贵阳高登世德金融科技有限公司

Golden Stand Financial Technology

现金流设计工具使用说明

-V1.0

目录

[1.设计背景 3](#_Toc476269852)

[1.1公司介绍 3](#_Toc476269853)

[1.2设计背景 3](#_Toc476269854)

[2.基本原理 4](#_Toc476269855)

[2.1算法原理 4](#_Toc476269856)

[2.2拆分过程 5](#_Toc476269857)

[2.3归集过程 6](#_Toc476269858)

[3.软件安装 10](#_Toc476269859)

[3.1环境要求 10](#_Toc476269860)

[3.2安装部署 10](#_Toc476269861)

[3.2.1软件包目录结构介绍 10](#_Toc476269862)

[3.2.2安装部署现金流工具 11](#_Toc476269863)

[4.使用说明 12](#_Toc476269864)

[4.1软件功能模块介绍 12](#_Toc476269865)

[4.1.1Task导航模块 12](#_Toc476269866)

[4.1.2Task编辑模块 12](#_Toc476269867)

[4.1.3CaculationView模块 14](#_Toc476269868)

[4.1.4结果展示模块 16](#_Toc476269869)

[4.2具体使用样例 17](#_Toc476269870)

# 1.设计背景

### 1.1公司介绍

SFM-资产证券化系统软件由高登世德公司设计，高登世德是一个基于金融大数据与互联网技术，提供海量资产资产证券化及大资产负债表，流动性管理解决方案的专业金融科技与财务顾问公司。公司注册资金1000万，在2015年10月10号新闻联播中作为贵阳大数据基地引入公司典型，重点报道了公司主营业务。团队核心成员全部在澳大利亚四大银行有十年以上的金融大数据工作经验。是国内唯一拥有自主知识产权的，覆盖资产证券化产品全生命周期的解决方案提供商。

### 1.2设计背景

现金流指现金流入与现金流出的差值，即：

Cashflow（）=cash in（） – cash out（）

现金流分拆指的是将还款计划拆分成按期还本金和按期还利息的过程。在很多公司或项目融资过程中往往不止一个途径和方式，需要多种方式和多方的融资途径才可能完成一个项目的庞大资金需要，整个融资过程算下来可能少则几十笔数额大小不等，还款期数不同且还款日期不同的资金，多则上百笔，也就意味着融资者需要专门去统计和盘算这些资金，确保及时还款，以免未及时还款而对公司信誉及项目进度造成损失。

本软件的设计就是针对这些不同的而又冗杂的还款计划的，实现每笔款项清楚明了。自动将还款计划拆分成按期还本金和按期还利息的详细表，做到笔笔清楚，简化用户在拆分现金流过程中的工作量以及减小出错率，对于用户来说是十分有意义且有针对性的。

下面举例说明。比如王先生向A、B机构各贷款15000，且贷款时间都为1年，利息都为5%。A机构每月本息一起还，B机构是先息后本。那么A的还款计划应为每月还款：（15000/12）+ 1500 X 5% = 2000 （元），且连续还款12个月。B的还款计划则是先每月还款：1500 X 5% = 750 （元），且连续还12个月，最后一年期满还本金15000。

这是一个小例子，仅仅针对这个例子可以看出对于还款，涉及到的不仅仅是款项大小不同、还款期数会有差异，更加令人头疼的问题是不同款项还款方式的不同，尤其对于一些大的款项来说，设计的还款方式可能不是单一的，那么此时涉及的算法也就是相对复杂的。对于几笔这样的款项去一一计算是可以接受的，而对于大量这样的数据就令人十分头疼了，对于这些数据的处理也十分复杂且容易出错， 那么本软件就是基于强大算法下的工具，基于用户的统计数据及需要的算法，为用户解决这一难题。

# 2.基本原理

## 2.1算法原理

说到算法原理，无非就是基于不同还款方式下的本金和利息的计算。

最常用的有等额本息或者等额本金等额本金和等额本息就比较复杂，等额本息法的特点是：每月的还款额相同，在月供中“本金与利息”的分配比例中，前半段时期所还的利息比例大、本金比例小，还款期限过半后逐步转为本金比例大、利息比例小。所支出的总利息比等额本金法多，而且贷款期限越长，利息相差越大。等额本金法是每月的还款额不同，它是将贷款额按还款的总月数均分（等额本金），再加上上期剩余本金的月利息，形成一个月还款额，所以等额本金法第一个月的还款额最多 ，尔后逐月减少，越还越少。所支出的总利息比等额本息法少。

1、等额本息还款：

；

P－贷款金额 n－贷款期限月数 I－贷款月利率

1）首月

首月正常应还本金 ＝ 月还款额－贷款金额×月利率

首月正常应还利息 ＝ 贷款金额×月利率×首月计息天数30天

首月计息天数＝贷款发放日至次月计息日的天数

2）末月（贷款到期日当月）

末月正常应还本金 ＝ 贷款余额－逾期本金

末月正常应还利息 ＝（贷款余额－逾期本金）×月利率×末月计息天数30天

末月计息天数 ＝ 上月计息日次日至贷款到期日天数

3）其他月（非首、末月）

当月正常应还利息 ＝（贷款余额－逾期本金）× 月利率

当月正常应还本金 ＝ 月还款额－当月正常应还利息

2、等额本金还款：

月还本金＝贷款金额÷贷款期限月数

1）首月

首月正常应还本金 ＝ 月还本金

首月正常应还利息 ＝ 贷款金额×月利率×首月计息天数30天

2）末月（贷款到期日当月）

末月正常应还本金 ＝ 贷款余额－逾期本金

末月正常应还利息 ＝ （贷款余额－逾期本金）×月利率×末月计息天数30天

末月计息天数 ＝ 上月计息日次日至贷款到期日天数

3）其他月（非首、末月）

当月正常应还利息 ＝（贷款余额－逾期本金）× 月利率

当月正常应还本金 ＝ 月还本金

基于以上算法，可能会有细微差别，但基本原理预算法基本一致。其差别可能在于有些是按季度还款，有些是按年还款，有点是按季度结息，有的按年度结息，差别仅此而已，原理基本一致。

## 2.2拆分过程

所谓的拆分即将一条数据拆分为多条数据。获取原始数据，所谓最原始数据即贷款金额、贷款期数、贷款利率、还款方式，而最后企业所求的一般是一个时期应还贷款的本金总额，而我们将其分开，将这一个时期分成多个时间段，且将总额进一步拆分，拆分成应还本金和应还利息。对于这一步有什么意义呢？在此作用下，可以看到每一天的贷款款项的还款情况，用户可以及时知道一天应还款项笔数及总额，清楚知道现金流的流动情况，对于企业来说意义重大。由于其代码过于复杂且过多，此处仅以流程图来说明其原理。，

应还总额

第二期应还总额

第n期应还总额

第一期应还总额

最后一期应还总额

本金

利息

本金

利息

本金

利息

本金

利息

## 2.3归集过程

拆分之后，需要对信息进行归集，对于多笔贷款而言，需要将同一天还款的信息归集在一起，方便用户知道每一天的现金流输出状态。仅以一段简单的代码就可说明其原理。“//“以后为原理解释部分。

public void onClick\_btnImutation(Event event)  
{  
  // 表单验证  
  if (!this.formValidator.validate()) {  
     return;  
  }  
  // 得到表单数据  
  PortfolioCashflowChartDto formDto = (PortfolioCashflowChartDto)this.formBinder.getModel();  
  // 归集日期列表  
  List<Date> imutationDates = new ArrayList();  
  // 第一个归集日，由用户录入  
  imutationDates.add(formDto.getFirstimutationdate());  
  // 最后一个归集日，根据池中所有资产的最后一个PayDate确定  
  Date last Date = ((CashflowDto)this.allcashflow.get(this.allcashflow.size() - 1)).getPaydate();  
  // 借用日历类来处理日期。  
  Calendar cld = Calendar.getInstance();  
  // 把日历的时间设置成第一个归集日  
  cld.setTime(formDto.getFirstimutationdate());  
  //用户设定的“是否月末规则”，默认为否。  
  boolean isMonthendrule = false;  
  //Calendar.DATE或者是5 ,得到 当前月的最大天数  
  int firstdaysofmonthend = cld.getActualMaximum(5);  
  int calendarField;  
  // 如果以月为周期类型  
  if (formDto.getCycleunit().equals("M")) {  
    int calendarField = 2;  
     // 只有当第一个归集日就在月末，且月末法则为“是”的时候，在真正设置月末法则。  
    if ((firstdaysofmonthend == cld.get(5)) && (formDto.getMonthendrule().equals("Y")))  
    {  
      isMonthendrule = true;  
   }  
  }  
  // 如果以天为周期类型  
  else {  
    calendarField = 5;  
  }  
  Date imutationDate;  
  // 循环加入归集日。  
  do { cld.add(calendarField, formDto.getCyclenumber().intValue());  
    if (isMonthendrule) {  
      int daysofmonthend = cld.getActualMaximum(5);  
      cld.set(5, daysofmonthend);  
    }  
    imutationDate = cld.getTime();  
    imutationDates.add(imutationDate);

  } while (!imutationDate.after(last Date));

  List<CashflowDto> cashflowDtoList = new ArrayList();  
  Date lastImutationDate = null;  
  // 根据归集日来设置本金和利息  
  for (Date imutationDate : imutationDates) {  
     //每一个归集日的利息和本金集合  
    List<CashflowDto> interestDtos = new ArrayList();  
    List<CashflowDto> returnDtos = new ArrayList();  
    for (CashflowDto cashflowDto : this.allcashflow) {        
       //lastImutationDate == null 表示第一个归集日，或者PayDay大于上一个归集日  
      if ((lastImutationDate == null) || (lastImutationDate.compareTo(cashflowDto.getPaydate()) < 0))  
      {  
          // 当PayDay的日期小于等于当前归集日  
        if (imutationDate.compareTo(cashflowDto.getPaydate()) >= 0)  
        {  
            // 属于当前归集日的利息和本金。  
          if (cashflowDto.getTypename().equals("利息")) {  
            interestDtos.add(cashflowDto);  
          }  
          else  
            returnDtos.add(cashflowDto);  
          }  
      }  
    }  
     // 利息汇总  
    CashflowDto interestSumCashflowDto = new CashflowDto();  
    BigDecimal interestSum = BigDecimal.ZERO;  
    for (CashflowDto cashflowDto : interestDtos) {  
      interestSum = interestSum.add(cashflowDto.getPayamounttotal());  
    }  
    if (interestSum.compareTo(BigDecimal.ZERO) > 0) {  
      interestSumCashflowDto.setPaydate(imutationDate);  
      interestSumCashflowDto.setPayamounttotal(interestSum);  
      interestSumCashflowDto.setTypename("利息");  
      cashflowDtoList.add(interestSumCashflowDto);  
    }  
    
    //本金汇总  
    CashflowDto returnSumCashflowDto = new CashflowDto();  
    BigDecimal returnSum = BigDecimal.ZERO;  
    for (CashflowDto cashflowDto : returnDtos) {  
      returnSum = returnSum.add(cashflowDto.getPayamounttotal());  
    }  
    if (returnSum.compareTo(BigDecimal.ZERO) > 0) {  
      returnSumCashflowDto.setPaydate(imutationDate);  
      returnSumCashflowDto.setPayamounttotal(returnSum);  
      returnSumCashflowDto.setTypename("本金");  
      cashflowDtoList.add(returnSumCashflowDto);  
    }  
    
    lastImutationDate = imutationDate;  
  }  
  
  renderCashflowChart(cashflowDtoList);

}

# 3.软件安装

## 3.1环境要求

操作系统： windows server 2012 r2。

Web服务器： IIS8.5。

数据库：MS Sqlerver 2014以上版本。

开发工具：Visual Studio 2013以上版本。

dotNetFramework :4.0以上版本。

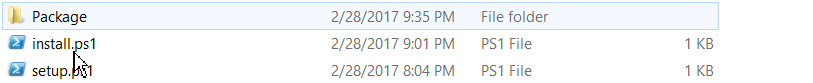
依赖的系统服务：ASP.NET 状态服务。

PowerShellVersion:4.0。

## 3.2安装部署

### 3.2.1软件包目录结构介绍

从FTP服务器获取现金流工具软件包，解压到相应的目录，可以看到解压后的目录结构如下。



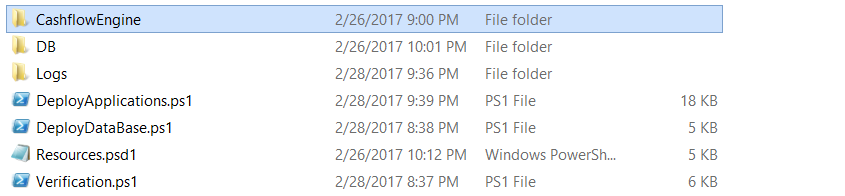
其中：

Packge： 软件包；

install.ps1：调用环境校验，部署应用程序，部署数据库工程的脚本；

setup.ps1：安装部署的现金流工具的脚本，欲部署安装现金流工具，只需双击运行此脚本即可。

打开Package目录，可以看到如下结构：



在此目录下，

CashflowEngine：现金流工具Application发布包，

DB：现金流工具数据库发布包，

Logs：安装部署现金流工具时日志文件存放区，

DeployApplication.ps1：IIS中部署现金流工具Application的PowerShell脚本，

DeployDataBase.ps1：部署数据库项目的脚本，

Resources.psd1:安装部署时，日志消息定义，

Verification.ps1：环境校验脚本。

### 3.2.2安装部署现金流工具

1、修改网站配置信息

打开Packge目录，找到DeployApplication.ps1,修改：

siteName：网站名字

sitePhysicalPath：网站物理路径

siteAppPoolName：应用程序池

sitePort：网站绑定的端口号

deploymentUserName：部署网站的用户名

deploymentUserPassword：用户密码

managedRunTimeVersion：.NET CTL版本

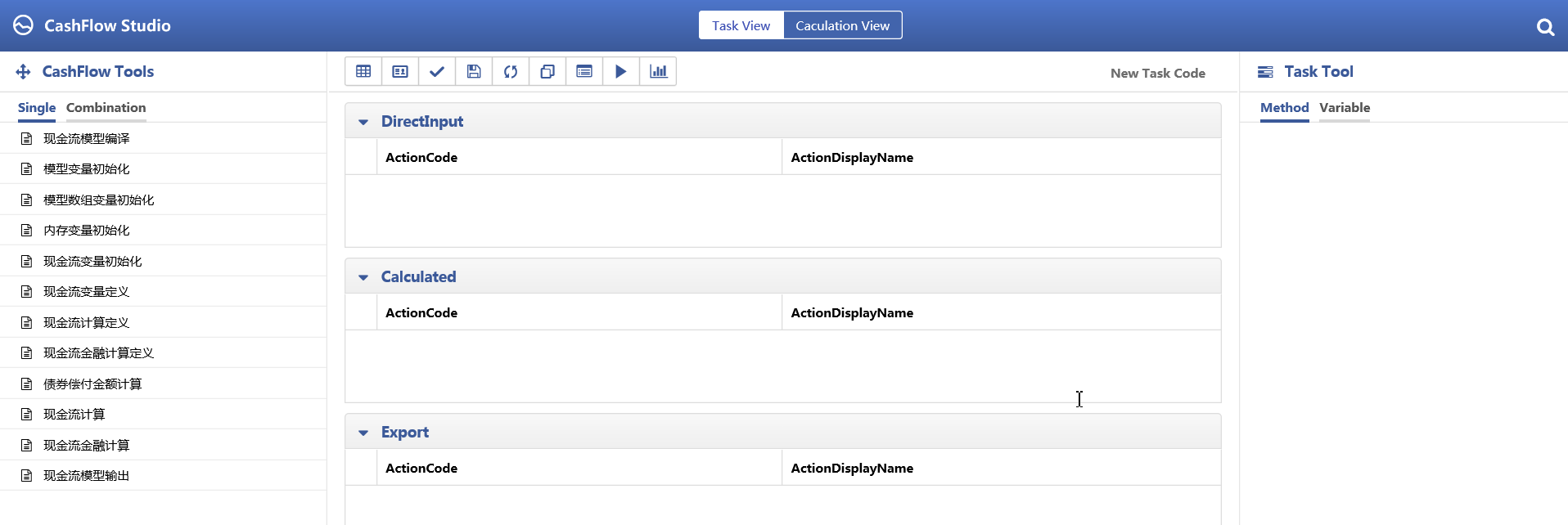
hostHeader：host Name

ipAddress:要绑定的IP地址

ppName：应用程序的名称

2、运行脚本进行自动安装部署

打开软件发布包，找到setup.ps1脚本文件，以管理员身份运行setup.ps1脚本，现金流工具自动安装部署，其中包括环境校验、数据库工程部署、应用程序部署。其中环境校验包括FramWork校验，PowerShell校验，IIS校验，SQL Server校验，Silverlight校验，脚本运行完成后，自动弹出现金流工具首页，安装部署日志文件存放在Logs文件夹下面，包括环境校验日志文件，数据库部署日志文件，应用程序部署日志文件。



# 4.使用说明

## 4.1软件功能模块介绍

此软件主要分Task导航模块，Task View模块，Caculation View模块，Results 显示四个模块。

### 4.1.1Task导航模块

用于加载并显示已存在的所有现金流计算的Task。此模块通常情况处于隐藏状态，当点击页面右上角标题栏中的收索按钮时方能弹出，当鼠标移除导航栏区域或点击导航栏右上角的关闭按钮时，导航栏将被隐藏



在Task导航栏中有Task搜索栏，Task筛选框以及Task列表显示栏三部分，在Task搜索栏中输入域名，点击搜索按钮，所有的Task将显示在Task列表中



在Task列表显示栏中，选中并双击一个Task选项，所有与此Task相关的Action，变量，方法，计算结果等都将被加载。在Task列表显示栏中，用户还能新增，删除Task。在Task筛选框中输入欲筛选的Task，系统自动筛选出相应的Task选项，并显示在列表显示栏中。

### 4.1.2Task编辑模块



Task编辑模块从左往右分别为现金流工具模块，Action编辑模块，Task工具模块，

其中：

现金流工具模块为设计Action提供模板，用户只需选中一个模板往Action编辑区拖，在编辑区中进行定制工作，此工具栏提供了两套模板，一套为单个的Action，另一套为组合模板（包含多个Action），即Combination面板中的选项。

Action编辑模块：在此模块中有工具栏，Action编辑区。工具栏从左往右依次为详细编辑，简略编辑，应用，保存，刷新，变量输入，预览，运行，结果显示，现金流Task名称编辑处。



详细编辑按钮：点击此按钮将弹出编辑Action的编辑框，供用户编辑Action

简略编辑按钮：此按钮功能与详细编辑按钮功能一样，只是相对来说简单了些

应用按钮：当编辑完一个Action后，唯有点击此按钮方能在保存之前点刷新按钮不被撤销。

保存按钮：保存编辑好了的Action。

刷新按钮：当编辑完一个Action后，且没点击应用和保存按钮时，点击刷新按钮可撤销编辑

变量输入：点击此按钮，弹出变量输入界面，供用户输入设计工作流时所需的变量。

预览：点击此按钮可以预览该现金流Task的内容

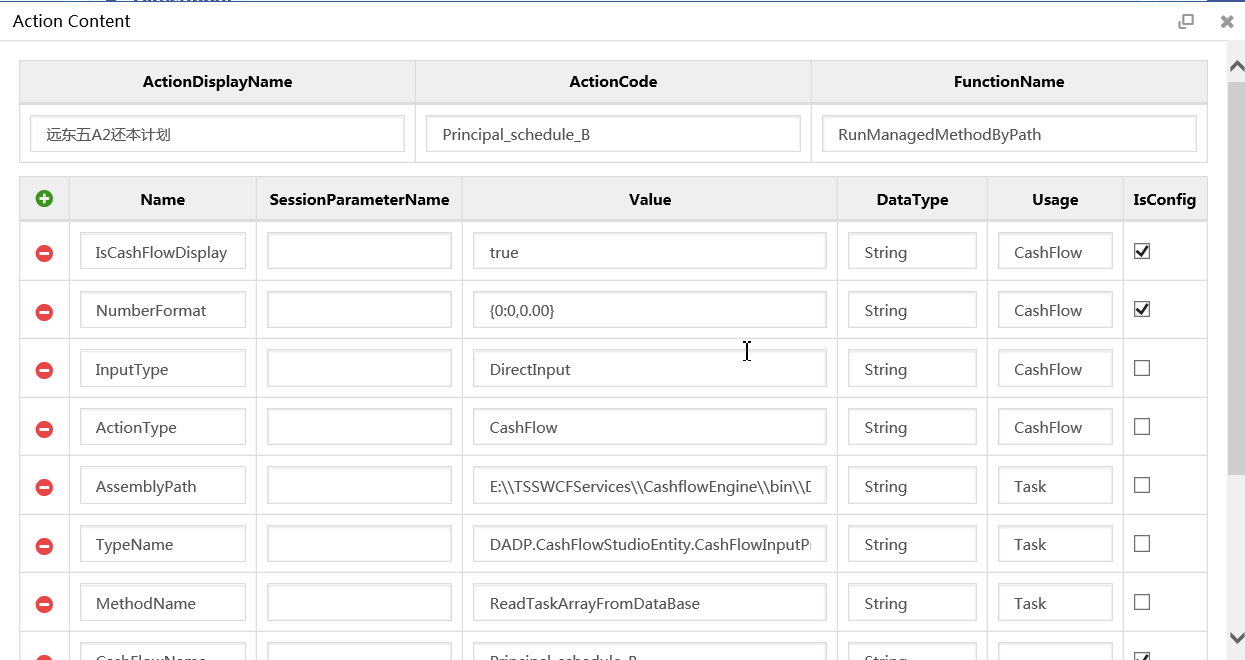
运行：点击此按钮运行现金流Task

结果显示：点击此按钮，弹出结果显示界面。

现金流Task名称编辑处：编辑现金流Task的名称

Action编辑区：Action编辑区中显示与该现金流Task相关的所有Action，选中此区域中的Action，双击则弹出详细Action编辑框，还可以对该区域的Action进行复制粘贴以及删除，选中Action，右击弹出操作框，以进行复制粘贴以及删除操作。此Action区的选项将会在计算方法定义模块中以MetaData方式被引用

详细Action编辑框：供用户编辑Action，界面如下



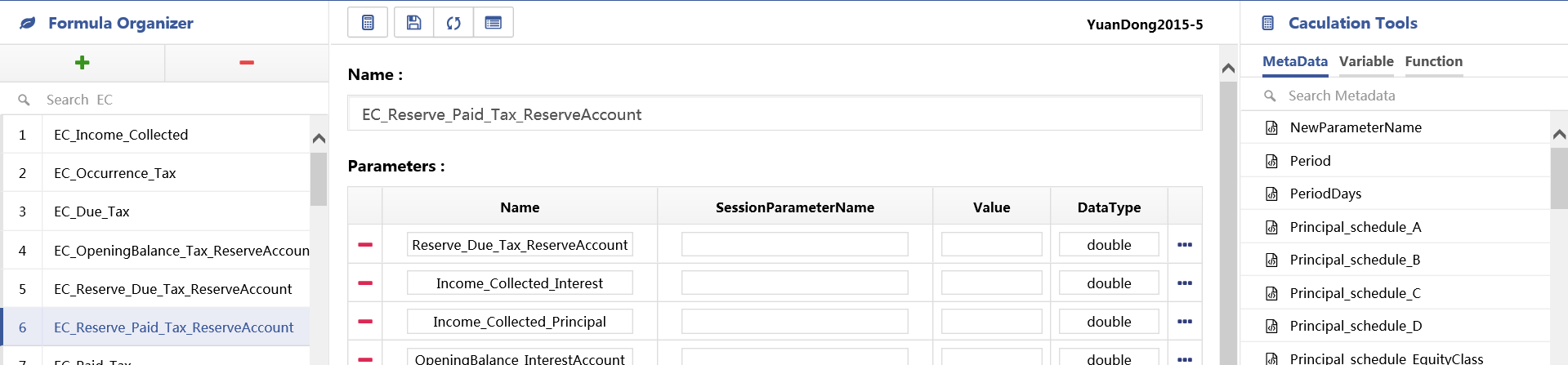
其中需要注意引用dll路径 是否正确，数据库连接是否正确。如果方法名字是计算方法定义模块中定义的话，双击它会跳转到计算方法定义模块进行编辑。点击此页面右上角的最大化按钮时，此页面可以推动，再点击此按钮，页面将恢复到原来的位置，点击关闭按钮时，页面被关闭

Task工具模块：Task工具模块中有两面板，一个叫方法面板，此面板的内容是由计算方法模块定义，

选中其中任一方法名并双击，将会跳转到该方法的编辑页，此面板的选项主要是用来编辑Action使用，可以选中列表中的一个方法拖动到详细Action编辑框中的方法名称处进行引用。另一个叫做变量面板，此面板中的选项由用户从变量输入界面输入，用于定义计算方法。

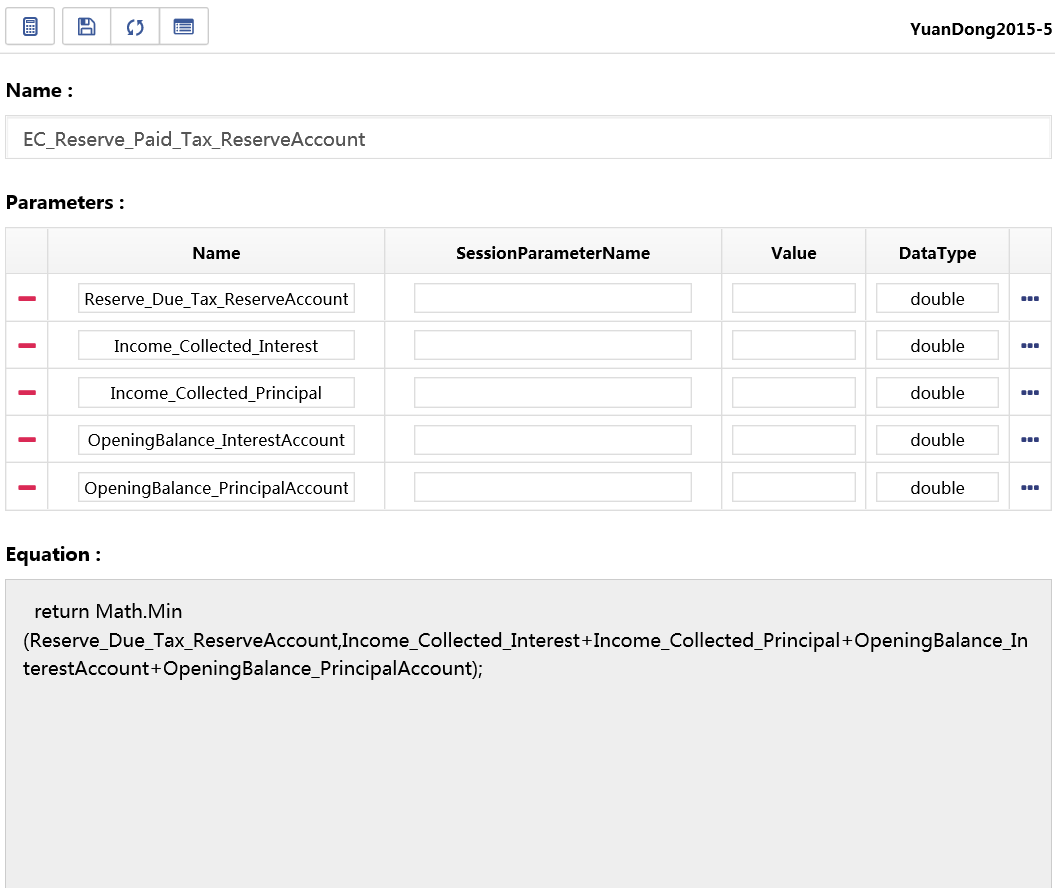
### 4.1.3CaculationView模块

点击标题栏的Caculation Viwe进入计算方法定义模块，此模块分为Formula Organizer，EC Edit View，和Caculation Tools三部分



其中Formula Organizer为EC Method名称列表显示栏，也是EC Method导航栏，用户可以在此栏中选中选项，然后在其右侧EC Edit View中进行编辑，编辑后的EC Method会在Task View 界面的 Task Tool中显示，供用户在编辑Action时使用，当用户在Task Tool栏下面的Method列表中选中EC Method 名称并双击，页面将跳转到Caculation View界面，且相应的EC Method处于编辑状态。用户可以在Formula Organizer面板下点击新增按钮新增一个EC Method。选中EC Method名称，点击删除按钮，被选中的EC Method将被删除。用户可以在Search EC筛选框中对EC Method进行筛选。

EC Edit View :即是EC编辑面板



从上往下分别为工具栏，EC Method编辑区

工具栏从左往右依次为Verify校验，保存，刷新，XML预览，EC名称编辑处。

Verify：对EC进行校验

保存：用户编辑完EC后点此按钮进行保存。

刷新：在用户编辑完还没保存时，点击此按钮可以撤销编辑。

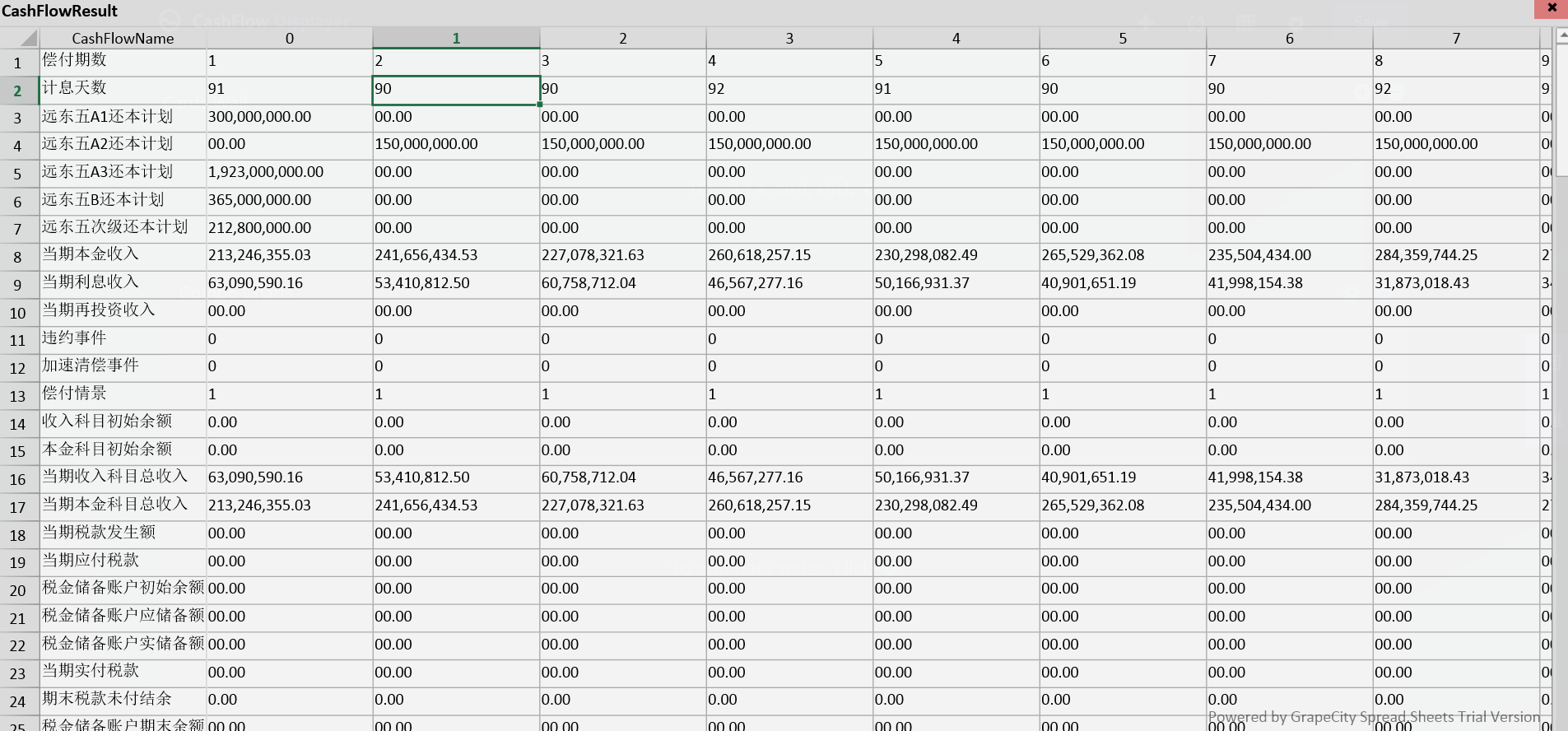
XML预览：当用户对整个EC编辑完后，点击此按钮进行预览，也可以直接在XML预览界面对EC进行编辑。

EC Method编辑区：当用户从从EC Method导航栏中选中EC Method名称时，此区域将显示与此EC Method相关的信息供用户编辑，用户也可以在Task View的Task tool中的Method列表下选中EC Method 名称，双击进入此编辑区，也可以在Action 编辑区中的Method Name后面的Value双击EC Method名称跳转到此编辑区。此区域包括EC Method名称编辑，所需参数编辑，以及计算公式的组织。编辑EC Method所需的参数可以从右侧Caculation Tools面板中的MetaData和Variable列表下拖入。在组织计算公式的时候，用户可以从参数编辑栏中拖拽变量到此位置，也可以在鼠标停留处按“@”符号进行选择，用到的运算符或者函数模板可以从右侧Functon列表中拖拽。

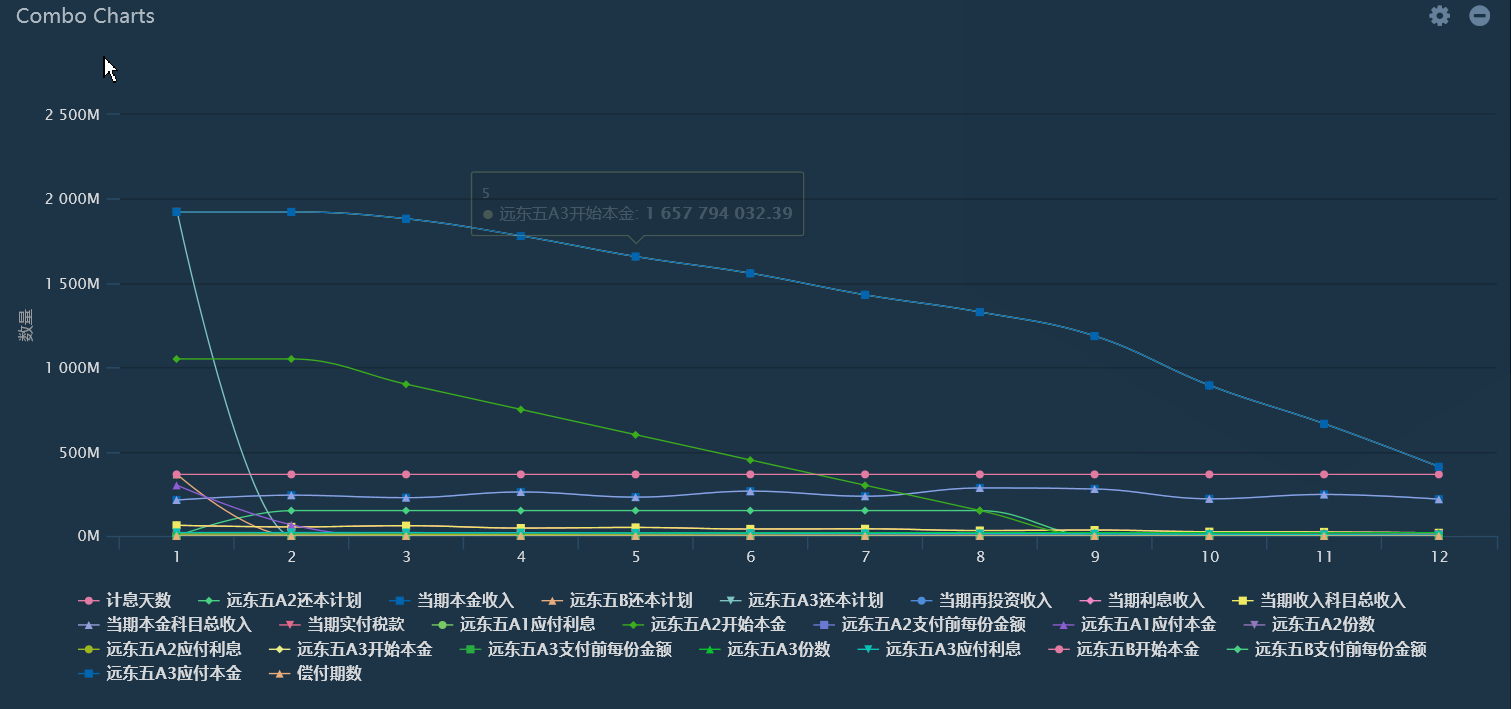
Caculation Tools：包含MetaData,Variable，Function三部分，主要为组织EC Method的计算公式提供变量和计算符号或者方法。其中需要强调的时MetaData为前面Task View中带有返回值的Action，此返回值在进行EC Method时被当做变量使用

### 4.1.4结果展示模块

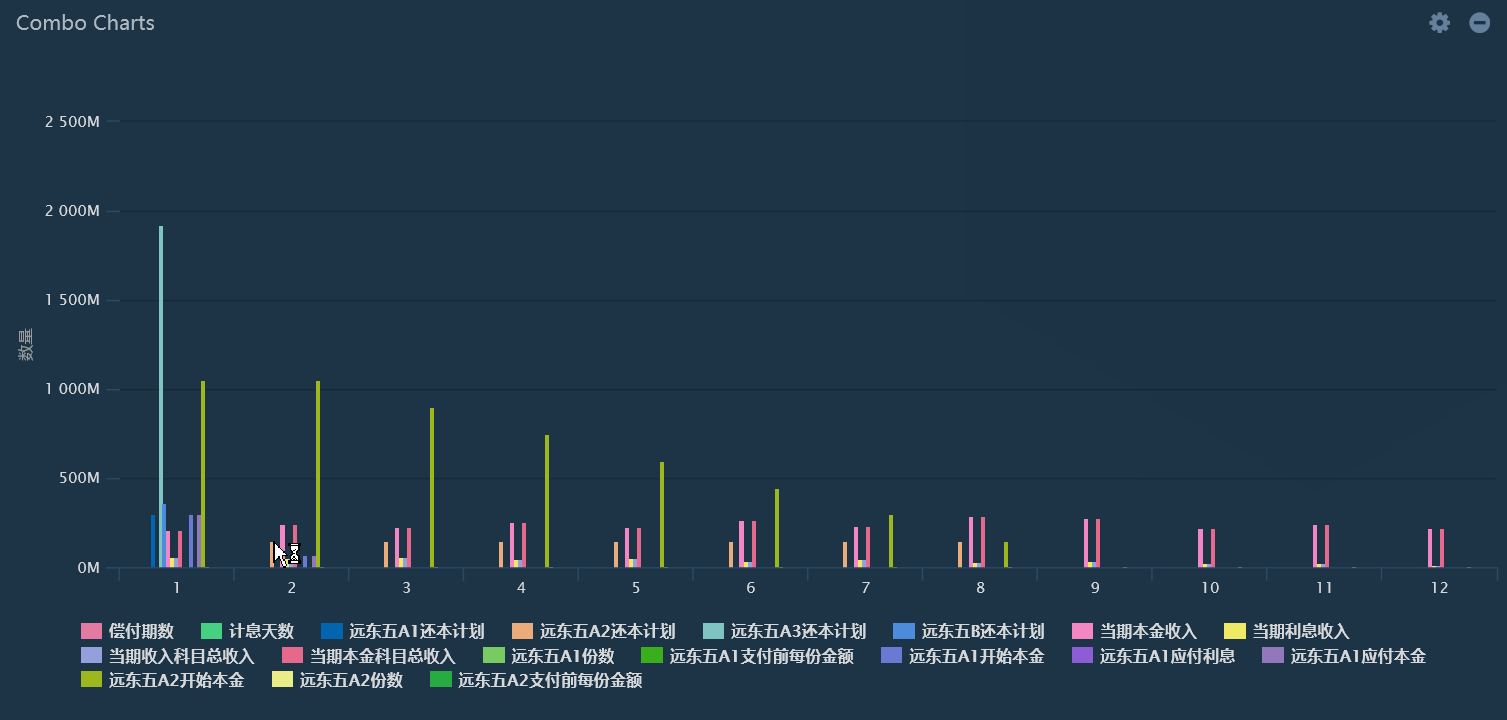
结果展示方式有卡片的形式，表格，曲线图，柱状图，板块图等多种方式，用户可以自定义结果展示的方式



（表格形式）



（曲线图）



（柱状图）

## 4.2具体使用样例

参见使用演示视频