函数可供自定义函数、自定义脚本插件使用, 这些函数封装在一个名为 APPPluginUtils 的类里, 在运行时这个类会实例化为 \$PLUGIN_UTILS 对象, 并以实参或者变量形式传递给自定义函数、自定义插件。具体调用方式请参考后面例子。

1.5.1 获得当前表单 id

方法: getFormId()

输入: 无

输出: 当前表单 id

1.5.2 获得当前表单 guid

方法: getFormGuid()

输入: 无。

输出: 当前表单 guid。

1.5.3 获得表单 guid

方法: `getFormGuidById($i_id)`

输入: `$i_id` 表单 id。

输出: 指定表单 `guid`。

1.5.4 获得字段 guid

方法: `getItemGuidByName ($i_formid, $s_title, $s_subtitle = "")`

输入: `$i_formid` 表单 id; `$s_title` 字段名称; `$s_subtitle` 列表子字段名称, 可不传。

输出: 字段 `guid`。

1.5.5 根据名称获得当前字段值

方法: `getValueByName($s_item_name, $s_subitem_name = "", $i_seq = 1)`

输入: `$s_item_name` 字段名称; `$s_subitem_name` 列表子字段名称, 可不传; `$i_seq` 行顺序号, 默认取第一行数据。

输出: 字段值。

1.5.6 获得当前字段值

方法: `getValue($s_item_guid, $i_seq = 1)`

输入: `$s_item_guid` 字段 `guid`; `$i_seq` 行顺序号, 默认取第一行数据。

输出: 字段值。

1.5.7 根据名称获得表单基本数据 (非列表数据) 或者 列表数据

方法: `getRowByName($s_list_name = "", $i_seq = 0)`

输入: `$s_list_name` 列表名称, 可不传, 不传时返回表单基本数据; `$i_seq` 行顺序号, 默认取所有数据。

输出: 基本数据或者列表数据。

1.5.8 获得表单基本数据（非列表数据）或者列表数据

方法：getRow(\$s_list_guid = "", \$i_seq = 0)

输入：\$s_list_guid 列表 guid，可不传，不传时返回表单基本数据；\$i_seq 行顺序号，默认取所有数据。

输出：基本数据或者列表数据。

1.5.9 设置字段数据

方法：setValue(\$s_item_guid, \$value, \$i_seq = 1)

输入：\$s_item_guid 字段 guid；\$value 字段值；\$i_seq 记录行顺序号。

输出：无。

1.5.10 设置字段选项数据

方法：setOptionValues(\$s_item_guid, \$arr_options, \$i_seq = 1)

输入：\$s_item_guid 字段 guid；\$arr_options 选项数组；\$i_seq 记录行顺序号。

输出：无。

1.5.11 清除列表数据

方法：function clearDetailTable(\$s_item_guid)

输入：\$s_item_guid 字段 guid。

输出：无。

1.5.12 获得应用中心数据库连接

方法：function getDb()

输入：无。

输出：应用中心数据库连接。

根据数据库连接操作数据库，常见方法举例如下：

```
$command = $connection->createCommand('SELECT * FROM post');
```

```
$posts = $command->queryAll();
```

```
$post = $command->queryOne();
```

```
$command = $connection->createCommand('UPDATE post SET status=1');
```

```
$command->execute();
```

```
$command = $connection->createCommand('SELECT * FROM post WHERE id=:id');
```

```
$command->bindValue(':id', $_GET['id']);
```

```
$post = $command->query();
```

1.5.13 获得数据源里配置的数据库连接

方法: `function getOtherDB($s_sername, $s_db = "")`

输入: `$s_sername` 数据源名称, 在《系统管理》数据源管理中设置; `$s_db` 数据库名称, 不传时使用数据源里设置的数据库。

输出: 数据库连接。

1.5.14 根据表单名称获得表单 id

方法: `function getFormIdByName($s_app, $s_form)`

输入: `$s_app` 应用名称; `$s_form` 表单名称。

输出: 表单 id。

1.5.15 根据表单 guid 获得表单 id

方法: `function getFormIdByGuid($s_guid)`

输入: `$s_guid` 表单 guid。

输出: 表单 id。

1.5.16 根据字段 GUID 获得字段 ID

方法: `function getItemIdByGuid($s_guid)`

输入: `$s_guid` 字段 `guid`。

输出: 字段 `id`。

1.5.17 根据字段名称获得字段 ID

方法: `function getItemIdByName($i_formid, $s_title, $s_subtitle = "")`

输入: `$i_formid` 表单 `id`; `$s_title` 字段名称; `$s_subtitle` 列表子字段名称, 可不传。

输出: 字段 `id`。

自定义函数

1.6 概述

应用中心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不一的数据处理。此外,应用中心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:

MYFUNC(<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:
MYFUNC(“get_aux_quantity”)。

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:
MYFUNC(“get_aux_quantity”)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\appbuilder\modules\appcenter\plugin\表单 id,
例如: webroot\general\reportshop\utils\plugins\1。

如何查看表单 id? 点击该表单设计界面, 浏览器地址栏上会显示 formId=XXX, XXX
即为该表单 id。

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。

1.7 输入参数与输出结果

应用中心调用自定义函数时, 将给自定义函数传递以下实参(即自定义函数的输入参数):

\$PLUGIN_UTILS: 插件公共对象, 包含了访问公共函数接口;
自定义函数输出结果为计算结果值。

1.8 举例

以倍数换算为例, 举例说明如何使用自定义函数。首先, 定义如下提取触发器:

数据范围： 全部 排重 指定

选择应用： 选择表：

数据来源：

数据填充

选择表单：

| 目标字段 | 填充方式 | 提取表达式 |
|------|----------------------------------|--------------------------------------|
| 基数 | <input type="text" value="无"/> | <input type="button" value="设置表达式"/> |
| 倍数 | <input type="text" value="无"/> | <input type="button" value="设置表达式"/> |
| 结果 | <input type="text" value="填入值"/> | MYFUNC("calculate_it") |

处理引擎：

数据关联

数据过滤：本表单.基数 ISNOTNULL && 本表单.倍数 ISNOTNULL

数据填充

数据排序

其中，MYFUNC("calculate_it")为自定义函数。处理引擎定义为：calculate.func。引擎实现文件 calculate.func.php 包含自定义函数 calculate_it，其脚本如下：

```
/**
 * 计算基数根据倍数翻番后的结果
 * @param $PLUGIN_UTILS 插件公共对象，提供公共方法
 * @return mixed
 */
function calculate_it($PLUGIN_UTILS) {
    $i_basic = $PLUGIN_UTILS->getValueByName("基数");
    $i_power = $PLUGIN_UTILS->getValueByName("倍数");
    $i_result = $i_basic;
    while($i_power > 1){
        $i_result *= $i_basic;
        $i_power--;
    }
    return $i_result;
}
```

运行一结果如下图所示：

单列组容器

基数

55

千分位 : 55

倍数

5

千分位 : 5

结果

503284375

千分位 : 503,284,375

自定义提取或回填引擎

1.9 概述

自定义函数仅在触发器的局部实现了自定义处理逻辑。除此之外，应用中心还支持在整个触发器范围内自定义处理逻辑。要实现整个触发器范围内的自定义处理，需要满足以下条件：

如果是提取触发器，填充方式里除了匹配之外，其他填充方式不得存在。

如果是回填触发器，则无需定义更新方式。

自定义提取或者回填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\appbuilder\modules\appcenter\plugin \模板 id`，例如：`webroot\general\appbuilder\modules\appcenter\plugin \1`。

1.10 输入参数与输出结果

应用中心传递给脚本引擎的参数，即引擎文件输入参数为：

- `$PLUGIN_UTILS` 插件公共对象，，包含了访问公共函数接口。

注意：由于自定义提取或回填脚本引擎是被嵌入到应用中心里一起执行的，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，以区别于应用中心里所使用的变量。

1.11 举例

以初始化数据为例，举例说明如何使用脚本引擎实现自定义提取逻辑，触发器定义如下：

数据范围： 全部 排重 指定

选择应用: 选择表:

数据来源:

数据关联

数据过滤 本表单.寄送方式 ISNULL

数据填充

数据排序

处理引擎:

处理引擎定义为“fill_it”，填充方式空着，即什么都不定义。

引擎脚本文件 fill_it.php 包含以下内容：

```
<?php
/**
 * 北京通达信科科技有限公司
 * Date: 2017/11/21
 * Time: 16:34
 */
$plugin_s_select =
$PLUGIN_UTILS->getItemGuidByName($PLUGIN_UTILS->getFormId(), "寄送
方式");
$PLUGIN_UTILS->setOptionValues($plugin_s_select,
[["code"=>"", "name"=>""], ["code"=>"01", "name"=>"自提
"], ["code"=>"02", "name"=>"邮寄"], ["code"=>"03", "name"=>"送达"]]);
$PLUGIN_UTILS->setValue($plugin_s_select, "01");
$plugin_s_list =
$PLUGIN_UTILS->getItemGuidByName($PLUGIN_UTILS->getFormId(), "列表
");
$plugin_s_guid =
$PLUGIN_UTILS->getItemGuidByName($PLUGIN_UTILS->getFormId(), "列表
", "发票名称");
$plugin_s_guid2 =
$PLUGIN_UTILS->getItemGuidByName($PLUGIN_UTILS->getFormId(), "列表
", "发票号");
$PLUGIN_UTILS->clearDetailTable($plugin_s_list);
$PLUGIN_UTILS->setValue($plugin_s_guid, "充值卡");
```

```
$PLUGIN_UTILS->setValue($plugin_s_guid2, 19918125);
$PLUGIN_UTILS->setValue($plugin_s_guid, "物业费", 2);
$PLUGIN_UTILS->setValue($plugin_s_guid2, 19918126, 2);
$PLUGIN_UTILS->setValue($plugin_s_guid, "采暖费", 3);
$PLUGIN_UTILS->setValue($plugin_s_guid2, 19918127, 3);
?>
```

以回写发票信息为例，举例说明如何使用脚本引擎实现自定义回填逻辑，触发器定义如下：

选择触发器类型 2 定义触发器 3 数据设置

名称： *

步骤流程：

触发方式： 保存触发 删除触发

目标不存在时新建 目标值为零时删除

选择触发器类型 3 数据设置

选择应用： 选择表：

回填目标：

处理引擎：

test112101.列表

发票号

=

test112102.列表

发票号



回写条件和更新方式不定义。处理引擎写上 `back_it`。

脚本引擎文件 `back_it.php` 包含如下内容：

```
<?php
/**
 * 北京通达信科科技有限公司
 * User: lhs
 * Date: 2017/11/24
 * Time: 14:22
 */
//把本表单列表数据回填到目标表单的列表中
//回写目标
$plugin_i_target_formid = $PLUGIN_UTILS->getFormIdByName("特性展示",
"test112102");//目标表单 id
$plugin_i_target_list_id =
$PLUGIN_UTILS->getItemIdByName($plugin_i_target_formid, "列表");//
目标列表 id
$plugin_i_target_fph_id =
$PLUGIN_UTILS->getItemIdByName($plugin_i_target_formid, "列表", "发
票号");//目标字段发票号 id
$plugin_i_target_bz_id =
$PLUGIN_UTILS->getItemIdByName($plugin_i_target_formid, "列表", "说
明");//目标字段说明 id
//本表单
$plugin_i_source_fph_id =
$PLUGIN_UTILS->getItemIdByGuid("{C55A3264-04D4-297B-7685-26AFE7FE
10C2}");//本表单字段发票号 id
$plugin_i_source_fpmc_id =
$PLUGIN_UTILS->getItemIdByGuid("{B8240BDB-5A0B-21F3-D676-2E9A07DC
36CF}");//本表单字段发票名称 id
////本表单列表数据
$plugin_arr_row = $PLUGIN_UTILS->getRowByName("列表");
if(!empty($plugin_arr_row)){
    foreach($plugin_arr_row as $plugin_row){
        $sql = "select id from
app_data_{\$plugin_i_target_formid}_list_{\$plugin_i_target_list_id}
where
col_{$plugin_i_target_fph_id}='{\$plugin_row[\"col_{$plugin_i_source_f
ph_id}]}'";
```

```
$plugin_target_row =
$PLUGIN_UTILS->getDb()->createCommand($sql)->queryOne();
    if(!empty($plugin_target_row)){
        $plugin_s_value = "发票号:
{$plugin_row["col_$plugin_i_source_fph_id"]}, 发票项目:
{$plugin_row["col_$plugin_i_source_fpmc_id"]}";
        $sql = "update
app_data_{$plugin_i_target_formid}_list_{$plugin_i_target_list_id}
set col_$plugin_i_target_bz_id='{$plugin_s_value}' where
id={$plugin_target_row[id]}";
        $PLUGIN_UTILS->getDb()->createCommand($sql)->execute();
    }
}
}
?>
```

客户服务

1.12 版本升级

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

1.13 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/service/>

1.14 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。

1.15 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。

1.16 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

官方网址：<http://www.tongda2000.com/>