驰骋工作流引擎JFlow &amp; CCFlow 操作手册 V1.0

概要说明

不支持低版本的IE浏览器,请使用火狐、谷歌浏览器查看.

**如果目前您已经看到此内容，就说明在线浏览支持您的浏览器。**

**有疑问请联系我们:**  0531-82374939

**关注微信号：**

图片地址：https://static.oschina.net/uploads/img/201507/06144438\_FC3c.png

关于JFlow与CCFlow

**关于JFlow 与 CCFlow**

驰骋工作流引擎包含了工作流引擎与表单引擎，表单引擎请参考《驰骋表单引擎说明书.doc》。

驰骋工作流引擎它有两个版本.net版本的ccflow与java版本的jflow，通常我们把JFlow 与 ccflow 通称为cc。

在本章中介绍了cc的产品信息、如何学好cc、cc可以满足什么样的应用环境、cc的开发模式、cc的演示环境、以及如何向cc提交代码。

因为维护开发文档的工作量巨大，但是两个版本的操作手册差异较小，所以，我们把操作手册合并一个。对于差异大的部分，我们分别描述，没有差异的部分我们合并一个章节讲述。

驰骋工作流引擎分为Java版本的JFlow与.net版本的ccflow，驰骋公司先研发了ccflow，然后在ccflow的基础上使用工具翻译出来了java版本的JFlow，两个版本我们统称为驰骋工作流程，所以两者有如下相同之处。

1, ccflow与jflow的功能完全相同，新增加的功能保持同步增加发布。

2, ccflow与jflow 表结构完全相同，设置与配置界面，操作方式相同，接口api名称相同。

3， ccflow与jflow 使用同一个操作手册。

4， ccflow与jflow 的流程模版，表单模版通用。

CCFLow研发于2003年，到一直持续到现在，功能强大丰富，图形化的配置，功能性配置较高，在中国国情下成长起来的优秀的工作流引擎。这两个版本的工作流程引擎都是100%开源的，在国内都有一定的市场地位，是国内著名的老牌工作流引擎。

**如果您想了解cc工作流引擎的功能列表，请参考cc的官方网站。**

ccflow官方网站(http://ccflow.org/)

jflow官方网站(http://jflow.cn/)

**如果您想学习驰骋表单引擎，请打开cc表单引擎操作手册**

驰骋表单引擎在线操作手册(http://jflow.mydoc.io)

致驰骋工作流的使用者一封信

**各位驰骋工作流的爱好者、使用者、入门者：**

欢迎您了解、认识使用驰骋工作流，在您决心要使用驰骋工作流之前您需要注意如下事项。

1，  并非所有的爱好者都能成功的学习好、使用上驰骋工作流。就算你开始抱有激情万丈的热情，也有粉身碎骨的可能，掌握好学习驰骋工作流的方法很重要。

2，  文档与视频是您学习好驰骋工作流的法宝，驰骋工作流的团队已经把所有您可能遇到的问题都写入了文档，所以您要付出充分的时间去学习与掌握这些文档，甚至背会这些驰骋工作流总结出来的规则、技巧、技术。根据我们的实践证明，那些使用上驰骋工作流的爱好者，在没有驰骋工作流技术支持的帮助下没有一个不是看文档成功的。

3，  没有学会应用之前，不要开始入门研究驰骋工作流的代码，那样您会陷入牛角尖里，并且您看代码对您并没有多少的帮助，除非您已经熟练的掌握好了驰骋工作流各个功能，了解了驰骋工作流的设计原理与思想。

4，  驰骋工作流的设计思想与整体架构与国内外的很多同类的软件不同，不要拿别人的思想去套用驰骋工作流，在您接触到其他的同类的软件后，在研究驰骋工作流往往是一些思路思想与驰骋工作流对应不上。

5，  驰骋工作流博大精深、完全掌握、游刃有余的使用驰骋工作流非一月之功。你要有足够的耐心，反复的揣摩、试验、碰撞，你才能领会到驰骋工作流设计的精华，你才会爱上驰骋工作流，离不开驰骋工作流。

6，  一般来说驰骋工作流的功能与驰骋工作流的说明书是同步的，有时会出现发了功能但是没有修改说明书的情况，请耐心等待。

7，  驰骋工作流已经向社会承诺，核心代码100%的开源，并且永远开源，并不排除增值软件收费的可能，如果你相信驰骋工作流,那就放心研究驰骋工作流。

8，  我们的技术支持是针对vip用户的，所谓的vip用户是付费向驰骋工作流申请专业服务的单位，驰骋工作流的发展基础是那些尊重知识产权、尊重作者劳动的vip用户的支持，如果您具有一定的条件请支持驰骋工作流的发展。

9，  为什么法律知识都明明白白的写成册还要聘请律师？为什么医学知识都是敞开的为什么还要请医生看病？这是因为学业有专攻，那些书本描述的再详细也替代不了律师与医生的经验知识。同样也是如此，我们已经尽我们的可能把此操作手册写的详细也并不能保证您一定能够使用好驰骋工作流，但是我们所做的所有努力都是帮助驰骋工作流的爱好者们使用好驰骋工作流。

致!!!

**驰骋工作流开发团队**

产品信息-JFlow

**产品名称:** 驰骋.net工作流程引擎.

**英文名称:** chicheng workflow engine 简称: JFlow

**开发历史:** 2003-2015年. 性质: 完全开源软件

**许可协议:** GPL URL: http://www.gnu.org/licenses/gpl.html

**官方网站:** http://jflow.cn 下载地址： http://jflow.cn/download.aspx

**支持数据库:** MSSQLSERVER, Oracle, Infomax, MySQL.

**客户端:** FireFox 3.0以上. IE6+,或者使用IE内核的浏览器.

**引擎性能：**请参考<< JFlow效率测试报告.docx >>

**组成部分:** 流程可视化图形设计器/表单可视化设计器/流程前台运行程序(BS结构)/流程服务(CS)/单元测试

**基本功能:** 图形化流程设计/智能表单web定义/免程序开发/流程轨迹图/单据自定义打印/邮件短信工作到达通知/自动任务分配/支持sdk模式开发/简洁集成/消息侦听/丰富事件接口/报表定义/工作量分析/绩效考核/手机访问/报表定义设计.

懂管理就会开发基本流程,会SQL就能达到中级开发水平解决复杂的逻辑问题与其它系统集成,有编程基础就可以使用JFlow的sdk做高级的开发.

产品信息-CCFlow

产品信息-CCFlow

**产品名称:** 驰骋.net工作流程引擎.

**英文名称:** chicheng workflow engine 简称: CCFlow版本: 6.0.0

**开发历史:** 2003-2015年. 性质: 完全开源软件

**许可协议:** GPL URL: http://www.gnu.org/licenses/gpl.html

官方网站: http://ccflow.org 源代码: http://ccflow.org/Download.aspx请用svn下载(地址可能有变化以官方网站为准).

**运行环境:** bs 结构. windows XP, windows server. win7 IIS6.0以上. .net4.0. Office2007 (如果需要单据打印.)

**开发语言:** VS2010. .net4.0 c#.net.支持数据库: MSSQLSERVER, Oracle, Infomax, MySQL.

**客户端:** FireFox 3.0以上. IE6+,或者使用IE内核的浏览器.

**引擎性能：**请参考<< ccflow效率测试报告.docx >>

**组成部分:** 流程可视化图形设计器/表单可视化设计器/流程前台运行程序(BS结构)/流程服务(CS)/单元测试

**基本功能:** 图形化流程设计/智能表单web定义/免程序开发/流程轨迹图/单据自定义打印/邮件短信工作到达通知/自动任务分配/支持sdk模式开发/简洁集成/消息侦听/丰富事件接口/报表定义/工作量分析/绩效考核/手机访问/报表定义设计.

**CCFlow:** 懂管理就会开发基本流程,会SQL就能达到中级开发水平解决复杂的逻辑问题与其它系统集成,有编程基础就可以使用ccflow的sdk做高级的开发。

对使用者要求

**对使用者要求**

驰骋工作流把使用者分为三种类型，流程基础设计人员、流程高级设计人员、流程二次开发人员，下面把不同层次的人员的知识要求与问题解决能力介绍如下。

**流程基础设计人员：**

知识要求：懂流程管理，流程业务知识，对信息化系统有一定的了解与掌握，能够做一些基础的维护。

流程处理能力：能够为单位建立基础的办公流程、业务流程。

说明：驰骋工作流基本都是web界面化的操作，不需要编译、设计所见既所得。

**流程高级设计人员：**

知识要求：满足[流程技术设计人员]条件，会基础的SQL语法，能够编写函数，存储过程。公司、单位系统管理员一般都具备这个水平。

流程处理能力：能够把驰骋工作流与其它系统集成起来工作，能够处理复杂的业务逻辑开发，能够有效的把驰骋工作流与其它的业务系统集成在一起。

**流程二次开发人员：**

知识要求：会写Javascript 脚本验证，具有一定的编程知识与能力，有1-2年的业务系统开发经验。

流程处理能力：能够利用API可以与其它系统集成起来工作，能够处理复杂的业务逻辑开发，能够有效的把驰骋工作流与其它的业务系统集成在一起。

安装集成升级

下载与安装

**JFlow的下载与安装**

http://git.oschina.net/chichengsoft/ccflow/wikis/pages(http://http://git.oschina.net/chichengsoft/ccflow/wikis/pages)

**CCFlow 的下载与安装**

[http://git.oschina.net/chichengsoft/ccflow/wikis/home](http://http://git.oschina.net/chichengsoft/ccflow/wikis/home)

组织机构的集成

**组织机构的模式：**

根据不同的情况我们把组织结构分为一人一部门多岗位的模式，我们称呼为OneOne模式，我们把一人多部门多岗位的模式成为OneMore模式。对于这两种模式在全局配置文件里OSMode 属性里。

JFlow中的配置，文件路径：D:JFlow.gitjflow-websrcmainresourcesconfweb.properties

图片地址：https://static.oschina.net/uploads/img/201605/30142936\_ECx6.png

CCFlow中配置, 文件路径：D:ccflow.gitCCFlowweb.config

图片地址：https://static.oschina.net/uploads/img/201605/30142944\_r7qz.png

**关于两种模式下所需要的表：**

OneOne模式的组织结构表（一人一部门多岗位）

图片地址：https://static.oschina.net/uploads/img/201605/30143514\_oQ86.png

**您可以通过如下SQL来查询出来:**

SELECT No,Name,Pass,FK\_Dept,SID FROM PORT\_EMP  
SELECT No,Name,ParentNo FROM PORT\_Dept  
SELECT No,Name,FK\_StationType FROM PORT\_Station  
SELECT No,Name FROM PORT\_StationType  
SELECT FK\_Emp,FK\_Station FROM PORT\_EmpStation

**OneMore模式的组织结构表（一人多部门多岗位）**

图片地址：https://static.oschina.net/uploads/img/201605/30143617\_AADE.png

您可以通过如下SQL来查询出来:

SELECT No,Name,Pass,FK\_Dept,SID FROM PORT\_EMP  
SELECT No,Name,ParentNo FROM PORT\_Dept  
SELECT No,Name,FK\_StationType FROM PORT\_Station  
SELECT No,Name FROM PORT\_StationType  
SELECT MyPK,FK\_Emp,FK\_Dept FROM PORT\_DeptEmp  
SELECT MyPK,FK\_Emp,FK\_Dept,FK\_Station FROM PORT\_DeptEmpStation

**组织结构表的概念：**

用于控制用户的权限的表都称为组织结构表，比如：操作员、部门、岗位。与ccbpm的集成也就是组织结构表的集成，以上表格里的表都是组织结构表。

组织结构的数据来源：

根据用户的不同运行环境，我们把组织机构数据来源分为0=通过数据库映射，2通过webservices获得数据。3通过AD,获得组织结构数据，请参考上图的OSDBSrc设置，目前对于非vip用户，请使用0模式。

**组织结构集成的核心思想：**

把ccbpm集成到您的应用系统里，首先要解决的组织结构的集成，然后根据自己的项目确定集成模式。不管什么方式的集成，首先要把组织结构集成在一起。

把ccbpm中的组织表删除了，换成数据结构一样的视图，数据是从被集成的系统数据库映射回来的。这样ccbpm在工作时，仍然不会影响。以一个sql来说明问题。

DROP TABLE Port\_Emp /\* 删除本机的人员表 \*/  
 GO  
 CREATE VIEW Port\_Emp /\* 创建人员视图表 \*/  
 AS  
 SELECT BH AS No, MC as Name, bumenbianhao as FK\_Dept, Password as Pass FROM ERP.dbo.Yonghu  
 GO

对于OneMore处理模式ccbpm如何识别当前登录部门属于那个部门？

我们知道一人多部门的信息存储到Port*DeptEmp表里，这里描述了一个人对应多个部门，但是用户进入系统后如何知道他是以那个人员的身份登录的？ccbpm读取的是Port*Emp表的FK\_Dept字段作为当前部门。如果用户要切换登录部门的时候，就要先修改这个字段值作为当前登录部门。

**对于历史版本的支持：**

CCFlow系统会自动升级，升级代码请参考BP.WF.Glo.UpdateVer()方法。

JFlow 请手动按照上述方式修改与配置。

**重要的事情说三遍：**

Ccbpm的组织机构集成，就是删除ccbpm的组织机构表，建立同数据结构一样的视图!!!

Ccbpm的组织机构集成，就是删除ccbpm的组织机构表，建立同数据结构一样的视图!!!

Ccbpm的组织机构集成，就是删除ccbpm的组织机构表，建立同数据结构一样的视图!!!

CCFlow代码模式集成

**第1步：** ccflow的项目引入到您的解决方案里。

|  |  |
| --- | --- |
| **类库** | **说明** |
| \Components\BP.En30 | BP框架的底层类库 |
| \Components\BP.Web.Controls30 | BP框架的控件类库 |
| \Components\BP.WF | 流程引擎层 |

**注意** 不建议引用dll，因为如果ccflow有更新，ccflow不会发布最新的dll而是发布最新的源代码，所以如果涉及到更新，首先更新带有源代码的类库。

**第2步：**开始执行编译，如果编译不通过，ccflow还会依赖其他的第三方开放的类库，它们都位于。DLL 目录下面.

**第3步：**把ccflow的必须的前台代码copy到您的前台解决方案里。

有重要的如下几个目录

|  |  |
| --- | --- |
| **目录名称** | **说明** |
| \CCFlow\DataUser | Ccflow的目录文件 |
| \CCFlow\WF | 前台流程应用文件 |
| \CCFlow\Web.config | 应用配置文件，如果您的应用程序也把一些配置放入了appSetting 节点里，请把他们整合在一起. |

**第4步：**设置数据库连接，并执行ccflow的安装。

打开web.config 配置文件，修改AppCenterDSN数据库连接，最好的方案是把您的应用程序数据库与ccflow的数据库放在一个数据库里。

执行ccflow的数据库的安装，直接运行：http://yourServerIP/WF/Admin/DBInstall.aspx。

按照提示步骤执行安装，如果您是正式的运行环境，就不要安装ccflow提供的demo了。

调用Webservices模式集成

**这种模式适合cs调用执行模式，或者wpf执行模式**。

这种方式就是把ccflow的api都发布到webservices上，在客户端进行调用，ccflow的手机版本程序，就是使用的该方式完成的。

**第1步：**下载ccflow到d:ccflow下面后就不要变动代码。

**第2步：**建立网站应用程序主目录设置为D:ccflowtrunkCCFlow。

**第3步：**调用ccflow封装的相关api，ccflow内置的调用地址: /WF/WorkOpt/ CCFlowAPI.asmx。

CCFlow对相对路径的支持

**如何让ccflow支持相对路径？为什么要支持相对路径？**

把ccflow集成到您的应用系统，请参考以前的方案。如果需要把ccflow的代码与目录移动到您所指定的目录里，比如 /Plug-in/CCFlow, 您就需要按照如下步骤执行。把WF , DataUser，ZhiDu（如果需要把流程与制度集成在一起）,AppDemo, AppDemoLigerUI , OrganizationalStructure.asmx 移动到 Plug-inCCFlow\*.\* 下面。

增加或者修改web.config 里面的信息.

**不建议使用该配置，最近几个月的修改一些功能不支持自定义相对路径了.**

<!-- 配置信息路径-->  
  
<add key="DataDirPath" value="Plug-in\CCFlow\WF" />  
  
<!--用户数据路径 -->  
  
<add key="DataUserDirPath" value="Plug-in\e\" />  
  
<!-- 应用程序目录 -->  
  
<add key="CCFlowAppPath" value="/Plug-in/CCFlow/" />

驰骋工作流引擎的升级

1.JFlow&CCFlow把更新分成三类, 应用程序代码更新、数据表结构更新、数据更新。

2.JFlow&CCFlow在您登录流程设计器时自动判断当前的版本与数据库版本是否一致，不一致它就会自动启用更新程序，把代码与数据库更新到一致的版本。

3.我们会保持2年更新，就是说您两年以前使用JFlow&CCFlow, 一直没有更新，如果您下载了JFlow&CCFlow新代码，JFlow&CCFlow会把2年内的补丁包都自动打上。

4.JFlow&CCFlow有数据表自动修复功能，在您使用JFlow&CCFlow是有时间第一次看到一个错误，第二次就没有了，这是因为您没有用升级的模式登录JFlow&CCFlow, 在遇到新程序与旧的数据表结构不一致时，JFlow&CCFlow会自动修复上，第二次运行就没有错误了。

5.不要尝试每次都去更新，如果现阶段驰骋工作流引擎已经满足了您的需要，那就不要去更新。

6.JFlow升级代码的位置.

图片地址：https://static.oschina.net/uploads/img/201507/06145515\_L2zi.png

7.在ccflow中的代码自动升级位置如下图，有兴趣的朋友可以去看看。

图片地址：https://static.oschina.net/uploads/img/201507/06145531\_3P14.png

工作流的组成部分

总体框架组成

在研究jflow&ccflow的sdk开发模式之前我们首先研究jflow&ccflow的组成部分。

我们用一个图来标识组成部分：

图片地址：https://static.oschina.net/uploads/img/201506/09172848\_CbET.png

**jflow&ccflow的后台组成部分：**流程设计器+表单设计器+流程服务。

**流程设计器是定义流程：**定义流程图、流程方向、流程条件、流程属性节点属性运动规则，定义表单与节点的关系，导入导出流程模版，对运行的流程干预（比如：流程删除、强制移交、强制终止、运行流程查询）。

表单设计器：定义表单的元素以及表单元素之间的业务规则，这部分由表单设计器手册单独描述。

流程服务：它是一个cs程序，用于启动定期的流程、执行自动发送的节点、执行发送邮件短信、执行把数据写入即时通讯系统。

**jflow&ccflow的前台组成部分：**流程解析执行器+表单解析执行器

**流程解析执行器：**读取流程的定义信息、按照定义的规则执行流程的运转。

**表单解析执行器：**读取表单的定义信息，展现表单元素，解释并执行表单的业务逻辑，这部分由《驰骋表单设计器操作手册详细说明》

流程四大菜单 API

登录

**登录**

如果采用集成的模式，登录的界面是集成人员编写的，验证通过后，执行jflow&ccflow的API让jflow&ccflow写入登录信息。

**登录实现的代码：**

//取到用户编号.  
 string userNo = this.TB\_No.Text;  
 // 调用登录接口,写入登录信息。  
 BP.WF.Dev2Interface.Port\_Login(userNo);  
  
这里输入代码

用户调用登录的api后, 就向jflow&ccflow写入了注册信息，在BP框架里（ccflow是采用驰骋公司BP框架做的开发），就可以使用BP. Web.WebUser.\* 访问登录者的用户帐号，用户名，部门信息。

BP.Web.WebUser.No 当前登录的帐号,

BP.Web.WebUser.Name 当前登录的名称,

BP.Web.WebUser.FK\_Dept 当前登录的部门,

BP.Web.WebUser.FK\_DeptName当前登录的部门名称。

**常见问题：**如果要把cc集成到自己的开发框架里，需要把先执行自己框架的登录，然后在执行bp的框架登录。

发起

**发起**

当前登录人员可以发起的流程，每个人进入可以发起的流程集合是不同的，系统为您提供了一个AIP,让您获取发起流程列表，然后您自己div一个发起界面。

<%  
 //获取可以发起的流程集合。  
 System.Data.DataTable dt=   
 BP.WF.Dev2Interface.DB\_GenerCanStartFlowsOfDataTable(BP.Web.WebUser.No);  
 // 输出集合.  
 %>  
 <table border="1" width="100%" >  
 <caption>发起(列出当前人员可以发起的流程,<a href="Do.aspx?DoType=ViewStartSrc" target=\_blank >查看源代码</a>)</caption>  
 <tr>  
 <th>序</th>  
 <th>类别</th>  
 <th>流程</th>  
 <th>操作</th>  
 </tr>  
 <%  
   
 int idx = 0;  
 foreach (System.Data.DataRow dr in dt.Rows)  
 {  
 idx++;  
  
 int flowNum = int.Parse(dr["No"].ToString());  
 %>  
 <tr>  
 <td class="Idx"><%= idx%></td>  
 <td><%=dr["FK\_FlowSortText"] %></td>  
  
  
  
 <td>  
 <a href="/WF/Rpt/Search.aspx?RptNo=ND<%=flowNum %>MyRpt&FK\_Flow=<%=dr["No"] %>" >查询</a>  
 </td>  
  
 </tr>  
 <% } %>   
 </table>

一个流程可以被那些人员发起，是有该流程的开始节点访问权限来决定的，点击连接就可以进入工作处理器功能界面。

**生成发起界面的API:**

我们通过API，就可以了解到了原来ccflow的发起功能如此简单，开发人员自己就可以调用接口来构造，事例文件位置：

vip用户：D:ccflowvalue-addedCCFlowWFAppSimpleStart.cs

普通用户：D:ccflowtruckCCFlowWFAppSimpleStart.cs

点击流程名称就连接到工作处理器上，连接工作处理器就是MyFlow.aspx?FK\_Flow=流程编号.

流程发起界面!

输入图片说明(https://static.oschina.net/uploads/img/201506/09180927\_QuoV.png)

待办

**待办**

待办：顾名思义就是等待当前操作人员需要处理的工作，点击待办后就可以进入工作处理器进行处理工作，待办与发起都是连接的工作处理器。当前登录人员可以处理的工作，点击一个工作的标题，就可以进入工作处理器的功能界面。

工作待办：

图片地址：https://static.oschina.net/uploads/img/201506/09181224\_E0Qq.png

获取待办的API

<%  
 //获取待办。  
 System.Data.DataTable dt = BP.WF.Dev2Interface.DB\_GenerEmpWorksOfDataTable();  
 // 输出结果  
 %>   
 <table style="width:90%; border:1px" ><caption>待办 (<%=BP.WF.Dev2Interface.Todolist\_EmpWorks %>) </caption>  
 <tr>  
 <th>序</th>  
 <th>是否读取?</th>  
 <th>标题</th>  
 <th>流程</th>  
 <th>发起时间</th>  
 <th>发起人</th>  
 <th>停留节点</th>  
 <th>(发送/抄送)人</th>  
 <th>流程备注</th>  
 <th>类型</th>  
 </tr>  
 <%  
 // 生成timke 方式浏览器打开旧的界面，方式界面缓存.   
 string t = DateTime.Now.ToString("MMddhhmmss");  
 int idx = 0;  
 foreach (System.Data.DataRow dr in dt.Rows)  
 {  
 //流程引擎的参数。  
 string paras = dr["AtPara"] as string;  
 if (paras == null)  
 paras = "";  
 //这个工作是否读取了？根据状态开发人员可以个性化的显示工作读取未读取效果.  
 int isRead = int.Parse(dr["IsRead"].ToString());  
  
 idx++;  
 %>  
 <tr>  
 <td class="Idx"><%= idx%></td>  
  
 <td><%=dr["isRead"]%></td>  
   
 <% if (isRead == 0)  
 {%>  
  
 <% }  
 else  
 { %>  
  
 <%} %>  
  
 <td><%=dr["FlowName"]%></td>  
 <td><%=dr["RDT"]%></td>  
 <td><%=dr["StarterName"]%></td>  
 <td><%=dr["NodeName"]%></td>  
 <td><%=dr["Sender"]%></td>  
 <td><%=dr["FlowNote"]%></td>  
  
 <% if (paras.Contains("IsCC")) { %>  
   
 <td>抄送</td>  
  
 <% } else { %>  
   
 <td>发送</td>  
   
 <%} %>  
  
  
 </tr>  
 <% } %>   
 </table>

什么是工作处理器，在下一章节里有详尽的介绍。

在途

在途也叫未完成，它是一个人参与的流程，但是该流程还没有走完，在途也叫未完成工作，点在途进入工作查看器界面。

在途的界面效果：

图片地址：https://static.oschina.net/uploads/img/201506/09181815\_Rsq2.png

在途的API:

<script type="text/javascript">  
 // 撤销。  
 function UnSend(appPath, fk\_flow, workid) {  
 if (window.confirm('您确定要撤销本次发送吗？') == false)  
 return;  
 var url = appPath + 'WF/Do.aspx?DoType=UnSend&WorkID=' + workid + '&FK\_Flow=' + fk\_flow;  
 window.location.href = url;  
 return;  
 }  
 function Press(appPath, fk\_flow, workid) {  
 var url = appPath + 'WF/WorkOpt/Press.aspx?WorkID=' + workid + '&FK\_Flow=' + fk\_flow;  
 var v = window.showModalDialog(url, 'sd', 'dialogHeight: 220px; dialogWidth: 430px;center: yes; help: no');  
 }  
 function CopyAndStart(appPath, fk\_flow, CopyFormNode, CopyFormWorkID) {  
 var url = appPath + 'WF/MyFlow.aspx?CopyFormWorkID=' + CopyFormWorkID + '&CopyFormNode=' + CopyFormNode + '&FK\_Flow=' + fk\_flow;  
 var v = window.open(url, 'sd', 'dialogHeight: 220px; dialogWidth: 430px;center: yes; help: no');  
 }  
  
 function WinOpenCC(ccid, fk\_flow, fk\_node, workid, fid, sta) {  
 var url = '';  
 if (sta == '0') {  
  
 }  
 else {  
  
 }  
 //window.parent.f\_addTab("cc" + fk\_flow + workid, "抄送" + fk\_flow + workid, url);  
 var newWindow = window.open(url, 'z');  
 newWindow.focus();  
 }  
   
 </script>  
</asp:Content>  
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">  
   
 <%  
 //获取在途工作。  
 System.Data.DataTable dt = null;  
 if (BP.Sys.SystemConfig.AppSettings["IsAddCC"] == "1")  
 dt = BP.WF.Dev2Interface.DB\_GenerRuning();  
 else  
 dt = BP.WF.Dev2Interface.DB\_GenerRuningAndCC();  
   
 string path = this.Request.ApplicationPath;  
 // 输出结果  
 %>  
 <table style="width:100%; border:1px">  
 <caption>在途 (<%=BP.WF.Dev2Interface.Todolist\_Runing %>)</caption>  
 <tr>  
 <th>序</th>  
 <th>标题</th>  
 <th>流程</th>  
 <th>发起时间</th>  
 <th>发起人</th>  
 <th>停留节点</th>  
 <th>当前处理人</th>  
 <th>操作</th>  
 </tr>  
 <%  
 int idx = 0;  
 foreach (System.Data.DataRow dr in dt.Rows)  
 {  
 string workid = dr["WorkID"].ToString();  
 string fk\_flow = dr["FK\_Flow"].ToString();  
 string nodeID = dr["FK\_Node"].ToString();  
 idx++;  
 %>  
 <tr>  
 <td class="Idx"><%= idx%></td>  
  
 <td>  
 <% if ( 1==1 || dr["Type"] + "" == "RUNNING")  
 { %>  
 <a href="/WF/WFRpt.aspx?FK\_Flow=<%= dr["FK\_Flow"] %>&WorkID=<%= dr["WorkID"] %>"  
 target="\_blank">  
 <%= dr["Title"] %>  
 </a>  
 <% }  
 else  
 { %>  
  
 <%=dr["Title"] %></a>  
 <% } %>  
 </td>  
  
 <td nowarp=true><%= dr["FlowName"] %></td>  
 <td nowarp=true><%= dr["RDT"] %></td>  
 <td nowarp=true><%= dr["StarterName"] %></td>  
 <td nowarp=true><%= dr["NodeName"] %></td>  
 <td nowarp=true><%= dr["TodoEmps"] %></td>  
 <% if (1 == 1 || dr["Type"] + "" == "RUNNING")  
 { %>  
 <td nowarp=true>  
 <% if (dr["FID"].ToString() != "0")  
 { %>  
 [<a href="/WF/DeleteWorkFlow.aspx?FK\_Flow=<%= fk\_flow %>&WorkID=<%= workid %>" target=\_blank >删除</a>]  
 <% }  
 else  
 { %>  
  
 [<a href="/WF/WorkOpt/UnSend.aspx?FK\_Flow=<%= fk\_flow %>&WorkID=<%= workid %>" target=\_blank >撤销发送</a>]  
  
 <%} %>  
 -[<a href="javascript:Press('<%= path %>','<%= fk\_flow %>','<%= workid %>')" >催办</a>]  
  
 </td>  
 <% } else {%>  
 <td> </td>  
 <% } %>  
 </tr>  
 <% } %>  
 </table>

工作处理器的3中工作模式

模式1-标准工作处理器

**流程解析执行器=菜单+工作执行器+工作查看器**

**菜单：** 发起、待办、未完成、已完成、查询，这些菜单列表都可以通过ccbpm的api获取一个数据集合datatable，然后呈现出来，这些功能呈现出来后，就要链接到工作处理器上或者工作查看器上。

**工作处理器 = 工作控制按钮工具栏+表单**

图片地址：https://static.oschina.net/uploads/img/201506/09182754\_UVlp.png

**预览：**

图片地址：https://static.oschina.net/uploads/img/201506/09182829\_4DOD.png

模式2-树形工作处理器

一个节点上需要绑定多个表单时，这些表单以树形的方式展现出来，我们称树形工作处理器，如下图所示。

图片地址：https://static.oschina.net/uploads/img/201506/10090600\_D9p8.png

在头部是流程控制器部分，左边是表单树右边则是表单工作区域。在头部的功能按钮是与标准模式都相同。

模式3-Tab标签页工作处理器

如果一个节点需要挂接多个表单工作时，系统就按照如下模式进行展现。

标签页的工作模式简洁方便，适应于表单不多的节点，如果该节点挂的表单很多，就造成表单展现拥挤，那就转化成树形表单来使用。

图片地址：https://static.oschina.net/uploads/img/201506/10090833\_NT7w.png

不管树形表单还是标签页表单，在表单之间切换时候，系统就会自动保存当前表单数据，如果表单的数据有变更，ccform就会出现\*号，来标识该表单数据有变化。

工作处理器部件

退回

说明：退回顾名思义就是，把当前的工作退回给指定的节点的处理人，让其重新办理，如下是执行退回窗口。

图片地址：https://static.oschina.net/uploads/img/201506/10103347\_1aI4.png

执行退回后，被退回人，就可以看到退回信息，如下图所示：

图片地址：https://static.oschina.net/uploads/img/201506/10103359\_SG5s.png

一个步骤上可能接受了多次退回，系统会把每次退回的信息写入日志，并把每次退回意见显示出来如上图。点工作日志，就可以看到当时退回的时候所记录的数据。

退回日志(https://static.oschina.net/uploads/img/201506/10103437\_7a5s.png)

退回时的工作信息!

在节点属性，功能按钮，接退回规则按钮，详细的描述了该功能如何使用。

接受人

定义：接收人是下一步工作节点的处理人。

事例：比如A发到B, B的节点的访问规则是由上一步的发送人来选择的。在A节点发送前需要选择一个或者多个工作的处理人，就叫接收人。

图片地址：https://static.oschina.net/uploads/img/201506/12115920\_cXim.png

在节点属性功能按钮，接收人按钮，详细的描述了该功能如何使用。

移交

**定义：**当事人处理的一件工作，因为不是自己处理，或者自己处理不了，或者自己离职需要交接给其他人处理，这就是工作移交。

图片地址：https://static.oschina.net/uploads/img/201506/10151033\_2g6U.png

**执行移交**点上图的工作移交后，就转入工作移交窗口，执行工作移交：

图片地址：https://static.oschina.net/uploads/img/201506/10151043\_FpyX.png

执行移交后，当前人员的待办消失，被移交人出现待办，被移交人进入系统后就出现如下提示信息。

**操作者进入：**让zhangyifan进入系统，就出现待办。

图片地址：https://static.oschina.net/uploads/img/201506/10151139\_Ut6Z.png

在节点属性-按钮权限-移交规则章节，有详细的描述。

加签

**定义：**甲处理当前工作时，需要请求乙的意见，这个行为叫加签。甲可以让乙在加签后，直接发送到下一步，也可以让乙填写意见后发给自己，如下图右边的加签按钮。

图片地址：https://static.oschina.net/uploads/img/201506/10152538\_txJi.png

**界面呈现：**点击加签按钮，进入二级界面如下图：

加签的选项有两个，如下图：

图片地址：https://static.oschina.net/uploads/img/201506/10152603\_lcxW.png

点击提交提示执行加签信息：

用zhoupeng帐号登录，他就可以找到一个待办工作，该工作被打开后，就提示如下信息：

图片地址：https://static.oschina.net/uploads/img/201506/10152626\_Im8s.png

点击【回复加签意见】连接，就到了填写加签意见窗口，如下图：

图片地址：https://static.oschina.net/uploads/img/201506/10152643\_mIzV.png

填写意见后，点提交就可以完成加签的工做，这个时候系统就会提示执行结果。如果是加签并原路返回，就会发送给加签人，否则，就会发送到下一步骤。

**在轨迹图的体现：**对于以上的操作，cc的工作日志详细的记录了该步骤，如下图：

图片地址：https://static.oschina.net/uploads/img/201506/10152701\_aLB4.png

抄送审核

A节点发送到B节点，要抄送给张三，张三看到信息后，他可以填写自己的意见，这就是抄送审核。

**抄送规则：**

**不能抄送：**不执行抄送，抄送按钮在表单页面中不显示。

**手工抄送：**用户自己选择抄送人，抄送按钮会在表单页面中显示，点击后，进入选择抄送人页面。

**自动抄送：**当前节点执行发送后，ccflow自动执行抄送。这个功能需要提前在节点属性中设置抄送人。

**手工与自动：**以上两种情况的结合，但是没有考虑重复抄送的问题。比如：自动抄送中，设置了抄送人A，在手工抄送的时候，又选择了A，那么就会发送给A两次。如果手工选择了B，那么就会发送给A和B两个人。

**按表单字段计算：**设置自动抄送的效果一样，只不过不需要在节点表单中设计抄送人，而是在表单中根据字段获取抄送人。

**在发送前打开抄送窗口：**实际上相当于一个发送前事件，这个功能，适用于强制性选择抄送人的情况。

删除

**定义**：流程走到特定的步骤后，当前工作人员有删除流程不让其向下运行的抄作。删除流程与结束流程不同，删除流程一般是彻底的删除该记录，包括运行轨迹，而结束流程就是运行到这里，流程就不向下运动。

如下图：

图片地址：https://static.oschina.net/uploads/img/201506/10153951\_n6Bk.png

点击确定后，提示执行的结果，流程的删除分好多种，每种删除的应用场景不一样，请参考节点属性中的按钮权限控制章节，有详细的介绍。

跳转

**定义：**在特定的步骤，当前的工作处理人员需要跳转到指定的节点上去，这种应用场景叫跳转。跳转是在不规范的情况下，或者特定的应用场景下的需求，当前节点与被跳转到的节点之间可以无连接线，可以跳转的节点选择范围是有当前节点的跳转规则来决定的。点击如下图的跳转按钮，就转到跳转界面上来。

图片地址：https://static.oschina.net/uploads/img/201506/10154041\_Lgrk.png

**跳转的执行：**选择一个要跳转的节点，系统就执行跳转操作。

图片地址：https://static.oschina.net/uploads/img/201506/10154049\_wUdK.png

**执行跳转的提示信息：**执行的跳转信息如下：

图片地址：https://static.oschina.net/uploads/img/201506/10154057\_KhK4.png

子流程

**应用背景：**在一个节点上，启动N个子流程后，当前操作人员想知道每个子流程的运行状态。

**测试案例：**立项流程的第2个节点，启动子流程节点。

图片地址：https://static.oschina.net/uploads/img/201506/10154553\_38Pa.png

在启动几个子流程后，点击子流程按钮，就出现如下窗口。

图片地址：https://static.oschina.net/uploads/img/201506/10154600\_OWuq.png

**打开连接：**点击每个连接，就可以看到子流程运行的详细信息，就进入工作查看器。

子线程

**应用背景：**在合流节点上，当前人员要想看到每个子线程运行的结果，并且要对每个子线程进行删除、退回操作。

图片地址：https://static.oschina.net/uploads/img/201506/12120303\_PX4V.png

**说明：**子线程是否可以被删除，是否可以被退回，都是有节点属性来控制的。

**子线程窗口：**点击子线程窗口，如下图：

图片地址：https://static.oschina.net/uploads/img/201506/12120327\_CiLi.png

点打开连接，就可以看到该子线程的表单信息，如下图：

图片地址：https://static.oschina.net/uploads/img/201506/12120340\_YgLD.png

**其他：**请参考节点属性-按钮权限-子线程按钮章节。

打印单据

在特定的节点需要把当前表单的信息按照一定的格式打印出来，系统提供了多种打印方式，请参考节点属性-按钮权限-打印规则。

批处理

参考节点属性--基本属性---节点工作批处理

流转自定义

应用背景：一个流程实例启动起来以后，他的以后节点的运行步骤，与轨迹是可以在该节点上定义的，通过此功能就可以执行流程的流转自定义。

图片地址：https://static.oschina.net/uploads/img/201506/23182313\_6Zsa.png

请参考：节点属性\功能按钮\流转自定义章节。

挂起

应用场景：一件工作，接受到以后，此工作因为其他原因，需要搁置一段时间处理，在指定的时间在出现待办，或者通过消息提醒当事人处理，我们把这种操作行为叫挂起。

图片地址：https://static.oschina.net/uploads/img/201506/23182352\_ryNw.png

公文

请参考 节点属性--公文按钮

工作查看器

**工作查看器：**用来查看一个流程实例运行的情况，可以查看已经完成与未完成的流程。

工作查看器是显示详尽的展示类该流程实例的信息，对于不同的人员可以对该流程事例所可以执行的操作，下图描述了工作查看器与工作处理器、与菜单之间的关系。

图片地址：https://static.oschina.net/uploads/img/201506/23183715\_gpfn.png

工作查看器有流程轨迹图、流程日志、流程操作三部分组成。

**流程轨迹图：**

图片地址：https://static.oschina.net/uploads/img/201506/23183734\_W3tf.png

**流程日志：**

图片地址：https://static.oschina.net/uploads/img/201506/23183747\_VfIp.png

**流程操作：**

图片地址：https://static.oschina.net/uploads/img/201506/23183803\_FkF5.png

对于一个流程实例的操作是有一定的权限的。

对于已经完成的工作可以执行的操作有：逻辑删除、彻底删除、回滚到指定的节点。

对于未完成的操作有：退回、撤销、逻辑删除、彻底删除。

默认admin 具有逻辑删除、彻底删除的权限。

流程设计器

绘制流程的基础操作

**新建空白流程**

图片地址：https://static.oschina.net/uploads/img/201506/14142103\_95Nf.png

在流程树上点击右键，选择新建，进入新建流程界面。

**填写流程流程参数**

图片地址：https://static.oschina.net/uploads/img/201506/14142123\_RypZ.png

点击确定，系统就会自动的创建两个节点的流程，如下图。

图片地址：https://static.oschina.net/uploads/img/201506/14142137\_u2Pt.png

**增加节点连接线：**

图片地址：https://static.oschina.net/uploads/img/201506/14142146\_F0i5.png

**连接线**,把鼠标放在节点中间，按下然后拖拽，设计器就会以节点为中心画一条虚线，还有一个箭头，您可以连接到目标节点上去。

图片地址：https://static.oschina.net/uploads/img/201506/14142204\_WpWY.png

点击开始节点的中间向外拖拽，就可以增加连接线，如上图。

**修改节点标题：**

双击节点标签，就可以编辑节点名称，如上图。

**打开流程属性：**

点击工具栏上的流程属性，在节点上右键编辑节点属性。

**节点属性**

节点右键操作打开节点属性：

**新建节点：**

右键空白处，选择新增节点，可以新增加节点

**设置方向条件：**

右键方向条件连接线，选择设置条件，可以进行条件设置

**添加标签：**

标签没有计算意义仅仅是辅助说明流程图。

流程的节点类型图例

**cc把节点分为:**普通的、分流点、合流点、分合流点、子线程点。驰骋工作流用图形分别标示出来节点的类型。

图片地址：https://static.oschina.net/uploads/img/201506/23183924\_1FKr.png

**流程图例:** 我们通过下一张图片可以看到不同的节点类型使用不同的形状标识出来。

图片地址：https://static.oschina.net/uploads/img/201506/23183928\_IRW8.png

**流程模式:**就是流程运行的模式类型，cc目前支持的类型有线性流程、父子流程、同表单分合流、异表单分合流、我们所分析的流程一般都是从这4种流程模式上延伸出来的，或者说都是它们的变形。

**节点图标:** 如果用户喜欢使用节点图标的方式，展现流程图，可以使用流程设计器上的

创建简单的流程步骤

**第一步：**画出流程图。

**第二步：**设计表单。

**第三步：**设置方向条件。

**第四步：**设置节点访问权限

**第五步：**检查并运行流程。

这类的流程被一个人发起以后，整个流程的生命周期，在同一个时间点，只能有一个人处理，不存在多线程问题。所以这类的流程的轨迹是一根线。我们称为线性流程。

图片地址：https://static.oschina.net/uploads/img/201506/26174806\_QVkI.png

流程检查的作用

在流程设计器的工具栏目中有流程【检查】按钮，此功能为我们处理如下内容。

图片地址：https://static.oschina.net/uploads/img/201507/09145406\_931b.png

检查流程是否有基本的设计问题，比如：是否设置的岗位，人员，该岗位下是否有人员。

(一) 检查是否在流程分支时增加了方向条件。

(二) 为每个节点表修复字段。

(三) 修复流程报表表NDxxxRpt表。

(四) 修复查询视图。我们会在以后为此功能增加更多的用途。

节点方向与节点的方向条件

**关键字：**节点方向方向条件

**技术信息：**节点方向存储在WF*Direction表里，节点的方向条件存储在WF*Cond表里。

节点的方向条件类型分为五种,分别为：按节点表单字段计算，按岗位计算，按部门计算，按SQL计算, 系统参数。每种类型应用在不同的场合里。

**按节点表单字段值计算：**这是通用的一种常见的计算方式，比如：报销金额、请假天数、一个枚举类型的值。cc在运行时判定设定的条件是否成立。

图片地址：https://static.oschina.net/uploads/img/201507/09145502\_AWzC.png

**设计技巧**：对于方向条件判定比较复杂的情况，您可以为表单设置一个隐藏的字段或者一个隐藏字段或者一个枚举类型的值，这个值在是动态的计算出来的，来解决转向的问题。

**按岗位设置方向条件**：就是按当前操作员具有什么样的岗位才能执行此操作。

图片地址：https://static.oschina.net/uploads/img/201507/09145523\_atyn.png

**说明：**上图中一个请假流程，不同岗位的人员所走的请假路线是不同的。只要当前的操作员具有该岗位这条路线对它来说是通的。

**按部门计算：**与按岗位计算相同，就是当前操作员人，具备那些岗位才可以走通这条路线。

**按SQL计算:** 在文本框里设置一个sql，它返回一行一列，此值必须为0 或大于等于1的正整数,来表示该条件是否成立。SQL 支持cc的表达式。

图片地址：https://static.oschina.net/uploads/img/201507/09145549\_WMxK.png

**方向条件优先级设置：**因为每一个方向都已经设置上了方向条件，cc在执行的过程中是按一定的顺序来取这些方向条件。如果判定了这个条件成立，cc就会认为这条方向是通过的从而放弃判断其它的条件。

图片地址：https://static.oschina.net/uploads/img/201507/09145606\_YLF7.png

在特定的场合下，方向条件的优先级别的判断非常重要的，比如：一个人既是科长岗位，又是科员岗位，在岗位条件判定中要优先判定他的科长岗位路线。在节点表单字段判定方式也是如此。

其它: 2012年10月23日起cc的方向条件支持各种类型的条件同时存在。

流程属性

基本配置

流程编号

**流程编号规则：**流程编号是一个string类型的三位数的编号，编号从001开始，最大999, 也就是在整个ccBPM流程系统中最多有999个流程，这个范围足够您使用，不必担心，流程编号全局唯一。

图片地址：https://static.oschina.net/uploads/img/201506/15225858\_z4yo.png

**节点编号规则：**节点编号是流程编号转化成int类型加上两位数的序号，比如流程编号是001，节点编号就是101开始，一直到199。也就是说在一个流程里最多有99个节点。如果流程编号是010，那么它的节点编号就是1001，1002, …. 1099。节点编号是int类型，节点编号。在cc里面我们把string类型的主键习惯称为编号，把int类型主键习惯称呼为ID。比如：流程编号，节点ID。

**技术资料：**流程信息存储在WF*Flow表里，节点信息存储在 WF*Node表里，每个字段的中文意思，在您安装系统之后就会有详尽的描述。

字段的详细信息。

图片地址：https://static.oschina.net/uploads/img/201506/15225657\_hDPF.png

流程标记

**关键字：**流程标记 FlowMark

**说明：**此属性可以为空，适合高级开发者，为流程设置一个好记的名称，必须是字母数字下划线组合。设置流程标记的意义在于，当流程的编号发生变化时，可以使用流程标记找到该流程。比如我们做一个固定资产采购流程，这个流程的编号是005，我们需要把该流程应用到其他项目上去，但是到了其他系统上的时候该流程编号并非是005了，没有流程标记就会使用流程编号，就会导致业务逻辑写的太死。有了流程标记后，就可以使用流程标记替代流程编号编写业务逻辑。就类似于孩子有一个大名，还有一个小名，通过两个名字都可以对应到这个孩子，但是标记在一个系统中不允许重复。在高级的开发中，调用ccbpm的api时，可以不使用编号而使用标记调用api。如果您使用流程事件实体来书写业务逻辑时，您必须为该流程设置一个标记。如何使用流程事件实体FEE来书写业务逻辑，请参考下一章节。

一个流程有一个固定编号，同时您也可以为它定义一个标识，此标识是在全局唯一的，也可以称之为助记码。

在调用API时可以使用助记码来调用API。

**例如：**创建一个流程此流程的编号为001，在新建时指定它的流程标识为ShengJuQingJia。在调用ccflow的API时就可以使用如下语法。

\*\*创建一个工作：\*\*  
  
BP.WF.Node\_CreateBlankWork(‘ShengJuQingJia’,null,null,null,0)  
  
\*\*删除一个流程:\*\*  
  
BP.WF.Flow\_DoDeleteFlowByReal(‘ShengJuQingJia’’,11111,false)  
  
CCBPM有一个api是可以把流程标识转化成流程编号:  
BP.WF. TurnFlowCodeToFlowNo(string flowCode)

流程类别

**说明：**流程类别就是对流程的归类，合理的归类能够让系统更加清晰，流程类别是一个树形结构。

流程类别没有计算意义，仅仅起一个归类作用，在流程树上点击右键，就可以编辑流程类别。

图片地址：https://static.oschina.net/uploads/img/201507/07100845\_93I1.png

**技术资料：**流程类别对应的表是 WF\_FlowSort，该表是一个树形结构表。

图片地址：https://static.oschina.net/uploads/img/201507/07101203\_Mdfw.png

ccBPM流程事件实体

**关键字：**ccBPM 流程事件实体 FEE

**说明：**本属性适合高级开发者，该属性只读。流程事件实体全称是FlowEventBase 简称 FEE。流程事件实体是Flow Event Entity。它是从ccBPM指定的基类BP.WF.FlowEventBase上集成下来的一个类，重写相关的事件方法，达到实现自己的业务逻辑目的。

图片地址：https://static.oschina.net/uploads/img/201507/07102259\_RpTU.png

BP.WF.FlowEventBase是ccBPM定义的一个基类，为开发者提供可以被继承下来并且重写相关的方法而实现特定的个性化的用户业务逻辑。

**流程事件：**流程在运动过程中，会发生很多事件，比如，节点表单装载前，节点保存前、保存后、节点发送前、发送成功时，发送失败实时。CcBPM可以通过重写这个事件实体来表达自己的业务逻辑。

我们以表单的保存事件为例来说明如何对明细表求合计。

图片地址：https://static.oschina.net/uploads/img/201506/15230323\_QcTf.png

首先我们写一个FEE从BP.WF.FlowEventBase上继承下来，然后重写它的保存后的方法，希望实现的业务逻辑是在保存后，自动计算出来明细表中的合计数据。

请参考: D:ccflowtrunk DemoAndTestingBP.DemoEventBaseF001.cs文件。

图片地址：https://static.oschina.net/uploads/img/201506/15230419\_MooX.png

重写该实体必须重写 FlowMark 这个属性，就是给流程标记，在您编写完成后，编译您的工程然后填写上流程标记属性，ccBPM系统就会自动的把该文本框的标记信息填写上。

**相关阅读：**如何把ccBPM的待办写入其他的新系统？

在这里你可以重写流程事件类实现该它，如下图。

图片地址：https://static.oschina.net/uploads/img/201506/15230511\_rtpQ.png

重写流程事件被重写的SendSucess 方法。

是否可以独立启用

**关键字：**ccBPM 隐藏子流程

**默认值：**可以启用。

只有可以启用的流程才能出现在发起列表里，如果该流程为子流程，不能独立发起，就不要让其显示在发起的列表里。

**应用场景：**一个财务付款流程，该流程不能独立发起，只能被其他流程所调用，所以该流程不能出现在流程发起的列表里，就要隐藏起来。

图片地址：https://static.oschina.net/uploads/img/201507/07103415\_yQIQ.png

是否是数据加密流程(MD5数据加密防篡改)

**关键字：**流程数据加密 md5 数据保密流程数据防篡改软加密

**设置方式：** 对工作流引擎的数据加密研究, 流程数据的加密方案与实现过程.

图片地址：https://static.oschina.net/uploads/img/201507/07103402\_JA0D.png

**需求背景**

1, 流程数据加密是为了防止流程数据被篡改的需要，从而造成对单位的不必要损失。

2, 流程数据篡改是具有管理员身份与非法入侵者，尤其是防范业务人员与具有系统管理员结合做坏事情。

比如: xxx公司在xx年曾经出现过这样的情况, 业务人员与管理员勾结,导致单位200万元损失，虽然最后查出来，罚款判刑,但是这是事后处理.

3, 并非所有的流程数据都需要加密, 比如请假流程就不需要，但是报销或者其它具有敏感数据的流程就需要了。

4, 被篡改后的数据需要警报，并写入log日志，还要不能被执行下去。

5, 要求每个节点的数据都要加密，整体流程数据也要加密。

**处理方案:**

1, 在流程中增加一个属性 IsMD5 是否要加密. 默认是否, 不加密.

2, 用户信息中包括一个盐值，用于后期的加密运算。

3, 凡是加密过的流程在每个节点表单上增加字段MD5 长度为 char(36) 的字段，这个字段用于存储这个节点数据的MD5值.

4，在NDxxRpt 表中增加字段MD5 用于存储流程数据加密。

5，生成MD5值: 在如下两个步骤中生成MD5值,并存储到相关的物理表中.

1, 节点发送成功后.

2, 节点发送失败时.

6, 生成MD5值的方法.

1, 把节点表单字段排除有些字段是默认值的随即获取的字段（比如:获取当前时间,当前操作员....），组合成一个大的文本, 然后通过获取审批用户盐值，一起MD5加密.

2, 加审批用户盐值意义是防止通过彩虹表暴力破解.

7, 检查是否被篡改的步骤.

1，获得节点与流程数据表的数据, 排除关键性的字段.

2, 执行MD5加密。

3，将两者串比较。

8, 在如下情况下检查MD5值是否正确.

**有涉及到打开节点表单的地方:**

1, 打开待办工作时.

2, 通过流程日志打开节点表单时.

3, 打开抄送工作时.

4，打开地途工作时.

有涉及到打开流程数据的地方.

1, 打开工作轨迹时的流程数据时.

以上有任何不对地方，都要禁止打开并写入log的文本日志中.

**对应的流程Demo:** \流程树\表单解决方案\请假流程(轨迹模式)

是否自动计算未来的处理人？

**关键字：**流程未来节点处理人

**业务背景：**一个流程在启动起来后，是可以对一些节点计算出来处理人是谁，流程的走向。对于另外一些节点处理人有可能需要相关的人员调整的。在一些审批的环境下，需要把能够计算出来的节点处理人在发起时计算出来，并显示给操作员。该属性默认为false, 启用后， ccBPM就会为已知道的节点填充处理人到WF\_SelectAccper表中，如果启用了审核组件并且查看未来的步骤，就会显示出来。启用的目的就是让发起人更能直观的看到自己的工作需要那些人审核。

流程应用类型

根据数据投递的权限范围不同，ccBPM区分为普通类流程与项目类流程。普通类型流程的流转的组织机构定义在整体单位中，对于项目类流程它们的流转范围在整个项目组中。

**普通类型:**普通类型流程是区别于项目类流程而言，它是我们通常所说的请假、报销等流程。它的组织权限范围在整体单位中。

**项目类流程:**概要说明:

1, Prj为项目工程所用的一些扩展应用,项目流程也可以称为工程流程，是为了建造一个跨期比较长的工程在每个环节上需要流程控制。

2, 项目流程权限控制比较特殊，比如一个人在a项目中是项目经理，在b项目中是工程师。这类的流程在节点权限控制时会遇到局限在本项目中流转。

3，一个项目会有资料树的概念，每一个有关项目的流程具有一些文档，这些文档会于项目关联。

4，项目流程ccBPM在表单设计中有一个约定。每个节点表单中（流程表单除外）都必须有PrjNo,PrjName两个属性.

在流程设计时必须指明此流程为项目类型的流程。

**环境搭建**

1. D:ccflowExpandingApplicationPrjPrj 复制到 D:ccflowtrunkCCFlow 中去.

2. 在CCFlowwebapplication 中引用 D:ccflowtrunkExpandingApplicationPrjComponentsBP.PRJBP.PRJ.csproj

3. 在目录: D:ccflowtrunkCCFlowDataUser 下创建一个Prj.Data 目录. 这个目录将会放资料树。

D:\ccflow\trunk\CCFlow\WF\DataUser\PrjData\Data 资料树数据．

　　D:ccflowtrunkCCFlowDataUserPrjDataTemplete　资料树模板数据.

1. 编译该项目.

**创建您的第一个工程项目:**

1, 在流程设计器中建立一个项目, 该项目的编号为 0001.

2, 为该项目设计项目树模板.

2.1 创建目录: D:ccflowtrunkCCFlowDataUserPrjDataTemplete001 2.2 在以上目录下创建目录模板, 系统中读取文件名.文件里面可以没有任何内容. (为了方便期间您可以把 D:ccflowExpandingApplicationPrjTemplete\*.\* copy 里面测试)

目前只支持一级的资料树.

2.3 说明:在您上传资料后，系统就会自动的把文件上传到 D:ccflowtrunkCCFlowDataUserPrjDataData001 下面。

1. 为该项目设计岗位与人员.设计资料树节点访问规则.

2. 新建一个流程，在流程属性中设置该流程为项目类型的流程.

3. 为该流程的每个节点表单都要增加一个PrjNo,PrjName字段.

6, 在表单上，增加一个超连接连接地址为 ./Prj/DocTree.aspx?FK*Prj=@PrjNo&FK*Node=当前节点ID. 系统就会打开项目树了.

时效性规则

整体流程需要完成的时间，一个流程从发起开始如何限制与计算该流程整体需要完成的时间，如果完成不了如何进行有效的提示，比如纳税人的开业登记流程按照规定不能超过n天，办理过程需要给申请人反馈结果。

此规则有两种，第一是按照每个节点需要的时间相加，另外一种是由开始节点的发起人输入指定字段值来确定该流程需要完成的时间。

草稿

**定义：**流程发起工作做了一部分，就关闭了下次处理，对这件工作我们就称为草稿。

应用背景：草稿也是常用功能之一，它是针对于，开始节点的表单才有意义。它类似于我们的邮件草稿，当您发起一个工作后，录入了一些数据，有与其他的原因，需要暂停该工作，但是数据需要保存下来，等待我下次打开在继续编辑。

**草稿的规则：**草稿规则有三种，不设置草稿、保存到待办列表、保存到草稿箱。

**不设置草稿：**始终只有一个草稿，您启动了流程打开表单后，系统就自动生成一个workid，您输入了一半，保存并关闭后，不发送该流程，这个时间就是一个草稿，下次您在启动这个流程，系统就会把这个草稿调出来，让您编辑，如果发送下去该草稿就没有了，草稿的状态变成了待办。

**保存到待办：**当您启动一个流程后打开一个开始节点表单，系统就会产生一个workid，输入部分数据，如果您保存了，这个时候系统就会产生一个待办，放入了待办列表里。如果您关闭了，下次可以从待办列表里找到，并打开它继续编辑，如果您发送了待办就没有了，工作流转到下一个节点上去了。

**保存到草稿箱：**当您启动一个流程后打开一个开始节点表单，系统就会产生一个workid，输入部分数据，如果您保存了，这个时候系统就会产生一个待办，放入了待办列表里。如果您关闭了，下次可以从草稿箱里找到，并打开它继续编辑，如果您发送了草稿箱就没有了，工作流转到下一个节点上去了。

数据存储模式

**关键字：**流程数据存储模式

说明：cc把流程数据模式分为两种：轨迹模式与合并模式。

**什么是轨迹模式？**ccBPM创建一个节点就会创建一个表与此节点相对应，此节点上的数据采集都记录在这个节点表中，比如节点编号为101那么ccBPM就会自动创建一个ND101的数据表，所有对此节点的数据采集都记录在这个表里，下一个几节点编号为102就创建一个102的表。基于这种模式的数据存储流程在运动过程中，就会忠实的记录下来每个节点上对应的人员数据更改。

比如：在费用申请流程里（流程编号为001），有三个节点费用申请单101，部门经理审批102，总经理审批103。申请人申请的金额是100万, 部门经理审批的是110万, 总经理审批的是90万。

那么在cc就会创建4个表 ND101, ND102,ND103，ND1Rpt.

流程启动时在ND101,ND1Rpt中各插入一条数据。费用金额字段为100万，运行到第二个节点后ND102与ND1Rpt中的金额会是110万，运行到最后节点时ND103与ND1Rpt的字段值为90万。在ND1Rpt表中始终反应的是最新修改的数据。在各个节点表单对应的数据表里反应的是当前自己修改的数据.

ccBPM就可以把这种轨迹的模式自动的记录下来，我们把这种模式叫做轨迹记录模式。

**什么是合并模式？** 就是所有的节点表单都对到应流程数据表，每个节点上表单的数据更改都是编辑同一条记录，这种模式是运算速度快，但是不能反应出来数据变化的情况。

这种模式下，ccBPM就不会在自动创建节点表，而是所有的该流程的节点表都映射到了NDxxxRpt表。

**对应的测试Demo:**

图片地址：https://static.oschina.net/uploads/img/201507/07113748\_IOmt.png

合并模式：

图片地址：https://static.oschina.net/uploads/img/201507/07113823\_xkyZ.png

从这两个流程上可以结合数据库就可以看到数据的变化情况。

**总结：**如果一个流程需要保留每个节点的数据轨迹，那末就要使用轨迹模式，比如：我们做一个资金预算审批流程，第1个节点填写一个预算金额，在第2个以及第2个以后的节点上需要改变这个金额，并且需要在对审批金额这个字段需要保留各自的修改痕迹，这样的情况下需要使用到轨迹模式。

是否外部用户参与流程?

**关键字：** ccBPM 外部用参与流程

**定义：**分组织结构内的人员参与的流程，称为外部用户参与流程也叫客户参与流程。

**应用背景：**我们给学校设计一个流程，学校教师是内部用户，是组织结构内的用户，学生就是外部用户。如果我们制作一个学生请假流程，一条流程中有多个节点是学生参与的。这样的节点是学生参与的，我们就称为该流程是客户参与流程，学生参与的节点就是外部用户参与节点。

这种应用场景非常之多，比如我们为税务局设计一个系统税务系统工作人员是内部用户，纳税人就是外部用户。

我们为企业设计一个erp ，企业内部的组织机构人员是内部用户，供应商、客户就是外部用户。

如果一个流程中的一个或者一个以上的节点是由外部用户参与的，我们就称该流程是外部用户参与流程，客户参与的节点就是外部用户执行的节点。

**提供的外部用户执行的api:**

BP.WF.Dev2InterfaceGuest.*.* Ccflow系统提供了外部用户登录接口，

**关于Guest用户:**

如果有具有外部用户参与流程的应用环境中，您需要在Port\_Emp的用户信息表中预置一个Guest的用户，该用户具备最低的操作权限，最低的查询权限，并且不需要给他任何岗位。

他是一个模拟用户，就是说当一个外部用户登录时(比如学号为201001003，姓名为:张三)的用户登录时, 需要调用BP.WF.Dev2InterfaceGuest.Port\_Login() 接口，从而记录外来用户编号，同时也让Guest 这个内部账号登陆了。这样cc就存储了两份信息。该同学在系统中具备两个身份一个是他自己，另外一个是Guest 内部模拟信息。

**Guest用户执行节点**如果该流程是外部参与流程，就在流程属性里做上标记，如果流程的一个节点是外部用户处理的节点，那就在该节点上做好设置，现在他就会流转到该外部用户身上。

标题生成规则

**关键字：**流程标题流程标题生成规则 ccBPM表达式.

标题：是流程标题，一个流程在开始节点是必须有一个标题字段Title ，并且这个字段不可别删除，默认为隐藏的。标题字段用于待办、在途的工作列表上，是能让使用者知道这是那一条流程。

标题字段不能修改，它是系统根据一定的规则自动生成的。

**默认生成规则为：**"@WebUser.FK\_DeptName,@WebUser.No,@WebUser.Name在@RDT发起."; 解析为：财务部,zhangna,张娜在11月12号10：20发起。

流程设计人员可以根据自己的需要设计自己的规则，规则表达式中支持节点表单字段。

**比如：**"@wenjianmingcheng在@RDT发起."; 其中@wenjianmingcheng是节点表单的数据库字段。

对于线性流程来说，只有一个标题，您只要为开始节点表单的Title字段设置默认值中中填写规则就可以了。

对于分合流程来说，流程发起后，开始节点的标题规则都是一样的，只是在分流节点中也要个性化设置子线程标题，所以子线程的标题生成规则设置在分流节点表单Title隐藏字段的默认值中。

**备注：**合理的设置流程标题规则，可以让操作员一眼就看出该流程的内部信息。

**扩展阅读：**如果要生成标题的表单是一个流程表单的字段怎么处理？请按照如下步骤处理:

第1：在节点表单上设置一个隐藏的字段此字段的字段名与流程表单的字段名保持一致。

第2：绑定流程表单时使用自定义方案，有定义字段属性的选择项目，是否写入节点表单，打上对勾就可以了，

如下图：

图片地址：https://static.oschina.net/uploads/img/201506/16090354\_zEgC.png

流程表单权限控制界面

单据编号格式

**定义：**用户希望每一笔业务都能实现一个按照自己定义的格式自动增长的编号，就叫单据编号。

**关于WorkID的描述：**一个流程启动起来以后，系统就会自动产生一个workid，这个workid是一个Int64类型、全局唯一的主键值。这个值贯穿了整个流程生命周期，所有节点表单、流程表单的主键都使用该值。

如果要设置单据编号，分为两个步骤。

**第一步，在节点表单中，创建一个文本框，定义名称为BillNo。**

图片地址：https://static.oschina.net/uploads/img/201512/29161309\_JHZ8.png

是否勾选“是否根据名称生成标签”，根据自己的方式确定。

如果勾选了，那么就会生成一个BillNo的标签（Label）。

图片地址：https://static.oschina.net/uploads/img/201512/29161323\_ch9D.png

**可以右键修改这个标签，改成你想要的文字信息。**

图片地址：https://static.oschina.net/uploads/img/201512/29161342\_oTUd.png

如果不勾选，就不会生成。

点击保存后，第一步完成。

**第二步，点击流程属性。**

图片地址：https://static.oschina.net/uploads/img/201512/29161404\_WMR8.png

**在基本配置栏中，设置单据编号格式，如下图。**

图片地址：https://static.oschina.net/uploads/img/201512/29161419\_pLPv.png

**上图中输出的单据号为**：CM2015-12-0001

**他的格式为**：特别符号+日期格式+流水号。

**其中特别符号可以是中英文；**

**日期格式是通用的日期转化格式**，例如：yyyy-MM-dd，然后用 “{ }”大括号分别包括起来，例如：｛yyyy｝-{MM}-{dd}；

**流水号比较特殊，是固定的模式**。｛LSH4｝表示一位4位数字，数字的范围在2-7之间。

**父子流程中的单据格式为**：｛ParentBillNo｝-{LSH3}。例如：如果以上流程为父流程，那么子流程成的单据号为：CM2015-12-0001-001

由此上两步，就可以让表单自动生成编号。

**下图为流程运行时，生成的单据号：**

图片地址：https://static.oschina.net/uploads/img/201512/29161444\_canA.png

备注的表达式

**关键字：**流程备注的表达式。

什么是流程备注？我们知道ccBPM有个流程标题，流程标题用来让操作者区别出来一条流程与另外的一条流程的，因为流程标题的长度与显示限制，操作员希望能够通过另外的一列获得流程的另外信息，那么这一列就是流程备注。

流程的备注的表达式在流程属性里，他的使用规则与流程标题一致，流程备注是流程系统的一个字段，在开始节点保存或者发送后，流程备注自动就会被自动填充。

流程备注根据需要会显示在待办、在途、挂起、删除列表中。

图片地址：https://static.oschina.net/uploads/img/201507/07120114\_mKlx.png

批量发起字段

**子字段与上一个属性一起起作用。**

**批量发起**，就是在对于同一个流程来说，用户一次想启动多个实例，也就是多个流程实例。

如果批量发起启用，此字段设置有效。此字段定义，一次发起多少个流程时，需要填写那些字段？

图片地址：https://static.oschina.net/uploads/img/201507/07120340\_dsvu.png

**定义的格式为，字段英文名称中间用逗号分看，结合是否可以批量发起流程属性起作用。**

是否可以批量发起流程？

**关键字：** 流程批量发起

**应用背景：**有的流程需要批量发起，就是，我一次要发起多个流程。

**流程demo:**  我们为您提供了一个批量发起的demo流程。

图片地址：https://static.oschina.net/uploads/img/201506/16092239\_7I94.png

**请看如下属性设置：**

图片地址：https://static.oschina.net/uploads/img/201507/07120224\_LKN4.png

**设置启用批量发起属性后，在发起列表里就有可以列出来：**

图片地址：https://static.oschina.net/uploads/img/201506/16092326\_Jxsq.png

**点击后进入批量发起界面：**

图片地址：https://static.oschina.net/uploads/img/201506/16092334\_cSCZ.png

**点击执行按钮后：**

图片地址：https://static.oschina.net/uploads/img/201506/16092344\_0elz.png

说明一次发起了多个流程。

备注

用于说明流程信息，没有计算意义。

启动方式

应用背景与定义

**应用背景：**通常模式下的流程启动是手工的启动，就是用户从一个发起列表，点击流程名字，就启动了该流程。但是有的时候，是系统自动发起该流程，比如：周例会流程，用户希望每个周都要启动例会通知流程这个启动是让系统自动发起而非人工发起。启动方式定义了流程如何手工运行与自动运行的模式。

**关键字：**自动发起定时启动按数据集启动触发式启动流程。

流程的自动执行包括，流程的自动发起与节点的自动执行两大部分。下面分别介绍这两种的实现方法.流程的自动发起.

图片地址：https://static.oschina.net/uploads/img/201507/07144122\_4zKm.png

启动设置

**流程的发起包含如下几种模式:**

**0、手工启动**

默认的模式，就是手工的发起流程。

**1 、指定人员按时启动**

指定特定的人在特定的时间发起流程比如 zhangshan@01:01@12:01 让张三在每天的 1点零1分，与12点零1分发起此流程。

图片地址：https://static.oschina.net/uploads/img/201507/07144348\_TcB5.png

**2、数据集按时启动**

为流程设置一个数据源，系统按时读取这个数据源发起流程。

表达式的意思是: 让zhangshan 在每天的01:01 与12:01两个时间点执行自动启动

数据源是从开始节点表单获取.

右图是填充开始节点的主从表的sql, 两者都是一个集合,对于主表要有一个唯一的MainPK字段，以防止数据重复发起。数据源的列与主表的字段一致。

图片地址：https://static.oschina.net/uploads/img/201507/07144658\_Psfl.png

**对于要启动的流程要求:**

要自动启动的流程在开始节点表单里必须要有MainPK字段，这个字段可以是隐藏的，长度根据产生的MyPK需要，增加此字段目的是避免流程启动重复。流程在启动起前就要检查一下开始节点中的MainPK字段，如果有类似的数据说明它已经发起来了，就放弃它。

**设置数据集按时启动的方法**

1, 在流程属性中的运行方式设置启动的顺序。

当运行方式为: 按天启动 执行内容 @01:01@02:01 表示01点:01分，与02点01分分别执行一次.

当运行方式为: 按小时启动 执行内容 @:01@:10@:20@:30@:40@:50 表示每间隔10分钟执行一次.

2，流程属性-》设置自动发起数据源，为开始表单设置数据源，（如果有从表也会让您设置从表的sql）。

主表集合中必须包含如下几个列:

Starter 发起人 MainPK 主键(唯一的主键,是为了不重复调度.)

**从表数据集合要求:**

RefMainPK 关联主键的值.  
特别说明: 两个SQL都是查询出来的集合.

3, 在开始节点属性中增加一个MainPK隐藏字段.

这个隐藏字段是用来保证数据发起不会重复。

4, 启动调度程序.

让其按时执行。

**3、触发式启动**

工作原理:第三方软件向特定的表(WF*Task)中写入数据，ccBPM就会读取这张表来完成流程的发起，发起成功后就把这条记录设置成已经发起的状态，并把信息写入WF*Task表的Msg里面。

关于WF\_Task表字段说明：

**特别说明：**如果需要自动启动，就要启动ccBPM的服务。

|  |  |  |
| --- | --- | --- |
| **字段** | **字段中文名** | **解释** |
| MyPK | 主键 | 唯一的主键不要有中文名称，长度尽量短，要求全局的流程的每 一条流程这个是唯一的。 |
| FK\_Flow | 流程编号 | 要发起的流程编号 |
| Starter | 发起人 | 发起人编号 |
| Paras | 字段参数 | 字段参数，格式为：@字段英文名1=字段值1@字段英文名2=字段值2比如: @StuNo=001@StuName=张三@StuAddr=山东济南 这个参数要与开始节点表单字段匹配， 就是说要发起的的流程开始节点表单中要有StuNo,StuName,StuAddr 才能匹配上。 |
| TaskSta | 任务状态 | 默认为0： 0=未发起，1=成功发起，2=发起失败。 此字段不需要程序写入值。 |
| Msg | 执行的消息 | 流程发起后的信息，成功的话就返回执行成功的信息，不成功就写入失败信息。 此字段不需要程序写入值。 |
| StartDT | 发起时间 | 如果为空就表示立刻发起，如果设置好发起的时间就按指定的时间发起， 指定的时间不能超过分钟。时间格式为yyyy-MM-dd HH:mm 比如:2012-06-01 01:01 也可以书写成2012-06-01 |

流程启动限制规则

流程启动限制规则

**定义**：并非每个流程都可以随意发起，它的发起需要一定的条件约束，比如：工作周报流程，就是每周发起一次，不能发起两次，纳税人注销流程，一个纳税人只能发起一次注销，不能两次注销。

流程发起限制是很多流程常用的一种行为，但是开发者也不重视去配置，会导致数据紊乱，统计出错。

比如有的流程是按照时限发起的，每天一人一次，每周一人一次等等。

有的流程是按照一定的主键发起的，比如纳税人注销流程，一个纳税人编号不能启动多次，因为一个纳税人只能启动一次注销流程，如果启用了多次就会出现业务逻辑的错误。

**配置界面：**

图片地址：https://static.oschina.net/uploads/img/201506/21221322\_O3su.png

根据各种应用场景ccBPM抽象出来如下几种模式，请根据自己的应用场景对号入座。

**不限制：默认不限制发起次数.**

**每人每天一次：**每人一天只能发起一次，如果需要时间限制就要在参数里设置，设置规则是：@时间点1-时间点2@时间点3-时间点4@时间点n-时间点n1

比如：@08:30-09:00@18:00-18:30，解释：该流程只能在08:30-09:00与18:00-18:30两个时间段发起。如果设置为：@08:30-09:00 只能在此时间点发起。

设置方式:

图片地址：https://static.oschina.net/uploads/img/201506/21221514\_DS6f.png

**每人每周一次：**每人每周天只能发起一次，如果需要时间限制就要在参数里设置，

设置规则是：@英文周,时间点1-时间点2@英文周,时间点3-时间点4@英文周,时间点n-时间点n1

英文的周是: 周日 Sunday,周1 Monday,周2 Tuesday,周3 Wednesday,周4 Thursday,周5 Friday,周6 Saturday。配置时注意不要拼写错误。

比如：@Sunday,08:30-09:00@Monday,18:00-18:30，解释：该流程只能在周日，与周一指定的时间段内发起。

设置方式:

图片地址：https://static.oschina.net/uploads/img/201506/21221609\_9uPG.png

**每人每月一次：**

每人每月只能发起一次，如果需要时间限制就要在参数里设置，

设置规则是：@-日期,时间点1-时间点2, 多个时间段用@符号隔开。

比如：@01,08:30-09:00，解释：该流程只能在每月的1号08:30-09:00发起。

设置方式:

图片地址：https://static.oschina.net/uploads/img/201506/21221636\_BNdO.png

**每人每季一次：**

每人每季只能发起一次，如果需要时间限制就要在参数里设置，

设置规则是：@MM-dd,时间点1-时间点2, @MM-dd,时间点1-时间点2, @MM-dd,时间点1-时间点2,。

比如：@01-01,08:30-09:00,@04-01,08:30-09:00,@07-01,08:30-09:00, 解释：该流程只能在1，2，3季度的第一天的08:30-09:00发起，其他时间点不能发起。

设置方式:

图片地址：https://static.oschina.net/uploads/img/201506/21221713\_W8kn.png

**每人每年一次：格式与季度相同.**

发起的列不能重复,(多个列可以用逗号分开) ：未完成

应用场景：有一些流程不能发起重复，比如我们为税务做一个纳税人注销审批流程，一个纳税人只能注销一次，如果发起多次就是逻辑错误，所以这个限制就应用到了这个场景。配置界面如下。

图片地址：https://static.oschina.net/uploads/img/201506/21221757\_ycg9.png

说明规则参数，可以是多个列，多个列用逗号分开。比如：纳税人月申报流程。一个纳税人只能在一月发起一次。

图片地址：https://static.oschina.net/uploads/img/201506/21221821\_CTZ1.png

**设置的SQL数据源为空,或者返回结果为零时可以启动. ：未完成**

流程启动的限制规则在以上条件都不能很好的满足时，ccBPM提供了按照SQL查询的规则，编写一个sql语句返回一行一列，如果信息是0，就是可以启动，非0就不可以启动。

图片地址：https://static.oschina.net/uploads/img/201506/21221857\_cMjt.png

**设置的SQL数据源为空,或者返回结果为零时不可以启动. ：同上**

所有的启动限制，我们已经编写完，流程设计人员根据自己的业务背景选择合适自己的发起限制规则。

流程相关功能

检查报告

流程的相关功能就是对该流程执行的操作。

图片地址：https://static.oschina.net/uploads/img/201507/07152149\_8fEo.png

流程的相关功能在流程属性的右侧，请参考上图。

关键字：流程体检检查报告

用途： ccBPM在升级时，在您设计完流程后，我不知道那里设计的不对，或者那里设计的有问题，我就要执行一次流程体检。

流程体检的意义：

1，  自动创建应有的数据解构，数据表。

2，  把旧的版本流程升级到新版本中来。

3，  体检出部分，并非全部流程设计的错误，设计错误的地方用红颜色提示，警告用黄颜色提示。并非所有的流程设计错误，ccBPM都能体检出来，只有完整的运行完流程才能把流程设计中的错误找出来。

4，  执行流程上的检查报告与流程设计器上的【检查】功能是一致的。

图片地址：https://static.oschina.net/uploads/img/201506/21222352\_apud.png

流程检查如下图：

图片地址：https://static.oschina.net/uploads/img/201506/21222415\_2egi.png

**出现的警告：**

图片地址：https://static.oschina.net/uploads/img/201506/21222443\_JwNl.png

设计报表

流程的报表设计需要了解流程运行的全部过程、了解系统生成的数据表结构才能进行设计，所以这里仅概要介绍说明，详细的设置请参考本说明书的流程数据报表设计章节。

**说明**：此功能与流程设计器中的，流程报表定义一样。

图片地址：https://static.oschina.net/uploads/img/201507/07153015\_4tEX.png

**基本知识：**

视图就是把一个流程中的各个表连起来，组合成新表，提供用户查询。

视图仅仅提供对流程数据的查询、分析。用户不可以修改这些数据。

各个岗位关心的数据项目不同所以系统允许设计多个视图。

设计完成一个视图后,就可以利用4个视图组件来发挥视图的数据作用。

定义完的流程都反映在前台流程报表的菜单中。

**视图的组件：**

图片地址：https://static.oschina.net/uploads/img/201507/07153029\_XBQd.png

**查询组件：**提供给用户数据查询，双点查询的行可以实现卡片数据显示。

**分组分析:** 提供给用户分组分析、用户可以根据自己的需要任意组合分组条件、分析的内容。实现三种图形分析，极其方便灵活、所有的组合可能都可以完成。

**对比分析：**在数据分析中，我们经常遇到客户要查看一个部分的，一项数值指标升高或者降低了多少.升高或者降低比率是多少? 这个组件，让您完成任意的可能发生的组合。满足多种对比分析的需要。

**交叉报表：**利用这个组件您可以完成多纬度的交叉报表，实现任意可能的分析情况。

**视图的访问权限：**

视图有查看的权限，所以在设计视图时，需要定义岗位访问权限与人员访问权限。

**视图的设计：**

图片地址：https://static.oschina.net/uploads/img/201507/07153039\_aJjV.png

**选择字段：**系统列举了该流程上的所有的字段，选择字段后确定。

**字段派序**：您可以利用向上、向下的箭头进行对字段排序。

**字段命名：**您可以对字段中文命名。

**查询条件：**用于显示在组件上的查询条件。

删除全部流程数据

删除当前流程运行的数据，这个功能在流程测试的时候使用，并且只有admin可以使用。这个操作将会删除流程的所有数据，要慎重执行。

这些数据包括：流程引擎控制表数据、流程运行数据、流程历史归档数据。

按照工作ID删除单个流程

删除指定的workid数据：

如下图：

图片地址：https://static.oschina.net/uploads/img/201506/21223023\_sgbd.png

点击【执行】按钮后，就直接删除该流程的所有信息，这个流程将被删除到流程日志里。

重新生成报表数据

如果因为意外原因NDxxxRpt表被删除了，该功能可以重新把数据生成出来放入NDxxxRpt表里。在数据误删，丢失的情况想可以用该工具把数据找回。

图片地址：https://static.oschina.net/uploads/img/201507/07155124\_hAXH.png

设置自动发起数据源

此功能与流程的运行方式配合使用，详细请参考流程的运行方式。

流程监控

对于运行中的流程，进行监控，流程干预。

图片地址：https://static.oschina.net/uploads/img/201507/07155147\_SzSQ.png

点击流程标题可以对流程进行删除、移交、跳转、查看等功能执行。

**轨迹**：查看该流程运行的详细信息。

**删除**：直接删除该流程。

**移交**：把当前的待办工作交给其他人处理。

**跳转**：跳转到指定的节点，让指定的人员处理。

回滚流程

**需求背景:**

流程结束后，操作员发现此流程有一个错误，需要退回重新处理。对于ccBPM来说整体流程已经完成并且流程运行控制表已经被删除了该流程的数据。

**处理方案：**

在流程属性里的相关功能里增加一个相关功能，接受两个参数一个workid,一个回滚原因.

此功能只有admin用户才能操作。

**其它说明：**

1, 如果您是为了测试估计不能测试到效果，因为ccBPM在判断最后一个节点是通过时间来排序的，测试的数据经常时间点是一致的，精确到分钟。

2, 流程回滚后ccBPM会把日志写入wf\_track表里。

图片地址：https://static.oschina.net/uploads/img/201506/21223354\_C4xU.png

**流程回滚的应用背景：**

**场景1：**一件流程在完成后，发现这个流程有问题，需要退回到流程指定的节点上与当时该节点的处理人身上，让他重做。

**场景2：**一个主流程启动了n个子流程，他在监控子流程时发现子流程问题，需要重新去做。

这两种场景下，都是在流程完全处理完后，并且流程引擎已经把该流程归档后要求重新回滚到指定的节点与该节点的处理人身上去，让其在重新去处理。

**回滚流程对应的API：**

/// <summary>

/// 恢复已完成的流程数据到指定的节点，如果节点为0就恢复到最后一个完成的节点上去.  
  
 /// 恢复失败抛出异常  
  
 /// </summary>  
  
 /// <param name="flowNo">要恢复的流程编号</param>  
  
 /// <param name="workid">要恢复的workid</param>  
  
 /// <param name="backToNodeID">恢复到的节点编号，如果是0，标识回复到流程最后一个节点上去.</param>  
  
 /// <param name="note">恢复的原因，此原因会记录到日志</param>  
 public static string Flow\_DoRebackWorkFlow(string flowNo, Int64 workid, int backToNodeID, string note)  
  
 {  
  
 BP.WF.Ext.FlowSheet fs = new Ext.FlowSheet(flowNo);  
  
 return fs.DoRebackFlowData(workid, backToNodeID, note);  
 }

**回滚流程的设计需求：**

1，  回滚人接受到的信息是退回状态，而非流程的运行状态。

2，  回滚后流程数据表的状态由完成恢复到退回，最后结束节点恢复到回滚的点。

3，  恢复流程引擎注册表数据，当前表状态标识为退回。

4，  恢复流程工作人员表，把所有参与该流程上的工作人员都恢复里面去。

5，  写入一条退回信息，让接受人员能够看到退回的原因。

6，  写入到流程日志，保证它的完整可读。

流程表单树

**树结构的流程表单效果图：**

图片地址：https://static.oschina.net/uploads/img/201507/07160855\_oxs8.png

**设置步骤：**

**第一步：**节点属性—》表单—》节点表单方案—》选择表单树 类型

图片地址：https://static.oschina.net/uploads/img/201507/07160922\_fCw2.png

**第二步：**右键节点，选择绑定表单，如果表单库中没有合适的表单或者没有表单，需要现在表单库中新建表单。

图片地址：https://static.oschina.net/uploads/img/201507/07160953\_ik7e.png

经典表单树效果预览：

http://blog.csdn.net/jflows/article/details/46888325

绑定流程表单

**第一步：**选择绑定表单，进入选择表单页面

图片地址：https://static.oschina.net/uploads/img/201507/07161052\_hqIT.png

**第二步：**选择需要绑定的表单，如果表单库中没有合适的表单或者没有表单，需要在表单库中添加

图片地址：https://static.oschina.net/uploads/img/201507/07161113\_9M7F.png

**第三步：**绑定完成

图片地址：https://static.oschina.net/uploads/img/201507/07161133\_QjWu.png

**显示效果：**根据节点中配置的表单类型显示也不相同

**树表单类型：**效果图参照---》流程表单树

**自由表单：**节点中的自由表单也会显示出来

图片地址：https://static.oschina.net/uploads/img/201507/07161155\_cq6f.png

**禁用（对多表单有效）：**节点中的自由表单将不会显示

图片地址：https://static.oschina.net/uploads/img/201507/07161216\_8cJU.png

发起前置导航方式

内置URL方式

**介绍：**我们经常会遇到用户发起流程前，首先进入一个实体列表界面（比如项目列表，成员列表、供应商列表。），选择一个实体后把该实体的信息带入开始节点的表单字段里，我们把这个应用场景叫做发起前置导航方式。发起前置导航方式有如下几种应用场景，开发者根据需要进行配置。

图片地址：https://static.oschina.net/uploads/img/201506/21223656\_FVuK.png

**内置URL方式**

内置URL模式，就是系统已经有一个解释流程属性的配置信息，来呈现给用户的模式。

应用案例与设置1-¬简单模式

**首先要设置数据源：**

图片地址：https://static.oschina.net/uploads/img/201507/07163749\_O15S.png

**流程发起前选择人员（或者其他信息）**

图片地址：https://static.oschina.net/uploads/img/201507/07163535\_xCeH.png

**选择后，将选择的内容带入表单中**

图片地址：https://static.oschina.net/uploads/img/201507/07163548\_2UoQ.png

应用案例与设置2-父子流程模式

关键字：ccBPM父子流程导航

我们从流程案例说起：第一个流程为立项申请流程，第二个流程为项目验收流程。

立项批准后，下一步就是项目验收审批，从逻辑上来看他们是一对一的父子流程，就是一个父流程仅仅对应一个子流程，我们称为一对一的父子流程，否则就是一对多的父子流程。

图片地址：https://static.oschina.net/uploads/img/201506/21223819\_zBpe.png

在启动项目验收流程时，客户希望填写验收申请单之前需要一个项目列表，用户选择一个项目后就进入到项目验收申请的表单，并且把该项目的信息带入到该表单上，如下图所示：

图片地址：https://static.oschina.net/uploads/img/201506/21223840\_BgDa.png

当用户选择一个项目后，就进入项目验收申请流程，如下一步骤。

图片地址：https://static.oschina.net/uploads/img/201506/21223917\_FllS.png

表单的数据是通过Url自动带入的，并且URL信息中包含了父子流程关系。

我们把以上的应用场景，叫做前置URL模式的，实现父子流程。

**设置方法**

打开流程属性

图片地址：https://static.oschina.net/uploads/img/201507/07164847\_W9Pe.png

在发起前置导航方式设置为：按照系统的URL。

**发起前置参数：**

SELECT XMBH as No, XMMC as Name,XMBH,XMMC,XMDZ,XMJE,’082’ as PFlowNo,OID as PWorkID FROM ND82Rpt WHERE XMBH LIKE ‘%@Key%’ or XMMC LIKE ‘%@Key%’

**说明：**

需要配置一个查询语句作为实体信息列表的数据源。

查询中必须有No,Name两个列。

如果包含PFlowNo,PWorkID 系统就会认为他要调用父子流程。

自定义URL方式

顾名思义就是自己写一个url, 配置到参数里，请参考ccflow的WFStartGuide.aspx界面。

从开始节点copy数据

**关键字：**ccBPM 从开始节点copy数据数据

**应用场景：**

一个流程在发起前操作员期望，从历史的列表里查询到一个已经发起的流程以此流程的数据为基础然后进行编辑在提交发送。

这样做的目的就是提高工作效率，减轻劳动强度，避免重复的输入。

这种配置方式如下：

图片地址：https://static.oschina.net/uploads/img/201507/07165124\_P4Ln.png

**特别说明：**

必须有OID这一列，此列标记着要从那一个流程上copy数据。第一列仍然是超链接的一列，其他的列可以省掉，但是不作为copy数据的依据。

测试流程：导航测试-Copy历史数据发起流程。

按自定义参数

顾名思义就是自己写一个url, 配置到参数里，请参考ccBPM的WFStartGuide.aspx界面。

是否启用数据重置按钮？

**关键字：**ccBPM开始节点数据重置表单数据重置数据重置

图例：

图片地址：https://static.oschina.net/uploads/img/201506/21224340\_WBjJ.png

在一些应用模式中，用户输入的数据，需要重置一下，重置的目的就是把保持一个空表单，没有加载任何数据的表单。

如果启用了此功能，就会在开始节点里出现【数据重置】按钮。

图片地址：https://static.oschina.net/uploads/img/201506/21224404\_oW83.png

点数据重置后，数据就为设置为空。

图片地址：https://static.oschina.net/uploads/img/201506/21224424\_exCp.png

方便用户输入，努力提高用户使用体验。

是否启用自动装载上一笔数据？

**关键字：**ccBPM 自动加载上一笔数据发起节点自动装载上一笔数据

**应用场景：**在发起流程时，大量的数据采集都在开始节点填写，但是这些填写几乎是重复的，或者每次的填写，

为了避免重复输入，

**设置位置：**

图片地址：https://static.oschina.net/uploads/img/201506/21224538\_XKVR.png

同样为了减少数据重复的输入，流程的发起人想在发起流程前把上一次的流程数据复制过来，然后在此基础上做修改。

此功能，只有在不启用导航的模式下才可以使用。

当用户发起一个流程时，ccBPM就会自动的把主表、从表、附加等等数据加载到该表单上。从而减轻了用户的重复劳动。方便用户输入，努力提高用户使用体验。

延续流程

延续流程

**定义：**

为了完成一个大的项目、处理一笔大业务往往需要数年才能完成跨度历史周期比较长。节点非常之多、过程之复杂，但是他是一条线的流程，为了避免流程走的很累，根据业务的需要把整体的业务化作一个个的里程碑，每个里程碑叫做一个流程，一个个的里程碑串接起来就是一个整体流程。一个流程就是上一个流程的延续，我们就称呼为延续流程。

图片地址：https://static.oschina.net/uploads/img/201507/07165352\_6gJQ.png

**设置方法：**略。

流程数据与业务数据同步

概述

**应用背景：**一个业务数据，可能要对应一个或者多条流程，这个业务主体信息存储在用户指定的一张表里，对该业务主题发起流程前后，或者在流程指定的节点需要对该业务主体进行数据同步，我们把这样的行为称为流程数据与业务数据同步。数据同步是一对一的关系。一条流程对应一个业务主题的同步。

**应用模型1：**

以新闻公告管理为研究对象，新闻公告是业务主题，对于他要执行编写、审核、发布、撤销等操作。

如果引入了流程管理，就需要有新闻稿件发布审核流程，我们提倡把业务表与流程数据表设置成一个，这种方式就避免了，数据同步的问题，业务表就是数据表，数据表就是业务表，工作ID，就是新闻主键ID。

总结：这种方式适合业务逻辑相对简单，对主键的编号设置无明确的要求。

**应用模型2：**

以固定资产为研究对象，固定资产就是业务主体，对于一个业务主题需要执行，固定资产采购、维修、调拨、报废等等流程。在没有流程系统以前，固定资产也许就是一个对该表的增加、删除、修改的一个操作，没有引入流程管理。如果引入了流程那么我们对固定资产的管理就充入了流程管理思想的灵魂。

固定资产的采购流程，就是对应原来系统的固定资产增加的操作，一条流程走完后，就需要对固定资产表，增加一条数据。

固定资产的维修流程，就是对应了原来系统的固定资产维护功能，就要对这个固定资产进行一次维护的操作。

固定资产报废流程，就是对应了原来固定资产的状态改变，这就要求改变状态。

总结：对于这样的应用模型，业务逻辑相对复杂，对主键的编号有严格的规范，我们建议业务表与流程数据表单独分开存放，使用流程事件或者流程数据与业务表数据同步规则来处理，两者之间的信息同步。

**综述：**

以上种种模式都需要流程数据，与业务数据同步，当然ccBPM的丰富的事件功能可以满足该需求，为了更友好方便的设置我们定义了如下规则，让流程数据与您的业务数据方便同步。

**设置方式：**

图片地址：https://static.oschina.net/uploads/img/201506/21225007\_F5U9.png

同步方式

设置的同步方式有三种，如下图：

图片地址：https://static.oschina.net/uploads/img/201506/24114922\_dQWb.jpg

每种方式应用的背景不同，下面介绍每种同步的方式。

**不同步：**

顾名思义就不是不同步流程数据到其他业务表中，如果设置不同步，以下本栏目所有的设置都无效。

**按照业务表指定的WorkID计算如下图：**

这种方式适合于对方的业务表，扩充一个列叫WorkID，根据WorkID系统根据WorkID来同步信息数据。如果指定了这一项，需要在 [参数设置] 里设置要同步的业务表名。

**按照业务表指定的字段计算：**

这种方式适合于对方的业务表，扩充一个列叫WorkID，根据WorkID系统根据WorkID来同步信息数据。

执行同步时间点

图片地址：https://static.oschina.net/uploads/img/201506/24115013\_b195.jpg

**参数配置：**

**A.同步方式：**如果您需要此项功能，那么同步方式默认的---不同步，就需要修改。选择自己需要的同步方式。

**B.业务表名：**例如上图中的”YGRZ”，是您希望流程产生的数据另存在哪一个表中，以方便客户后期的使用；是数据库中真实存在的表，此表需要客户自己手动建立。

**C.业务表主键：**例如”OID”是业务表”YGRZ”的主键。

**D.执行同步时间点：**可以根据需要配置，如果您认为流程运行完第一个节点产生的数据就可以满足你的业务分析要求，那么您可以按上图示配置---流程编号+节点编号。

**E.流程数据存储表：**此项可以保持默认，一般为系统默认的，比如ND134Rpt。当然您也可以修改它，如上图示的MyFlowData。

完成上图配置后，点击上方的---保存按钮，此时ccBPM会对您所填的数据进行一个基本的检查以判断是否满足要求。

如果通过检查，单击---业务表字段同步配置项。选择您需要的流程字段和业务表字段的相互匹配关系。

在此需要注意：不可以为一列设置多个匹配项，就是说业务表中的一个字段不可以既对应流程数据表中的A字段，又对应B字段。

如果您的配置满足了要求，就像本例所示的那样(节点1运行完成后产生数据)，开始运行流程，数据就开始自动导入了

存储主表

**关键字：**流程数据存储主表

**说明**：默认为空，对应字段WF\_Flow.PTable。

程在创建时，ccBPM会创建一个流程主表，默认为ND+流程编号转化成int类型+Rpt. 比如一个流程编号为001 它的流程主表为ND1Rpt ，编号为010 主表为ND10Rpt.这个主表是系统自动创建的，这个主表记录了流程的基础信息。

如果为改变这个主表的名称在创建此流程时，可以自定义此它。

该表的字段由业务字段与流程字段组成，流程字段是用来记录流程在运行过程中的字段，一共有如下：

流程在运行过程的数据都存储在该表中，对流程的统计、分析、查询都是基于该表，此表的意义重大，它是与您业务信息关联的纽带或者您的业务主表就可以使用此表。

|  |  |  |
| --- | --- | --- |
| **字段名** | **类型** | **说明** |
| OID | Int | 工作实例ID |
| FID | Int | 流程ID |
| Title | string | 标题 |
| FlowEmps | string | 参与人员格式为@zhangsan,张三@lisi,李四 |
| FK\_NY | string | 发起年月，格式为 yyyy-MM ，方便按照月份统计分析。 |
| FlowStarter | string | 发起人编号 |
| FlowStartRDT | string | 发起日期 |
| FK\_Dept | string | 发起人部门编号 |
| WFState | 枚举 | 流程状态(详细状态)请参考： 在ccflow中有两个流程状态 WFSta与WFState有什么区别与 联系? http://bbs.ccflow.org/ showtopic-3031.aspx |
| WFSta | 枚举 | 流程状态(概要状态) |
| FlowEnder | string | 结束人 |
| FlowEnderRDT | string | 最后活动日期 |
| FlowDaySpan | Int | 跨度 |
| FlowEndNode | Int | 结束节点 |
| PWorkID | Int | 父流程WorkID |
| PNodeID | Int | 父流程节点发送 |
| PFlowNo | string | 父流程编号 |
| PEmp | string | 调用父流程的工作人员 |
| GuestNo | string | 客户编号 |
| GuestName | string | 客户名称 |
| BillNo | string | 单据编号 |
| FlowNote | string | 流程备注 |
| CWorkID | Int | 延续流程ID |
| CFlowNo | string | 延续流程编号 |
| AtPara | string | 参数 |
| PrjNo | string | 项目编号 |
| PrjName | string | 项目名称 |
| GUID | string | GUID |
| MyNum | Int | 数量 |

节点属性

基本属性

基本属性

**节点属性与流程属性，是控制流程运行的重要部分:**

流程属性是控制整体流程的属性设置，节点属性是控制特定节点的属性设置。

一条流程中有多个节点，只有一个流程属性，每个节点都有一个节点属性。

图片地址：https://static.oschina.net/uploads/img/201507/07174533\_O4Pt.png

目前节点属性分为四大部分区域，分别是基本属性、分合流、自动跳转、功能按钮标签、考核属性、审核组件、父子流程、公文按钮设置、移动设置。

**节点的基本属性是用来控制节点的基础信息的属性集合，如下图：**

图片地址：https://static.oschina.net/uploads/img/201506/21230910\_W7Zm.png

每个属性功能与应用场景都在下面的章节讲述

节点编号

**请参考流程编号章节: 3.2.1.1 章节.**

**简述：**流程编号规则是xxx, 流程编号是三位数的序号，从001到999，节点编号是流程编号转化成int类型加两位的序号。

**比如：**一个流程编号为001它的节点编号就是101,102,103 到199。如果是050那它的节点编号是5001，5002，5003 到5099。

步骤（无计算意义）

**步骤：** 它是一个int类型的数值，用于节点显示排序，在流程运转过程中没有实际的计算意义。主要用于在没有前后时间关联的节点情况下，按照步骤的排序。

节点名称

**说明：**每个节点都有一个名称，在一个流程中，节点名称可以重复。

**建议:**起名称时如果有岗位处理加上岗位名称。比如：分局长审批、秘书处处长审批、大厅受理、人力资源部经理审批。节点名称是在流程设计器中定义，执行保存就可以。

操作提示

**用于提示**：当前节点需要做的工作、应该做的工作或者工作的内容等。

**就是一个提示作用，具体怎么提示，可以根据需求填写。**

图例：

图片地址：https://static.oschina.net/uploads/img/201506/23135851\_2XVe.jpg

**显示效果：**

图片地址：https://static.oschina.net/uploads/img/201506/23135905\_4qzp.png

是否允许分配工作?

**关键字：**是否允许分配工作

**应用场景：** 节点A运动到节点B，如果B上有多个人可以处理B工作，A节点上的发送人，是否可以指定接受人的范围，这个指定接受人范围的过程也叫工作分配，如果允许分配工作，分配工作的连接就可见，否则就不可见。

相关功能：投递路径自动记忆功能。

是否启用投递路径自动记忆功能？

**关键字**：投递路径自动记忆

**相关功能**：是否允许分配工作?

节点A运动到节点B，如果B上有多个人可以处理B工作，A节点上的发送人，是否可以指定接受人的范围，这个指定接受人范围的过程也叫工作分配，如果允许分配工作，分配工作的连接就可见，否则就不可见。

如果允许分配工作了，对本次的分配是否自动记忆？让下一次该人，该节点上发送是否按照本次分配的计算？免去了多次分配工作的麻烦。

**相关功能**：投递路径自动记忆功能

**应用场景**：

在一个单位里，一个处的处长x，需要发送给副厅长审批。但是在这个单位里副厅长有多个，但是他们分管业务处室不同。发送后他们需要指定一个副厅长审批，也就是说要分配一次工作，每次发送，都要指定一次就会给用户带来麻烦。

启用此功能后，ccBPM就会按照最近指定的一次进行计算，自动的记忆了投递路径，免去的多次分配的麻烦。

该功能只有在启用允许分配工作才有效。

**技术信息：**

该自动记忆信息存储到了WF\_RememberMe这个表

访问规则

访问规则

**关键字**：ccBPM 访问规则

**相关功能**：访问规则处理内容。

**节点属性配置**:如下图

图片地址：https://static.oschina.net/uploads/img/201507/07175149\_69ps.png

**解释说明**：就是下一步工作人员的接受人范围处理规则。A运动到B，如何确定B的处理人范围。根据不同的业务场景，ccBPM提供了如下几种模式，您可以根据自动不同的业务背景设置自己的业务规则。

**说明**:

1, 下列设置类型，都设置当前节点作用于下一步节点。

2, 每一种类型，都有路径自动记忆功能，所说自动记忆功能是当节点第一次向下一个节点投递时，它把要投递的人记录下来。

如果您执行了分配系统就把分配的人员，做为接受人员计算.

**可以设置的投递的类型**:

为了更好的说明该规则，cc为我们提供了一个流程测试案例，如下图

图片地址：https://static.oschina.net/uploads/img/201506/21231308\_RXBv.png

该案例详尽的设置了各个模式的方法，请打开相关的节点属性，对照节点的名称，运行该流程

各种规则详细说明

本章节详细的介绍了每种访问规则在不同场景下的应用，用户可以根据不同的情况使用不同的访问规则。

**01.按当前操作员所属组织结构逐级查找岗位**:

\*\*设置方法: \*\*   
 在下一个节点上的节点属性里，设置节点岗位。  
 这是默认的投递规则，他是在下一个节点设置岗位时按照岗位计算. 他的计算方式，首先按照当前操作员的部门范围计算。如果该操作员部门下没有这个工作岗位的人员，ccBPM就会把当前操作员的部门级次提高一个级别，在寻找，依次计算。理解了这个算法，您就不难理解为什么，本部分的业务，只能让本部门的经理审批了

.

\*\* 举例说明: \*\*

一个省机关下面有n个县，n个市，n个县. n个所. 一个所员受理人员的业务，只能让自己的所长审批，所长的业务只能投递到本区县的相关业务部分审批，而非其它区县业务部分审批。

这就是岗位的权限与部门权限的交叉形成的被投递的人员集合. 这就是ccBPM经常说的。

岗位：表示能做什么事情。

部门: 表示能做那里的事情。

岗位+部门: 表示一个操作员能做那里的那些事情。

**02.按节点绑定的部门计算**

设置方法: 在当前节点上的节点属性里,设置节点岗位.  
  
cc会按照您指定的部门下面的人员，进行投递, 就是这个n个部门下面都可以接受这个工作. 这个类于发送邮件的按照邮件组进行发送。

**03.按设置的SQL获取接受人计算:**

按SQL计算通俗好理解，就是cc在执行一个查询sql时，返回一个数据源，在数据源里约定该节点的接收人信息。

设置方法: 在当前节点属性里 [接受人SQL]设置一个sql 语句. 这个select 查询语句有一个列. No 分别表示，操作

编号, 操作员名称. 这个sql可以有参数.

比如:

1, SELECT No,Name FROM PORT\_EMP WHERE FK\_Dept=@WebUser.FK\_Dept

查询出来当前操作员中的部门下的所有人员.

2, SELECT xxx as No, yyy as Name FROM dbo.xxxx.YourTable WHERE 字段名称=@表单字段名称.

从您的业务系统中，查找一组人员，变量可以是当前节点字段的编号，格式为 @+字段英文名称.

**按指定的人员:**

设置方法: 在当前节点属性的相关功能里 [接受人员]功能, 设置一批操作人员,当前节点向下一步发送时系统就会按照当前设置的范围投递。

这种场合应用于，特定的条件下, 一些特殊的临时的岗位不系统管理员不愿意去设置，而是直接选择投递到具体的人员.

**关于合流点的接受人按sql获取接受的表达式的问题**

注意子线程向合流点发送时，接受人规则的表达式的变量是临近合流点的子线程节点变量。

比如: 流程编号为ABC三个节点.

A 是分流点， C是合流点 C是子线程。

如果C的接受人员规则是按sql计算:

配置的表达式如下表达式是错误的:

select UserNo as No, xx as Name from ND2701 WHERE OID=@OID

如下表达式才是正确的:

select UserNo as No from ND2701 WHERE OID=@FID

这是因为子线程在发送时获取的变量OID 是子线程的ID而非，干流上的WorkID.

**关于子线程接受人的特殊约定：**

如果遇到分组的维度,就约定返回4个列来解决问题，流程demo:\流程树\同表单分合流\一人多子线程模式(批次维度任务模式)流程.

在第2个子线程节点配置了如下SQL。

图片地址：https://static.oschina.net/uploads/img/201506/21231659\_Sbx3.png

该数据源返回了三个列，分别是：No,Name,BatchNo。

No=操作员编号，Name=操作员名称，BatchNo批次编号。

对动态表单树的支持:

什么是动态表单树？请参节点属性、表单、表单类型章节。简单的说，该节点的表单是有上一步发送人员动态指定的。该节点大部分是子线程节点，也可以是多人处理的普通节点。

应用场景：a节点发向b节点，张三需要分配给，甲乙丙丁四个人去工作，但是这四个人工作内容不同。虽然甲乙丙丁四个人都可以接受到该节点的工作，但是填写的内容是由张三动态的分配的。

我们就要在这里约定数据源来表达接收人的信息，第一种情况没有批次号：返回的列需要有如下要求，No,Name,FrmIDs 第3列是表单ID，多个表单ID用逗号分开。

第二中情况具有批次号：需要返回的列是， No,Name,BatchNo,FrmIDs.

CcBPM为该种应用场景做了一个demo,请参考

图片地址：https://static.oschina.net/uploads/img/201506/21231758\_SIBW.png

图片地址：https://static.oschina.net/uploads/img/201506/21231809\_aAxn.png

**实现步骤：**

在开始节点里ccBPM的节点表单里设计了一个明细表。

**规则:04.按节点绑定的人员计算:**

节点绑定那些人员，该系统就会发送给这些人，如下图设置。

图片地址：https://static.oschina.net/uploads/img/201506/21231844\_EoIR.png

**规则:05.由上一节点发送人通过“人员选择器”选择接受人**

设置方式: 此此节点的上一个节点中启用【接受人】按钮，本步骤设置按接受人选择，上一个用户在发送前通过点【接受人】按钮选择指定的接受人（类于发送邮件时的接受人员选择）。

选择此项后系统就会自动在功能按钮栏显示一个选择接受人功能按钮，点击这里查看如何设置接收人选择器。

**技术细节：**

1, 当前节点的上一节点会有选择人的范围过滤问题，如果您在此节点上设置了岗位集合ccBPM在上一个节点弹出选择人的对话框时，就会按当前的岗位集合来过滤人员。

2, 在当前节点的上一个节点可能存在多个节点点分支。这个时间打开选择接受人对话框，就会出来首先要用户选择到那里去，然后在列出为这个节点的人员来。

**规则:06.按上一节点表单指定的字段值作为本步骤的接受人**:

设置方式: 在当前节点属性访问规则处理内容中指定此方式，在上一个节点的表单上添加一个SysSendEmps的文本框。

运行方式: 在用户填写上一个步骤的节点表单时，这个指定的字段可以用逗号分号分开，可以输入多个接受人员的编号。下一步的接受人员就按用户输入的内容结束。

说明：这种方式就类似于发送邮件。

**规则:07.与上一节点处理人员相同**

节点A是甲处理，发送到节点B,也是需要甲处理。

**规则:08.与开始节点处理人相同**

当前节点的处理人与开始节点一致，发起人是zhangsan,现在节点的处理人也是他。

**规则:09.与指定节点处理人相同**

应用场景1：A B C 三个节点, B向C发送时C的接受人员要求与A的工作人员一致。

设置方式: 在[访问规则处理内容]中设置一个节点ID比如:101。

应用场景2：如下图，当一个节点可以多个节点可以到达时，在【访问规则处理内容】需要配置多个节点的ID值，如下图：

图片地址：https://static.oschina.net/uploads/img/201506/21232056\_45OA.png

节点3，可能是从节点1，或者节点2转到节点3，如果在节点3上配置此规则就要配置节点2节点1的两个节点ID，用逗号分开，例如: 507,509。这种情况下ccBPM就会自动判断节点3究竟是从那个节点上过来了，从而把处理人投递给节点3。

对父子流程的支持：

2015年1月28日为珠海高凌变更：如果是父子流程，在子流程上的一个节点要指定与父流程的一个节点的人员相同，配置方式不变化。

比如：父亲流程甲，调用子流程乙，在乙的一个节点上的工作处理人员与甲的一个节点处理人员相同，那就在该参数里设置甲的节点编号，可以是多个变化，如果甲是一个子线程也同样支持。

**规则:10.按绑定的岗位与部门交集计算**

设置方式:在节点岗位，节点部门都设置。

运行方式：ccBPM会取既具备此岗位集合的又具备此部门集合的人员，做为本节点的接受人员。

规则:11.按绑定的岗位计算并且以绑定的部门集合为纬度

**规则:12.按指定节点的人员岗位计算**

应用场景：为一个单位设置一个设备维修流程，此单位下分好多部门，有一个IT部门负责计算机设备维修。每个部门的成员如果有设备维护的需要，首先填写一个单子向这个IT部门的受理人员发送详细的故障说明。IT受理人员接受到此请求后，根据情况发送到该发起人的部门领导那里去。

这是简单的三个步骤，发起-》IT部门受理-》发起的部门负责人审批。第一步骤基层人员发起，第二步骤是IT受理岗人员受理。第三个步骤中层领导审批。在第三个节点访问规则就是按按指定节点岗位计算。因为如果按岗位计算在第二步骤就要发送给IT部门经理审批而非发起人的部门经理审批了。默认的按岗位计算就是按上一个节点的岗位计算，现在的应用场景就是要按指定的节点岗位计算了。

设置方式:在接受对象中设置一个节点编号比如:101。

运行方式：ccBPM在处理接受人时，会按指定节点上的人员身份计算，而非按上一步骤的人员身份计算了。

其它：这种方式是对按岗位计算的补充。

**规则:13.按SQL确定子线程接受人与数据源**

此方法与分合流相关，只有当前节点是子线程才有意义。

**14.由上一节点的明细表来决定子线程的接受人**

同13.

**规则:15.仅按绑定的岗位计算**

按照节点上绑定的岗位来计算接受人，这里去掉了部门维度的过滤。

**规则:16.由FEE来决定.**

用流程事件，通过调用设置接受的接口，来设置当前节点的接收人，请参考：

图片地址：https://static.oschina.net/uploads/img/201506/21232348\_QVG6.png

这种方式通过编程，手工的在上一个节点设置该节点的接收人，实现的把接受人信息写入接收人列表里。

**规则：17.按ccBPM的BPM模式处理.**

这种模式下只有在BPM的模式下才能工作.

**BPM模式**：使用cc提供的GPM（权限管理）,CCFLOW（流程设计器）,CCOA（前台办公）集成的一整套的办公管理软件，我们称之为bpm或BPM模式。

在流程运转中，可通过此规则，按照组织结构（GPM权限管理中）查找处理人，包括部门领导、权限组角色等。

设置接受人选择器的人员范围

**接受人选择器样式：**

图片地址：https://static.oschina.net/uploads/img/201506/21232552\_1DWH.png

接收人的范围需要确定下来，可以让用户更精准的选择下一步接收人。本章节讲解了如何配置接收人选择器的范围，请参考如下：

如果当前节点的访问规则设置了由上一步发送人选择，那么就要定义[接受人选择器]的样式与可选择人员的范围内容。

如下图：

图片地址：https://static.oschina.net/uploads/img/201506/21232613\_Kaiv.png

控制人员选择器的样式方式有两种：表格与树。

控制人员范围的方式有如下几种，每种方式可以处理不同的场景。

按岗位：在绑定的岗位中选择接受人

按部门：在绑定的部门中选择接受人

按人员：根据绑定的人员选择接受人

按SQL：

在参数1里配置一个获取分组数据源的sql，它有No,Name两个列。

注意SQL参数支持ccflow表达式。

配置示例:

图片地址：https://static.oschina.net/uploads/img/201507/07175819\_BzZ1.png

显示示例：

图片地址：https://static.oschina.net/uploads/img/201507/07175827\_fWWR.png

自定义URL:

添加自己编写的人员选择器，并调用ccBPM的发送方法。

谁执行它

**关键字**：ccBPM 谁执行它

**解释**：谁能执行当前节点的工作？

对于节点工作的执行人，也就是说工作处理人，ccBPM 分为如下三种情况。操作员执行、机器执行、混合执行。

操作员执行：是节点的工作处理人员打开系统去处理工作，它是默认的方式。

**机器执行**：是在启动ccBPM的服务的模式下，由服务的定期的模拟工作处理人登录代替他执行，这种模式下工作的处理人员是看不到待办此节点上的待办工作的，当然就不能手工的处理此节点上的工作。

**混合执行**：是在启动ccBPM的服务的模式下，由服务的定期的模拟工作处理人登录代替他执行，这种模式下工作的处理人员是可以看到此节点上的待办并且能够处理他。

**补充说明**：

如果一个节点是需要定期的获取外部系统的数据，或者等待一个条件满足时才可以向下运动，这种应用场景就需要与节点事件组合使用来设置。比如：节点B, 需要等待返回扣款成功，才能向下运动，在节点B的发送前事件里写一个脚本如果扣款不成功就抛出异常，阻止节点B的向下运动。如果设置成让机器执行那么他就会间隔一段时间就去模拟当前操作员登录并发送此节点。

启动流程服务很简单，只要打开保持它是运行的就可以

发送后转向

**关键字**：发送后转向，发送后提示，发送后个性化提示信息

在节点成功发送后有如下处理方式:

**0 提示ccBPM默认信息**

系统提示详细的发送信息，包括下一步的接受人。

图片地址：https://static.oschina.net/uploads/img/201506/21233042\_LiVW.png

**1 提示指定信息**

按照您定义的信息提示给用户。此选项需要在[转向处理内容]文本框里输入提示内容。

图片地址：https://static.oschina.net/uploads/img/201506/21233108\_P4RS.png

提示信息可以支持变量，除了当前节点的字段变量以外系统还支持，发送中运行的变量，这些变量是ccBPM指定的变量列表。

上图的设置就标识了发送后的提示信息包含变量，发送后提示信息如下。

图片地址：https://static.oschina.net/uploads/img/201506/21233130\_E4Br.png

截止2012年12月18日的变量列表，不排除以后要增加：

|  |  |
| --- | --- |
| **变量名** | **中文名** |
| VarCurrNodeID | 当前节点ID |
| VarCurrNodeName | 当前节点名称 |
| VarToNodeID | 到达节点ID |
| VarToNodeName | 到达节点名称 |
| VarAcceptersName | 接受人集合的名称(用逗号分开) |
| VarAcceptersID | 接受人集合的ID(用逗号分开) |
| VarAcceptersNID | 接受人集合的ID&Name(用逗号分开) |

**2 转向指定的URL**

按照您定义的url转向，此选项需要在[转向处理内容]文本框里输入url。

**3 按照条件转向**

在相关功能里定义转向条件。

**其它：**

利用此属性您可以个性化提示信息。可以在流程处理完成后转向指定的url,也可以根据表单的字段的值转向指定的url。

比如：父子流程，当子流程启动起来之后，就可以利用此项功能转到父流程节点上去。

在子流程完成后，从子流程转向父流程配置案例：打开子流程的最后一个节点属性界面，设置成功发送后转向方式，为3，既：转向指定的url

1，  在转向处理内容输入一个url

MyFlow.aspx?WorkID=@FID&FK*Flow=021&FK*Node=2116，当这个节点完成后，它就会转向这个url。

启动一个新流程的事例：

MyFlow.aspx?FK*Flow=021&FK*Node=2101 ，当前节点上所有的数据将会根据同名字段，进行copy过去，包括从表、附件信息。

经过如上的配置，子流程在完成后就转向父流程的节点了。

已读回执

**关键字**：工作流引擎已读回执CCBPM已读回执

**工作流引擎已读回执定义**：

在工作流程引擎中：节点A 向节点B运动后，A的处理人想知道B是否打开此工作。

我们处理工作的时间周期，察看工作- 打开工作- 处理工作- 发送工作。

在打开工作时触发工作已读回执动作，在发送工作时处理节点收听动作，关于什么是节点收听请参考驰骋工作流引擎操作手册的其他章节。

**粗体**扩展阅读 - 电子邮件中已读回执:

如果您在使用电邮写信时选择了“已读回执”，当收件人打开您的邮件时，会提示发送方要求发送已读回执，如果对方同意发送回执，那么就会有一个已读回执发送回来给你，您就可以确认对方是否收到了该邮件，并且知道对方是什么时候阅读该邮件的。如果对方拒绝发送回执，那样您就无法知道接收方是否已读。

但是CCBPM没有处理拒绝发送已读回执动作。

**应用背景**:

在财务报销流程中，申请者想在第一时间里知道他的上级领导是否打开他申请单。CcBPM的抽象模式:

不设置，自动回执，由上一节点的表单字段决定，由sdk开发者参数决定四种ccBPM已读模式。

不设置：默认为不设置。

由上一节点的表单字段决定：比如A发向B ，在B节点上设置此规则，要在A的节点表单上增加一个系统字段SysIsReadReceipts建议为boolen类型，0表示不回执，1表示需要阅读回执。

由sdk开发者参数决定：如果您采用sdk模式作开发，可以用此种方式作参数告诉CCBPM需要阅读回执。比如：A 发向 B，A节点发送的sdk 参数传入SysIsReadReceipts值为0 或者1 ，表示是否需要已读回执。

**设置与应用**：

设置ccBPM的回执规则路径 =〉节点属性 -基本配置

图片地址：https://static.oschina.net/uploads/img/201507/07181155\_TJbc.png

**下面介绍测试用例：如下图**

图片地址：https://static.oschina.net/uploads/img/201506/21233440\_DL40.png

**设计说明**：

1，在第二步的部门经理审批节点的节点属性里设置了，自动回执。也就是说，申请人填写后，发送给部门经理审批，部门经理打开待办工作后，CCBPM会自动地向申请人发一个消息，告诉申请人已经阅读了他的工作。

2，在部门经理审批表单里增加了一个boolen字段，是否需要已读回执，在人力资源部归档节点设置了“由上一节点的表单字段决定”。

运行的效果与预期一致。

单元测试：

参考D:ccflowtrunkDemoAndTestingBP.TestCaseBaseFuncReadReceipts.cs

方向条件控制规则

**关键字**： ccBPM 方向条件控制规则，方向条件控制方式。

**解释**：一个节点向下运动遇到分支后，需要有方向条件规则来控制，这个规则有如下两种。

**由连接线的条件来控制**：

在连接线上设置条件转向，当满足条件时，就走这一条路线，我们以请假流程为例来说明

图片地址：https://static.oschina.net/uploads/img/201507/07181411\_BM6H.png

方向条件有很多种类型，可以根据不同的情况设置方向条件。

**让用户手工选择**：

用户发送后，就转到一个选择到达节点界面，选择到达的节点。让用户来决定，将流程走向那里。

请参考demo流程：

图片地址：https://static.oschina.net/uploads/img/201506/21233621\_XTFe.png

在第一个节点用户保存后，执行发送

图片地址：https://static.oschina.net/uploads/img/201506/21233633\_4gkc.png

**条件选择规则**：

到达普通的节点、分流节点、合流节点、同表单的子线程节点，只能选择一个。

到达异表单节点可以选择多个，但是不能在选择以上的单选节点。

如果到达的节点的接受人是由上一步发送人员选择，就需要出现选择接受人连接。

图片地址：https://static.oschina.net/uploads/img/201506/21233654\_tKTq.png

这个测试案例涵盖了所有的由手工转向的方式。

生命周期

撤销规则

**关键字**：ccbpm撤销规则

**应用背景**：一个操作员在发送之后，处理完当前任何一个节点之后，发现当前他的工作需要重做，这里就需要用到了撤销。

图片地址：https://static.oschina.net/uploads/img/201506/21233812\_0xX7.png

一个节点可以撤销那些节点的工作，是有这个撤销规则控制的。

**上一步可以撤销**：顾名思义就是上一步骤的发送人，比如： E 发送到F，E点可以撤销自己的发送。

**不能撤销**：该节点上不能被撤销。

**上一步与开始节点可以撤销**：A,B,C,D,E,F节点， E发送给F，待办停留在F上，只有A与E点的处理人员可以撤销发送。

**指定的节点可以撤销**：此功能需要配合相关功能的，指定撤销节点才能有效。

**选择指定的节点可以撤销**，点击保存，之后设置可以撤销的节点。

图片地址：https://static.oschina.net/uploads/img/201506/21233847\_mnKt.png

**其他**：

撤销功能的操作，在未完成，或者叫在途的工作列表里，可以执行此功能，手机端也是如此。

图片地址：https://static.oschina.net/uploads/img/201506/21233912\_q3eQ.png

点击撤销后，如果撤销成功，该工作就进入了当前人的待办。

节点工作批处理

**关键字**： ccbpm节点工作批处理

**应用背景**：有一些节点上的操作是需要批量的处理，常见的就是批量审核。我们都知道，ccBPM没有审核节点的概念，任何节点都是数据采集采集。对于审核也是节点数据采集。

批量处理就是一个节点上的多个待办一次性填写（完成数据采集），一次性发送。

图片地址：https://static.oschina.net/uploads/img/201507/07181710\_z40f.png

**批处理参数**：就是批量采集该节点的字段。 如果参数配置一个URL而不是字段, cc将会认为要让这个url来处理批量发送。

其他：Demo流程

图片地址：https://static.oschina.net/uploads/img/201506/21234039\_qyvC.png

**前台执行效果**

图片地址：https://static.oschina.net/uploads/img/201506/21234106\_3V5Y.png

图片地址：https://static.oschina.net/uploads/img/201506/21234113\_B9h7.png

点击批量处理的按钮，就可以执行已经选择的需要批量处理的流程。

表单

节点表单方案

概述

**关键字**： ccBPM 表单URL. 表单类型, 表单解决方案。

**相关功能**：表单URL, 节点表单ID.

目前为止 CCBPM流程引擎设计器支持如下几种表单类型.

傻瓜表单 自由表单 自定义表单 SDK表单 树形表单禁用(对多表单流程有效)

流程设计人员可以根据客户的需求选择对应的表单类型。

图片地址：https://static.oschina.net/uploads/img/201506/21234625\_cgK5.png

本章节对应的demo详细的介绍了该章节信息

图片地址：https://static.oschina.net/uploads/img/201506/21234644\_JPZJ.png

傻瓜表单

由n列与字段分组构成,它的格式固定一列说明，一列控件，一列说明一列控件。配以字段分组可以解决字段排列的问题。

详细请参考: D:ccflowdoc图片介绍零代码.傻瓜设计器表单

图片地址：https://static.oschina.net/uploads/img/201506/21234720\_4y7W.png

图片地址：https://static.oschina.net/uploads/img/201506/21234736\_K6Jf.png

对应的测试Demo: \流程树\线性流程\表单方案\傻瓜表单

傻瓜表单简洁大方、设计速度、运算速度都很好，但是风格僵硬统一

自由表单

可以自己按照自己的布局生成的表单，这种表单格式灵活,控件丰富.

详细请参考: D:ccflowdoc图片介绍零代码.自由表单设计器

图片地址：https://static.oschina.net/uploads/img/201506/21234817\_SfYE.png

测试demo： \流程树\线性流程\表单方案\多种请假流程.

图片地址：https://static.oschina.net/uploads/img/201506/21234829\_cfBn.png

启动流程后的表单如下.

图片地址：https://static.oschina.net/uploads/img/201506/21234842\_2cl7.png

Tab标签页表单

在流程设置为傻瓜或者自由表单时，如果该节点绑定了其他的流程表单，系统就会按照标签页的方式显示出来

图片地址：https://static.oschina.net/uploads/img/201506/21234908\_1UbK.png

如果表单的数据发生变化后，就在标签头部显示星号，如果切换标签页，ccform会自动保存。

如何控制一个表单在不同的节点的权限，请参考下表单库章节。

自定义表单

**定义概述**：一个已经做好的表单需要绑定到节点上。

**自定义表单工作模式**：流程控制按钮区域是ccBPM来完成，表单区域是放在控制区域下面的框架里，如下图所示。

图片地址：https://static.oschina.net/uploads/img/201506/21234943\_4BRS.png

注意:蓝色边框一下是一个框架，框架里面的界面是一个url，这个就是自定义表单。

**应用背景**：这种类型是解决我自己已经有了一个成熟稳定的产品或者项目，里面的表单已经做好了，我只想使用流程引擎负责流转功能，业务数据的存储不变化。这中类型有如下特点。

A,数据的存取有自己开发人员来确定。

B,仅仅使用ccBPM的流程引擎。

C,最大限度的减少程序开发人员的开发。

D,最大完整的保障自己的业务独立与业务完整。

如果要适应这种方式需要在把自己的表单做如下改造：

A,把自己的表单主表增加一个int类型的workid列用于做主键或者用程序保障他的唯一性。

B,自己开发的表单，要接受来自ccBPM的参数，参数格式如下。比如：您输入的是 http://localhost/MySelf.aspx?aa=xxx 系统在调用时就会是http://localhost/MySelf.aspx?aa=xxx&WorkID=123&FK\_Node=101&UserNo=zhoupeng&SID=xxxxx，这些参数是系统自定传递的，它表示那个流程实例、到达了那个节点、那个流程，与当前的登录人员，SID 是验证该人员是否合法，请参考其他章节。开发人员可以根据节点编号灵活控制权限。

如果您了解了ccBPM的数据存储结构（请参考流程属性章节），我们建议您把该流程的流程属性，数据存储表名与您的业务表单表名一致。

图片地址：https://static.oschina.net/uploads/img/201506/21235019\_6rer.png

**这样做的好处有**：

1,不用担心数据存储主键，不用关心流程的业务逻辑。

2,系统会给该业务字段自动创建其他的流程字段，用户可以根据这些字段信息扩展更多的业务逻辑编写。

**自定义表单流程demo:**

流程编号：129

图片地址：https://static.oschina.net/uploads/img/201506/21235046\_GweU.png

第一步：开发一个自定义表单位置在

图片地址：https://static.oschina.net/uploads/img/201506/21235101\_S179.png

在该表单里有一保存按钮，并让其隐藏，该按钮的功能是出来保存整体表单数据，如下图

图片地址：https://static.oschina.net/uploads/img/201506/21235112\_vMJo.png

定义一个保存函数Save，这个保存函数是系统约定的，保存函数里激活保存按钮从而完成保存，调用Save保存是流程的发送与保存工具栏按钮，如下图就是表单运行时的样子。

图片地址：https://static.oschina.net/uploads/img/201506/21235126\_rDcc.png

第2步：把配置节点属性。把辨析的该表单定义如下，并保存，第2，3个节点同样的设置。

图片地址：https://static.oschina.net/uploads/img/201506/21235137\_xVGS.png

第3步：设置步骤2，3的审核组件属性，让其启用，并把高度与宽度设置为0，表示高度宽度为100%。

图片地址：https://static.oschina.net/uploads/img/201506/21235150\_Cyf0.png

第3步骤：启用调试运行。

节点1效果：

图片地址：https://static.oschina.net/uploads/img/201506/21235203\_tdGn.png

在用户点击保存与发送按钮后，系统就会自动触发了框架里面的save方法，完成框架内的数据保存，如果框架内的函数返回false,标识发送失败，系统就停止发送。

发送之后效果：

图片地址：https://static.oschina.net/uploads/img/201506/21235215\_E9mn.png

节点2效果：

图片地址：https://static.oschina.net/uploads/img/201506/21235229\_wgAM.png

节点3效果：

图片地址：https://static.oschina.net/uploads/img/201506/21235243\_1y3Q.png

该案例详细的说明了如何使用自定义表单完成对业务的快速实现，更多的信息请参考源代码与源代码里面的注释。

SDK表单

如果您采用应用dll的模式开发您的工作流程系统那就采用这种方式，这种方式的特点：

1，引用dll,调用开发接口。

2，自己编写表单格式，处理表单业务逻辑，并且自己托放发送，保存，退回，转发，删除按钮。

3，在每个流程功能按钮下填写业务逻辑外，还要编写调用流程接口方法。

比如:调用发送方法.

/\*

string msg = BP.WF.Dev2Interface.Node\_SendWork(this.FK\_Flow,this.WorkID,this.GenerWorkInfo());  
 msg = msg.Replace("@@", "@");  
 msg = msg.Replace("@", "<BR>@");  
 this.Alert("发送提示", msg);

\*/

事例路径:

D:ccflowtrunkCCFlowSDKFlowsQingJia

详细请参考ccBPM的sdk开发相关章节。

SL表单（测试版）

使用Silverlight 开发的表单，目前在测试过程中。

表单树

用树形的结构显示表单，详细请参考树形表单的权限控制

图片地址：https://static.oschina.net/uploads/img/201506/21235425\_aQDY.png

动态表单树

动态表单树表单树节点符合如下几个特征。

1， 节点表单是一个多表单节点。

2， 表单是以树型的方式展现的。

3， 表单树里的表单是可以动态根据不同的接收人变化而变化的。

禁用(对多表单流程有效)

也叫禁用节点表单, 就是不使用节点表单仅使用流程表单，这样的场景仅仅应用到Tab标签页表单。

如果选择了该项，那在标签页里就不会显示节点表单，仅仅显示您关联的流程表单。

图片地址：https://static.oschina.net/uploads/img/201507/07184131\_KfHw.png

焦点字段

**焦点字段的定义**:在一个表单中有一个特殊的字段，并且是一个可以输入(比如审批意见)，这个字段的采集信息可以反应到轨迹表中。

**应用场景**：如下流程图是一个典型的对附件的审批流程，起草人发起一个标准文件，经过各部门审批，意见征集（分合流），各部门汇签（分合流）。

图片地址：https://static.oschina.net/uploads/img/201506/21235557\_aIVm.png

在工作日志中，用户希望看到日志记录每次审批意见，如果有多次退回，多次审批意见，个是什么。如下图：

图片地址：https://static.oschina.net/uploads/img/201506/21235614\_6XMD.png

如何让ccBPM要知道记录表单的那个字段，这个字段可能是审批意见也可能是备注。如果您不设置它，ccBPM就记录不上。

**解释说明**：

节点属性中的焦点字段是用来处理记录一个表单中重要的字段，比如：一个审批表单中的审批处理意见字段，对这个表单来说他非常重要，在流程中我要记录这个字段作为此节点的审核信息。

并非所有的节点都要设置焦点字段。

如下功能中用到焦点字段，工作日志、转发、退回。

**配置方法**：此配置支持ccBPM的表达式，比如：部门审批意见@BM\_Note@, 什么是ccBPM表达式请参考相关章节。

**在工作日志中应用**：

如果您为此节点设置了焦点字段，系统在日志记录时，就会记录到日志中去。如果没有则不记录，如果对一个审批流程来说，记录每个节点的审批意见记录工作日志非常中要。

**在转发中应用**：

如果设置了焦点字段，用户就可以在当前表单上在这个字段上填写意见，按下转发按钮后这个字段的数据就会被带到转发功能界面里去，符合操作习惯。

**在退回中应用**：

如果您设置了审批意见为焦点字段，如果用户按下退回按钮就会认为审批意见填写的信息为退回信息。

**在流程检查中体现**：

您在节点属性中设置了这个焦点字段A，由于当前节点的表单变更丢失了字段A，那么就会检查错误。

节点表单ID

**定义**：在一条流程中，如果一个节点的表单信息与该流程中的表单内容是一样的，为了避免重复设计，可以指定该节点的表单ID，为指定的节点表单ID。

流程demo: \流程树\线性流程\自由表单-审核组件

图片地址：https://static.oschina.net/uploads/img/201506/21235737\_emE2.png

该流程中，第3，4个节点的表单内容都与节点2相同，为了避免重复设计表单，该属性里做了如下设置

图片地址：https://static.oschina.net/uploads/img/201506/21235758\_ObLK.png

保存方式

关键字： ccBPM保存方式应用范围：ccBPM的节点表单

说明: 在单表单节点里，保存首先更新的节点表单数据表，但是如果需要更新流程数据表，就要选择该项

图片地址：https://static.oschina.net/uploads/img/201507/07184336\_WoXI.png

审核组件

概述

审核组件是为了方便流程设计人员进行开发的一个组件，在一个流程里，一般来说开始节点填写一个或者多个表单，在以后的节点里，都是审批，我们把这样的流程也称为审批方式流程，为了解决这样的流程工作模式，我们创建了审批组件。

图片地址：https://static.oschina.net/uploads/img/201506/21235919\_k4QC.png

下面是关于审批组件的一些特征描述：

1.一个节点里只有一个审批组件。

2.审核组件可以工作在sdk的组件里，关于什么是sdk组件请参考相关章节。

3.审核组件的数据存储到NDxxxTrack里，每一个审核都会写入一笔数据在里面。一个审核人员、一个节点、一个流程实例，只能写入一笔数据。

很多的设计人员，都没有看到审核组件的作用，也没有了解跳转规则，比这视频就自己开发表单，采取瀑布方式的设计，这样导致，第一个表单变化后，后续的表单都要变化？增加了开发难度，降低了实施的效率.

如何解决此问题？  
  
1， 多使用ccBPM的跳转功能。  
  
2， 使用ccform的审核组件。  
  
跳转功能能够解决，重名的节点问题，这样配合审核组件来完成。  
  
大多的应用场景如下：  
  
 1，开始节点填写一个申请单据.  
  
 2, 以后的节点都跟着审批，审批，直到流程结束.  
  
  
使用审核组件，就可以很好的解决此类的问题,开发步骤如下.  
  
1, 在开始节点设计表单.  
  
2, 把表单复制到第2个节点上去, 默认字段都是只读的.  
  
3, 在第2个表单上增加一个审核组件, 这个时候审核组件的状态是启用的.  
  
4, 打开第3个节点属性. 选择"表单"标签页, 填写第二个节点表单的ID值，比如:ND102 选择节点属性的 “审核组件”标签页 => 审核组件状态 设置为 启用。  
  
5, 第n个节点，类似的做法.   
  
总结:  
  
1, 审核组件能够让减轻开发工作量, 没有必要个个节点都要画表单.  
  
2, 每个节点的审核，都增加一条数据. 此数据存入了 NDxxxTruck 里面.   
  
3, 如果对数据轨迹要求比较严格，那就不能安心的自己一个个的表单画吧。

审核组件状态

审核组件有三种状态，禁用、启用、只读

图片地址：https://static.oschina.net/uploads/img/201506/22000045\_njrH.png

禁用：就是不体现在节点表单里，在节点表单里不工作。

启用：在节点表单里工作，用户可以填写审批意见。

只读：只能查看，不能在编辑审核意见。

显示方式

显示方式就是审核组件的工作方式。

自由模式：

表格模式：

**这里只支持表格模式。**

审核组件工作方式

**工作方式分类**：审核组件模式、日报模式、周报模式、月报模式。

**审核组件模式**：一个节点上一个人只能填写一条数据。这种模式主要是解决审核的问题，同时考虑到了多人审批一个工作那就是每个人都有一条记录。

**日报模式**：一个节点上，每个人，每天只能填写一条数据。应用场景：我们在为用户设计一个销售跟踪流程，当前点是一个客户跟踪节点，销售员需要在每天跟踪的情况记录下来，知道客户跟踪更有了结果在发送下去，否则始终停留在该节点上，这个就是日报模式。

**周报模式**：与日报模式相同。

**月报模式**：与日报模式相同。

**总结与应用**：审核组件用于一节点一个人一个记录模式，其他的属于一个节点一个人根据时间来控制多记录的模式，用于停留一个节点上，不断的根据时间要求采集外部信息。销售跟踪流程就是一个典型的例子，销售人员要跟踪一个客户，需要经常的定期的拜访，每次拜访的结果，作为一条记录写入到这个组里。

运行模式

概述

此部分阐述了，节点运行的模式，分流合流，分合流的设置

图片地址：https://static.oschina.net/uploads/img/201506/22000253\_vDeZ.png

节点模式

节点在流程中运行的模式：节点模式分为普通，分流，合流，分合流、子线程。

分流点：它发送时n个操作人员都可以接受。

合流点：n个操作人员把采集的数据发送到此节点上。

分合流：既是分流也是合流，我们称为分合流。

下图是节点模式图例。

图片地址：https://static.oschina.net/uploads/img/201506/22000324\_7jvD.png

如何该表一个节点的节点类型，请参考，需要在如下节点属性上设置

图片地址：https://static.oschina.net/uploads/img/201507/08090330\_jnq0.png

子线程删除模式

**关键字**： ccBPM 子线程删除模式

**应用背景**：子线程向合流节点运动时，如果达到了完成率，合流节点上的处理人就可以处理该合流节点的工作了。CcBPM的规则是，如果合流节点向下运行，必须所有的子线程完成后才能执行下一步骤，如果在子线程没有处理完的情况下向下发送，系统就会提示，“有一些子线程没有完成，您不能向下发送。”的提示。但是如果完成通过率不到100%，就会造成合流节点有了待办以后，仍然有子线程没有处理，他要向下运动，必须把未完成的子线程删除掉才可以。该模式规定了如何删除子线程。

删除规则如下：

**不能删除**：必须等到所有的子线程运行完成后，该合流节点才能向下运动。

**手工删除**：在合流节点上打开子线程查看器，手工的删除未完成的子线程才能向下运动，必须在合流节点上启动子线程按钮，才能执行此操作

图片地址：https://static.oschina.net/uploads/img/201506/22000527\_crA2.png

点击子线程后，就显示

图片地址：https://static.oschina.net/uploads/img/201506/22000538\_FmhG.png

如果设置了手工删除子线程，就会呈现如下界面：

图片地址：https://static.oschina.net/uploads/img/201506/22000549\_XacA.png

**自动删除**：在想下发送时，ccBPM就会自动的把未完成的子线程删除掉，然后向下运行

是否允许子线程接受人员重复

图片地址：https://static.oschina.net/uploads/img/201507/08090736\_Fe3M.png

**允许：**就代表分流点向子线程发送时，接受人允许重复。

**不允许：**子线程节点处理人不允许重复。

子线程类型

**关键字**：ccBPM 子线程子线程类型

**相关功能**：分合流，分流，合流

**说明**：首先该节点是子线程节点才有效，我们把分合流，分为同表单的分合流，与异表单的分合流。在流程运行时，ccBPM需要设计人员指定，该节点是同表单的分合流还是异表单的分合流。

相关demo流程：

**异表单分合流**

图片地址：https://static.oschina.net/uploads/img/201506/22000651\_f7TE.png

图片地址：https://static.oschina.net/uploads/img/201506/22000659\_rMGi.png

完成通过率

**概念：**完成率 = 子线程上已经完成的数据/所有子线程数量\*100%

**说明：**该节点对于合流节点与分合流节点有效，当子线程的完成率达到该值的时候，该节点的待办才能显示出来，否则该节点的人员不能处理待办。如果合流节点的处理人能够看到待办，他就可以对该流程进行操作，比如：发送、删除、退回、删除子线程等等。

子线程启动方式

子线程启动方式，分为两种：

**指定的字段启动**

**按明细表启动**

图片地址：https://static.oschina.net/uploads/img/201507/08091101\_lCzD.png

多人待办工作处理模式

多人待办工作处理模式，也是待办处理模式。是当接受的节点是多个人的时候，如何处理待办？

图片地址：https://static.oschina.net/uploads/img/201507/08091121\_CAdZ.png

根据不用的场景，ccBPM把多人在普通节点下的处理模式分为如下几种。

**抢办模式**:

A发送到B ，B节点上有n个人可以处理。这n个人都可以看到待办，当其中一个人处理后，其他人的待办就消失了。这样的工作模式属于抢办，这n个人可以同时打开，当一个人发送后，其他人都不能在发送了。通俗的说，也就是谁抢到了这件工作，就是谁处理的。抢办模式是一个默认的处理模式。

**协作模式**:

A发送到B ，B节点上有n个人可以处理。这个n个人都可以去处理，都可以去发送，有最后一个处理人发送到下一个节点上去，这n个人的处理不分顺序。这样的模式我们叫做协作模式。

协作模式通常用在具有审核组件的表单中，每个人都可以填写自己的意见，发表自己的看法，在审核组件里审核每个人在一个节点上只能写入一条数据，也就是说只能表达一次意见。

也经常用在传阅的情况下，必须这些人都读取了，知道了才能向下运动。它与抄送不同的地方是，抄送可以不看，就可以向下运行，但是他必须处理才能向下运行。

通常的协作模式，就是多人都要点击一下发送功能按钮，来确认自己处理了，确认自己同意了或者认可了。

**队列模式**:

A发送到B ，B节点上有n个人可以处理。这个n个人按照设置的先后顺序去处理，在同一个时间点必须有一个人待办，其他人看不到。

应用场景: 一个公文发送给一个部门，该部门下n个人可以接受（注意这是一个节点），按照国内的行政制度，这n个人会从小到达排一个队列，有最小的级别的工作人员审批，然后依次类推，走到最后由最后一个人发送到下一个节点去。这n个人都是该节点的处理人，这n个人的审核步骤是由他们的顺序确定的，所以在设置该节点的接受人时，要按照顺序执行。

通常这样的模式也与审核组件一起使用，每个人表达意见后，就发送到下一个节点，当然这样的模式也

**共享模式**:

A发送到B ，B节点上有n个人可以处理。ccBPM有一个任务池，这写人都需要从这个任务池里获取。

　　关于对ccBPM任务池的描述：

　　１，在流程属性与全局配置设置了启动任务池的状态下，如果一个节点遇到多个人处理，这个任务就放在了任务池里，只有参与处理的人人员才能看得到。

　　２，工作人员在处理该工作之前需要申请，放到自己的代办里，一人申请之后，其他人就不能申请了，多个人同时申请第一个人申请的先得到。

　　３，申请之后，任务池里就不出现在条工作而是进入了被申请人的待办里。

　　４，如果申请人处理不了这件工作，他可以把该工作放入任务池中，放入任务池的工作可以被其他的Ｎ个人可以看到。

　　５，不管取回与放入，都需要写入流程日志。

　　6，任务池的设计：

　　１，在流程引擎控制表 WF\_GenerWorkFlow 中增加了如下字段:

　　　　 TaskSta,　枚举类型， 0=不启用，1=共享，2=被申请。

　　　　 TodoEmpsNum, Int类型，处理人员数量，当前节点工作有多少人待办，如果此值大于等于２，讨论共享任务才有意义。

TodoEmps varchar(500) 待办人员,格式为　处理人员编号，处理人员名称；　比如：guobaogeng,郭宝庚;fuhui,福惠;

　　２，在流程引擎控制表的人员列表中　WF\_GenerWorkerList 中对于IsEnable字段,进行扩充了，如果等于　-1=任务池禁用，０＝分配工作被屏蔽，１＝正常。

　　３，增加了一个功能界面　/WF/TaskPoolSmaller.aspx 次功能界面是有在任务池启用才能被使用。

　　４，在流程属性里增加一个字段　IsEnableTaskPool 是否启用任务池？　此字段用来控制该流程是否启用任务池。

**任务池的ＳＤＫ接口**：

　　１，获得共享任务：　DB\_TaskPool

　　２，获得共享任务中我的待办：DB\_TaskPoolOfMy

　　３，从共享任务中取回：Node\_TaskPoolTakebackOne

　　４，把取回的共享任务放入任务池：　Node\_TaskPoolPutOne

**自动跳转规则**

自动跳转，就是遇到指定的节点

图片地址：https://static.oschina.net/uploads/img/201506/22000957\_pviS.png

处理人就是发起人：如果后面节点中的处理人是发起人，就进行跳转

处理人出现过：如果后面节点中的处理人出现过，就进行跳转。

处理人与上一步相同：如果后面节点中的处理人与上一步的处理人相同，就进行跳转。

找不到处理人规则：提示错误或者自动转到下一步。

发送阻塞模式

**定义**:一个节点在向下运动时候，在特定的条件下需要阻塞向下运动，这些条件满足了要阻塞向下运动。

**解释**:比如一个节点启动了多个子流程，该节点需要所有的子流程都完成才能向下运动。这就符合发送阻塞模式的第二种，如下图。

图片地址：https://static.oschina.net/uploads/img/201506/22001048\_eFnr.png

**规则说明**：

**不阻塞**：就是没有条件，它是默认的规则。

**当前节点的所有未完成的子流程**：当该节点启动一个或者多个，一种或者多种子流程时，需要等到所有的子流程都完成才能向下移动。

实例：在B节点上，我启动了甲流程三个条，乙流程4条。该节点要向下运动，必须等到所有启动甲的流程与所有启动乙的子流程都完成了，才能向下运动。

**按约定的格式阻塞未完成的子流程**：

当该节点向下运动时，要检查指定的历史节点曾经启动的指定的子流程全部完成，作为条件。

实例：在D节点上，要检查曾经在C节点上启动的甲子流程是否全部完成，如果完成就不阻塞。

配置格式：@指定的节点1=子流程编号1@指定的节点n=子流程编号n。

**按SQL阻塞**：

执行该SQL，返回大于0就阻塞，小于0不阻塞，在表达式里配置该sql,sql支持表达式。

配置格式： SELECT COUNT(\*) FROM XXXX WHERE ABC=@XXXX

**按照表达式阻塞**：

在表达式框里填写一个表达式，如果表达式成立就阻塞，不成立不阻塞。

配置格式：@参数空格运算符常量/变量值 , 比如: @JineE >= 1000 , @FaType = 1

**相关属性**：如下两个属性来与该模式一起工作，一下两个属性都支持cc的表达式。

**阻塞表达式**：根据选择的模式不同，设置该模式相关的表达式。

**被阻塞时信息提示**：如果被阻塞，需要提示什么样的友好信息給用户。

**事例演示**：\流程树\父子流程\经典流程\节点发送阻塞测试

图片地址：https://static.oschina.net/uploads/img/201506/22001246\_ZR8s.png

在第2个节点上，启动子流程，但是子流程没有结束，直接发送就抛出如下错误。

图片地址：https://static.oschina.net/uploads/img/201506/22001258\_8XSq.png

以上讲解了如何使用阻塞规则，来友好的、可配置的实现用户的需求。

跳转

**基本概念**：跳转在流程运行中经常用到，具有跳转的功能往往是与审核组件一起使用。跳转的意思就是当运行到当前节点时，是自动的让其跳过本节点，还是停留在本节点。

功能按钮

概述

按钮控制是在当前处理人可以执行的当前节点的那些操作，以及这些操作的规则方式是什么？也是节点工作处理的权限控制。

更直接来说就是工作处理器的那些部件在当前节点中可以使用，关于每个工作处理器的部件功能请参考，以上章节[工作处理器功能部件]介绍。

图片地址：https://static.oschina.net/uploads/img/201506/22225245\_3Dp1.png

读取功能按钮并工作的就是工作处理器，就是当前工作人员能够执行那些功能，如下图

图片地址：https://static.oschina.net/uploads/img/201506/22225322\_xAcd.png

工作处理器的头部按钮就是有此功能来控制的，读取节点属性的配置，不仅仅在工作处理器上，也在sdk开发模式的工具栏组件里，也是读取该配置。

**为什么要控制前台的标签按钮?**

我们知道按钮的标签文字都是一个名字，但是这个名字不一定适合用户的要求。

比如在请假流程有三个点，发起流程，部门经理审批，人事备案。

1, 在开始节点的 [发送] 按钮名词就比较专业化，如果修改成[发起请假流程]就比较贴切。

2, 在第二个节点 [退回] 如果修改成[退回给发起人] 就比较让人容易看懂。

3, 在人事备案节点上，如果不需要[退回]按钮，可以禁用它。

**特别说明**：以上设置对于采用嵌入方式开发有效，如果采用sdk模式开发，这些设置不起作用。有两个重要的权限控制，分配是退回规则与附件权限，请参考下章节。

发送按钮JS函数

发送按钮必须启用，所以就没有是否启用的配置。

在流程发送前，经常会遇到数据安全性、完成性检查的需求。那就需要在发送前调用一个js检查函数。这个js函数里设置检查逻辑。

图片地址：https://static.oschina.net/uploads/img/201506/22225529\_3DA0.png

做数据安全性检查，我们提倡用事件处理。

我们知道，表单提交前的安全性检查是很有必要的，很多的开发者首先要考虑的是javascript检查，就是在发送的按钮里写一个事件,用这个事件来执行检查用户的输入是否合法。

在这里ccBPM要提倡的这种检查放在事件里进行，有如下原因。

1，事件里容易表达业务逻辑。  
  
2，事件里支持多种表单的检查，目前ccBPM提供了aspx表单的解析执行器，正在完善着slilver的表单解析执行性，与正在开发的android表单解析执行器。所以你在编写js的脚本在这些设备下不能很好的应用。

基于以上两个原因，所以提倡你使用事件做安全性检查，熟练的使用ccBPM的事件是你在驰骋工作流引擎扩展开发的基础。

如果需要做js端的检查，请参考表单引擎操作手册

退回规则

概述

流程引擎的退回与发送，分别是前进与后退，它是流程引擎的基础功能操作，流程的退回根据不同的应用场景，也是需要不同的方式来控制，我们把这些方式叫做规则处理。

退回规则设置

退回规则在节点按钮标签栏目中的退回标签设置，如下图：

图片地址：https://static.oschina.net/uploads/img/201506/22225755\_Bm9s.png

**不能退回**:当前节点不能执行退回功能，当前节点的操作人员就不能看到退回按钮。

**只能退回上一个节点**：只能退回上一个节点，从那里发送来的，就退回到那里去。

**可以退回以前任意节点**:不限制退回的节点，但是退回的节点必须是当前节点以前的节点。

**可退回指定的节点**: 退回指定的节点，此功能需要在流程属性中的可退回的节点中设置它。

图片地址：https://static.oschina.net/uploads/img/201506/22225846\_4eq1.png

**总结：**

1,根据实际业务需求，设置不同的退回方式。

2, 配合退回前、退回后的事件完成业务的可逆的操作。

退回的消息处理

1.执行退回后，系统都会向执行人发送消息，发送对象仅限于上一节点的执行人员，这样上被退回的点上的工作人员就有一个待办工作，如果您集成了ccim它就会自动发一个消息提醒。

2.退回的动作写入WF\_Track中，流程轨迹中就能很好的反应出来。

3.被退回的人在进入当前工作时，第一次会有消息提示。CcBPM如何处理流程退回过程的数据的完整性？

流程在退回时，有一段流程数据就是从当前点到退回点的所做的工作，这部分节点的数据如何处理成为了我们要探讨与取舍的难点。

以请假流程为例，申请人发起，部门经理审批，总经理审批，人力资源归档。如果总经理退回到第一个点，可以解释为，部门经理做的无效的工作，此部分工作需要删除，在3.0以前的版本，ccBPM都是这样的处理的，这样的解释也是用户所接受的。

但是在其它的流程就不能这样解释了，因为他需要保留历史痕迹，并且在退回后有如下可能要发生。

1.退回到指定的点后，发起人删除流程。

2.退回到退回节点后，发起人修改表单后发送，按原节点发回来。

3.退回到退回节点后，发起人修改表单后发送，经历与其它的路线步骤到当前点。

4.退回到退回节点后，发起人修改表单后发送，该走其它的路线不经当前点。

基于如上可能性的发生ccBPM，做了如下处理。

1.退回阶段流程数据写入txt 文件里，放在D:ccflowtrunkCCFlowDataUserReturnLog

2.增加了流程报告与节点的焦点字段功能，系统把每一步骤的操作都记到日志表里了，通过焦点字段的配合，可以让操作员方便明晰的看到轨迹。

CcBPM4.5通过如上两个方法解决退回数据的完整性问题。

退回并原路返回

**与节点属性中的[是否可以退回并原路返回？] 配合使用**

图片地址：https://static.oschina.net/uploads/img/201507/08093715\_WmE3.png

应用场景：一个流程走过了ABCDEFG几个节点，在G节点上发现要退回给B节点上去，还期望B节点的人员完成后直接发送给G节点上来，这种应用场景就是是否可以在退回后原路返回。如果是直接退回并不原路返回，那么ccBPM将会删除退回点与退回到点中间的数据，否则就不删除它。

图片地址：https://static.oschina.net/uploads/img/201506/22230249\_1Pby.png

退回信息填写字段

用户经常会在审批意见的字段中填写意见然后点退回按钮，审批意见就是该操作员的审核意见，这个时候ccBPM需要把审核意见带入退回窗口，这个字段就是退回信息填写字段。

图片地址：https://static.oschina.net/uploads/img/201506/22230319\_ela8.png

在demo的第二个节点，我们看看退回的效果，我们先看看测试效果。

图片地址：https://static.oschina.net/uploads/img/201506/22230332\_Ye9O.png

点退回，ccBPM就会把审核意见放到退回的窗口里面。

图片地址：https://static.oschina.net/uploads/img/201506/22230343\_t7EA.png

跳转规则

**应用背景**：

对于一个各项制度未成熟的单位，各种随机发生违反流程规则的都有可能存在，但是这种行为在当时的管理条件下也是不得已的。流程设计人员在流程设计中未考虑到，或者认为非法的路径在现实的管理活动中需要体现出来。

这就要求流程引擎能够运行指定的节点在可以跳转到指定的位置中去。

**跳转规则**:

  不允许跳转。

  只能向后跳转

  只能向前跳转。

  任何节点跳转。

  跳转指定的节点。

目前ccBPM支持以上5种跳转规则，跳转指定的节点，可以在节点的基本属性里书写一个sql表达式。查询的结果必须有两列，节点编号，与节点名称，顺序不能变化。

如何设置跳转规则？

第一步：在允许跳转的节点打开节点属性，如下图：

图片地址：https://static.oschina.net/uploads/img/201506/22230456\_QgBo.png

根据自己的业务需要选择一种跳转规则：

图片地址：https://static.oschina.net/uploads/img/201506/22230508\_a26h.png

显示效果:

图片地址：https://static.oschina.net/uploads/img/201506/22230521\_QDzQ.png

按下跳转按钮，系统自动显示出来根据跳转规则能要跳转的节点。

**跳转操作界面**:

图片地址：https://static.oschina.net/uploads/img/201506/22230537\_204d.png

选择要跳转的节点后，系统就可以自动转向过去。

抄送规则

**关键字**：ccBPM工作流引擎抄送规则设计

**需求背景**：在流程运转过程中，把关心流程数据的人员分为流程内部人员，与流程外部人员。流程内部人员就是参与流程的工作人员，流程外部人员就是关心流程运行的人员。比如：低于于5000元的报销单据不需要董事长审批，但是董事长如果要求需要知道。这就需要ccBPM在总经理审批后向董事长抄送一份，这就是抄送。董事长不参与整个流程的运转。

**ccBPM设计**：根据用户的需求抄送的内容为：标题、信息内容、工作报告、流程轨迹图。

**操送规则**：不抄送、手工抄送、自动抄送、手工与自动抄送并行。

在节点属性里设置抄送规则

图片地址：https://static.oschina.net/uploads/img/201506/22230643\_RYIz.png

**说明**：

不能抄送：就是不显示抄送按钮。

手工抄送：可以打开抄送功能界面填写信息进行抄送。

自动抄送：需要设计抄送规则才可以进行。

手工与自动：两者并行。

**自动抄送设置**：

图片地址：https://static.oschina.net/uploads/img/201506/22230711\_npLe.png

**控制方式**：

按岗位：抄送的范围设置岗位内的人员。

按部门：抄送的范围是设置的部门人员。

按人员：抄送的范围是指定的人员。

按SQ: 在sql表达式里设置一个对操作员查询语句包含No,Name两个列，ccBPM就会认为这是您的抄送范围。

抄送标题与抄送内容：就是信息标题，与信息内容，支持变量表达式。其中@Accepter 就是接受人，这是特殊的变量。

**操作效果**

发送时：

图片地址：https://static.oschina.net/uploads/img/201506/22230743\_74iL.png

进入看到抄送界面：

图片地址：https://static.oschina.net/uploads/img/201506/22230801\_wPiP.png

打开详细信息数据：

图片地址：https://static.oschina.net/uploads/img/201506/22230816\_CVeH.png

抄送执行顺序：

图片地址：https://static.oschina.net/uploads/img/201506/22230827\_SaAJ.png

在工具栏点抄送按钮，选择抄送人，填写抄送信息，点执行抄送按钮。

抄送写入规则

**关键字**： ccBPM 写入规则

**应用场景**:

节点A运动到节点B, 需要集合1的人去处理，需要集合2的人去查阅但不能处理，这样集合2的人可以使用抄送来实现. 也就是说一个邮件发送给集合1，抄送给集合2.

但是对于操作人员来说，不习惯使用抄送功能来查阅节点信息，或者对于抄送来的信息都不怎么重视，都习惯使用待办来看到需要自己查阅或者处理的工作。

图片地址：https://static.oschina.net/uploads/img/201506/22230952\_d3Vf.png

**设置方法**:

如何实现上面所表达的应用场景，请按照如下步骤实现:

第一步:设置抄送写入规则.

请打开节点属性-》找到基本设置-》找到抄送写入规则-》ccBPM为您提供了如下三种设置。

写入抄送列表：数据写入抄送列表，只有打开抄送才能看到.

写入待办：工作处理信息写入待办列表，与正常工作的待办一样，只是打开这个待办工作就可以直接进入ccBPM的OneWork处理界面.

写入待办与抄送列表:实现以上两者的功能.

第二步:设置抄送规则.

请参考ccBPM的抄送规则的设置，设置要抄送的人员.

**实现原理**:

1, 在WF\_GenerWorkerlist 的Paras 增加了一个属性IsCC=1.

2, 在进入MyFlow.aspx 中增加了一个标记判断, 如果Paras包含IsCC就转到OneWork功能界面.

**升级办法**:

登录流程设计器，用升级的模式进入，就可以完成自动升级.

**其他事项**:

1,当集合1与集合2有交集时，按照待办来计算.

2,需要升级使用该功能.

挂起与解除挂起

**应用场景**:

一件工作因为一些原因,外部的条件不能执行下去，需要把这件工作挂起来，等待条件满足了才能执行下去，这就是节点工作的挂起.

图片地址：https://static.oschina.net/uploads/img/201507/08100459\_g98d.png

**输入参数**:

挂起原因,挂起天数,通知的人员,操作的人人员.

**挂起作用**:

1, 可以告知该参与抄送该工作的人员, 这件工作被延期的原因.

2, 挂起时如果选择了指定的挂起天数,流程引擎就会自动的向它发送一个消息(邮件短信..)进行通知它去处理.

3, 挂起的天数里不能做为时效考核.

**挂起实现机制**:

1, 设计一个表用来存放挂起信息.

MyPK(FK\_Node+WorkID), WorkID(工作ID),Rec(执行人),NoticeTo(通知给),Note(挂起原因),HungUpDays(挂起天数),RDT(操作日期),SendDT(发送日期(工作处理日期)

2, 在节点工作工具栏上增加一个[挂起]按钮.

3, 在工作待办列表里增加一个挂起状态，是正常，预警，预期，挂起的一种。

流转自定义

**自由流程定义**：可以由使用者自定义流程步骤、流程步骤处理人、流程步骤处理模式（单人处理还是多人处理、单人审批还是多人会签）的线性流程。

**特征**：

1.每个节点的表单相同、操作权限相同。  
  
 2.自由化程度高  
  
 3.可以与其他类型的流程一起使用，比如父子流程中，在子流程中添加步骤、或是在父流程中添加步骤。

**缺陷**：

1.不能满足具有转向条件（流程分支）的流程、父子流程以及分合流流程。只能满足没有分支的线性流程（顺序流）。  
  
 2.每个步骤的表单的操作权限都相同，不能分别设置。（后期会实现，可以为每个步骤的每个字段设置权限。）  
  
 3.单节点的流程不能使用自定义模式。

**流程操作演示（演示demo为BPMN2.0版本）**：

**1.首先设置流程**，初始节点可根据相应的需求而定。默认的是2个节点，即开始节点与结束节点，也是实现流程自定义的最低要求的节点数量。

图片地址：https://static.oschina.net/uploads/img/201512/29172155\_Xh4Y.png

**2.设置流程自定义功能。**

设置方式：节点属性---按钮权限---勾选流转自定义。

注意：设置流程自定义的节点，后面必须有一个活动节点，否则，无法实现自定义功能。比如一个流程有2个节点，那么只能在第一个节点中设置；如果有3个节点，可以在第一、第二个节点中设置。可以设置的节点数为：n-1，n不空且不等于1.

图片地址：https://static.oschina.net/uploads/img/201512/29172234\_tAiX.png

**3.设置表单、流程结束事件或其他事件等**。根据具体需求，可进行相关功能、事件、表单的设计，可有可无。

**4.流程运行过程中，在设置流程自定义的节点（步骤）中点击流转自定义按钮，添加流程步骤、选择处理人、选择处理模式。**

图片地址：https://static.oschina.net/uploads/img/201512/29172252\_955T.png

**添加步骤后，效果如图所示：**

图片地址：https://static.oschina.net/uploads/img/201512/29172307\_dAgT.png

点击“执行手动设置运行模式并返回”，就可以按照设置的步骤等内容运行。

**两个功能按钮的区别：**

执行自动运行模式并返回：根据在流程设计器中设计的流程运行。比如流程有2个节点，那么就根据设置的节点访问规则与处理人运行2步，然后结束。

执行手动设置运行模式并返回：根据自定义的设置的内容运行。

打印单据

概述

单据打印是对与表单内容按照一定的格式，输出到打印机上备份、存档。

根据不同的应用场景需要，用户对打印的要求也不一样，所以ccBPM支持如下格式的单据打印。

图片地址：https://static.oschina.net/uploads/img/201506/22231426\_fn5z.png

打印网页

用户打印网页，不需要其他的额外的配置，仅仅需设置该选择项目就可以了，选择该项目后在表单上就出现如下按钮。

图片地址：https://static.oschina.net/uploads/img/201506/22231452\_O3w5.png

点击打印按钮，如果是第一此运行，系统就会提示安装打印插件，该插件是一个免费的插件，如果您需要支持作者，请按照软件地址联系他。

**该方式的局限性**：

1，  仅仅运行在IE上，是一个activeX插件。

2，  不能打印明细表。

打印网页效果：

图片地址：https://static.oschina.net/uploads/img/201506/22231517\_hJlW.png

打印RTF格式模版

概述

流程运转到特定的节点，需要把流程数据打印出来，这种根据流程数据生成特定格式文件的过程叫做单据打印。

单据打印首先需要根据用户的对打印的要求制作一个单据模板。

1, 一个节点表单可以打印多个单据。

2, 单据模版必须是 rtf 格式的文件。

单据模板

单据模板的制作方式，与单据规则我们先看一个描述就明白了。

此模板位于：D:ccflowtrunkCCFlowDataUserCyclostyleFile单据打印样本.rtf

**单据模板事例**

以下是基本的字段输出

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

普通类型字段输出:

文本输出：<Title>输出内容: 按文字输出.

文本输出：<RDT>输出内容: 2011-05-11

日期文本输出：<RDT.NYR>输出内容: 2011年05月11日

金额文本输出：<baoxiaojine>输出内容: 101.123

金额人民币输出：<baoxiaojine.RMB>输出内容: 101.12

金额人民币大写输出：<baoxiaojine.RMBDX>输出内容: 壹百零壹元壹角贰分

布尔类型数据输出：<IsPass>输出 0

布尔类型数据输出： <IsPass> 输出 0

布尔类型数据转意输出： <IsPas.BoolenText> 输出 否

布尔类型数据转意输出： <IsPass. Boolen> 输出 [X] ， 如果是输出 [V]

以性别为例

枚举类型输出: <XB>直接输出 1

枚举类型输出标签: <XBText>直接输出男

以部门为例

外键类型输出: <FK\_Dept>直接输出 01

外键类型输出标签: <FK\_DeptText>直接输出总经理室

签名：<Rec.Siganture>

以下是表格的明细输出：

报销内容  单价  数量  小计

<Dtl.BXNR>  <Dtl.danjia>  <Dtl.shuliang>  <Dtl.xiaoji>

合计    <Dtl.shuliang.SUM>  <Dtl.xiaoji.SUM>

单据样本:

图片地址：https://static.oschina.net/uploads/img/201506/22231854\_nAPj.png

单据打印的调用方式

**默认调用方式**：

在节点发送成功后，ccBPM就会生成表单单据，发送提示界面就会提示单据打印连接。

**按钮调用方式**:

在节点属性里启用打印按钮。

**外部接口调用方式**:

对于傻瓜表单来说可以通过，写一个调用外部接口的方式调用表单。对于自由表单来说可以增加一个button 让这个button 执行一段js脚本调用。

调用URL : ./WorkOpt/PrintDoc.aspx?FK*Node=@FK*Node&OID=@WorkID。

单据打印的调试技巧

我们知道，通常的流程单据打印都是在最后步骤，如果单据出现了问题，调试一个单据需要走完全部的流程才可以。现在提供如下调试技巧.

0, 一定要用记事本或者写字板打开与制作单据模版

1, 把表单绑定在开始节点上，启动第一个步骤就让它生成单据.

2, 如果单据中的字段没有替换下来，直接找到单据文件.

它位于，D:ccflowtrunkCCFlowDataUserCyclostyleFile 直接修改它.

3, 每次的修改都要iisreset 一下，清空ccBPM缓存, 然后调试打印.

4, 主表的数据都可以在开始节点输出出来，从表就要注意修改从表的表名才能输出打印.

单据打印常见问题

1, 乱码. 输出的格式是乱码。

原因：是您的格式文件中有西文字符。

解决办法：全选模板数据，字体，选择宋代或者其它的中文字段。

2, 字段没有别替换掉。

原因：两个尖括号的标记中间有其它的字符，导致ccBPM未识别您的标记。

解决办法：用写字本打开模板文件，搜索字段名，定位到节点标记位置。原来可能是< MyField >替换成<MyField>。就可以解决问题了。

3, 从表未输出，或者从表数据的求和，求平均行出现多行。

原因:在table 标签周围失去了指定的标记.

解决办法:把示例中的表格完整的copy过去然后进行在修改它。

调用外部接口(傻瓜表单有效)

特别说明：

**对傻瓜表单有效**：如果在当前的节点执行相关功能时，可以配置功能连接在这上面。

外部接口的定义是什么？

在自由表单设计器中，已经有了按钮元素，可以替代此部分的功能。

图片地址：https://static.oschina.net/uploads/img/201506/22232124\_Ekpx.png

打印Word模版

**打印效果：**

图片地址：https://static.oschina.net/uploads/img/201507/08104802\_LW76.png

**设置步骤：**

1.在节点属性中设置打印方式为打印word模版

图片地址：https://static.oschina.net/uploads/img/201507/08104819\_kSF7.png

2.将设置好的模版由管理员放入对应的路径CCFlowDataUserFrmOfficeTemplate<流程编号>

图片地址：https://static.oschina.net/uploads/img/201507/08104831\_9gLG.png

**制作模版：**

根据表单，设置模版样式

图片地址：https://static.oschina.net/uploads/img/201507/08104914\_7rZc.png

模版样式设置完成以后，需要在要填充的字段位置加一个书签，主表数据字段的书签名以ccflow*bm*main*开头，后面加字段名称，比如：ccflow*bm*main*Name。

图片地址：https://static.oschina.net/uploads/img/201507/08104955\_FyuH.png

接受人规则

**说明**：当一个节点向下一个节点运动时，如果下一个节点的接受人范围，需要当前节点工作人员来选择的，在当前节点就需要启用接收人按钮。如果当前节点不启用，在发送时，系统就会转到方向选择的功能界面里，让用户在选择接收人，就是说不管用户是否启用这个按钮，系统都不会报错误。

根据不同的应用环境，系统提供了如下几种规则：

图片地址：https://static.oschina.net/uploads/img/201506/22232236\_0Ai5.png

**不启用**：就是不显示接受人按钮。

**单独启用**：接收人按钮，就会在控制区域显示。

**在发送前打开**：用户点击发送按钮后，首先打开接受人，选择接收人确定后，在执行发送动作。

**转入新页面**：用户点发送后，首先执行表单的保存，然后转入方向选择界面，在方向选择界面里，设置下一步骤的接受人，如下图所示：

图片地址：https://static.oschina.net/uploads/img/201506/22232301\_00nM.png

结束流程

一条流程走到一定的步骤后，当前的节点有权利停止该流程向下运动，但是他不能把流程删除掉，该数据仍然需要保存起来，这种操作叫结束流程。

**结束流程与删除流程不同的是:**

1结束流程数据仍然在存在.

2，流程结束了WFState是结束状态，而删除则是删除状态。

启用结束流程后，结束按钮才能显示出来，根据用户的场景来确定是否启用结束流程按钮，默认是不启用。

其他：也可以使用流程完成条件，让其ccBPM自动判断该条件是否成立，如果成立，该流程自动结束。启用当前按钮是手工结束流程，设置流程完成条件的模式是让系统自动结束流程。

总结：结束流程的模式有三种，第一手工结束、第二根据流程属性的流程完成条件自动结束、第三流程走到最后一个节点自动结束。

流程删除规则

**应用背景**：

流程启动运行后，在一些节点当前处理人有权利执行流程终止，把当前的流程实例进行删除，根据不同的应用场景删除的处理方式有很多种，每种删除方式都对应ccBPM相关的api操作，下面我们把这几种删除方式说明一下，以方便让设计人员根据实际的规则来选择不同的方式。

**不能删除**: 该流程不允许用户删除.

**逻辑删除**：仅仅做上删除标记, 数据仍然在节点表单与流程报表里面.

图片地址：https://static.oschina.net/uploads/img/201506/22232508\_0fvw.png

**记录日志方式删除** : 删除节点表数据，流程报表数据，并记录到 wf\_workflowDeleteLog 表里作备案.

**彻底删除** : 彻底清除该流程的所有数据，包括该工作实例的节点表单数据、流程报表数据、轨迹数据、退回、移交操作信息。

**让用户决定删除方式** : 显示出对话框，让用户选择上面的删除方式.

**其他**：删除规则的API请查看代码: BP.WF.Dev2Interface.Flow\_DoDeleteFlowBy \*

1，  开始节点如果设置了删除按钮，如果不是退回来的节点，或者没有发送过的节点，就可能导致错误。

保存

**说明**：如果启用的保存按钮，那么在表单中，可以将填写的信息保存，下次再打开表单时，上一次填写的信息还存在表单中。

**扩展应用**：

图片地址：https://static.oschina.net/uploads/img/201507/28104850\_YAvM.png

我们在保存的功能中，添加了新的功能，就是草稿功能。

**点击流程属性，选择草稿的保存方式。**

**保存到待办**：填写完表单信息后，点击保存，将会把当前填写的流程放入待办中。可以填写多个表单并保存到待办，并不会覆盖之前填写的内容。虽然写入待办，但是当前的活动节点还是开始节点。

**保存到草稿箱**：填写完表单后，点击保存，将流程放入草稿箱中。下次需要发起时，去草稿箱选择发送即可。

轨迹

**说明**：查看流程运行的线路以及，运行中的一些基本信息。比如：节点处理人、运行轨迹、附件等。

图片地址：https://static.oschina.net/uploads/img/201507/28110950\_jPB2.png

启用轨迹按钮后，表单中会显示出来。

图片地址：https://static.oschina.net/uploads/img/201507/28111036\_LMDU.png

点击轨迹按钮，展示的信息如下图所示：

图片地址：https://static.oschina.net/uploads/img/201507/28111049\_Pj3F.png

查询

**说明：**启用查询按钮后，可以在流程中点击查询按钮，进行流程数据查询。

图片地址：https://static.oschina.net/uploads/img/201507/28112347\_qy3j.png

点击查询按钮，如下图所示：

图片地址：https://static.oschina.net/uploads/img/201507/28112404\_Cj4n.png

公文按钮

概述

公文按钮是用来控制公文工具栏上的操作功能，它与按钮权限不同的是，按钮权限控制的是流程引擎功能，而公文按钮是控制的公文的的文件操作。

图片地址：https://static.oschina.net/uploads/img/201506/22232637\_Hs3b.png

公文流程定义与组成

公文流程也叫word表单流程，它是在一个业务表单加一个公文表单组合在加一个附件组件组合的，如下图：

图片地址：https://static.oschina.net/uploads/img/201506/22232708\_PjBv.png

以上是公文正文表单，一下是公文基础信息表单。

图片地址：https://static.oschina.net/uploads/img/201506/22232718\_Xyni.png

公文流程相关的文件目录

公文类的流程需要一些模版文件、套红图片、公章图片、以及流程实例文件来存储，归纳起来有两种类型，一类是设计设置的模版类文件、一类是流程运行中的数据文件也叫流程实例公文文件。

|  |  |  |
| --- | --- | --- |
| **文件类型** | **位置** | **说明** |
| 公文模版文件 | DataUser\OfficeTemplate |  |
| 公章图片 | DataUser\OfficeSeal | 图片签章，电子签章是由第三方厂商提供 |
| 套红模版 | DataUser\OfficeOverTemplate | 套红文件格式是 .doc、.docx、.jpg等 |
| 公文实例文件 | DataUser\OfficeFile | 公文流程运转的事例都会以日期的格式存储 |

在对于前三者的流程设计的维护，是可以在流程设计器的后台管理可以维护的。

图片地址：https://static.oschina.net/uploads/img/201506/22233019\_Y3Ug.png

设计一个公文流程的步骤

**1.概述：**

**表单效果图：**

图片地址：https://static.oschina.net/uploads/img/201512/29183908\_rwuG.png

**选择模版效果图：**

图片地址：https://static.oschina.net/uploads/img/201512/29183920\_srPS.png

**公文编辑：**

图片地址：https://static.oschina.net/uploads/img/201512/29183931\_a6nb.png

**附件上传：**

图片地址：https://static.oschina.net/uploads/img/201512/29183942\_8Zey.png

**2.公文表单与相关流程的实现步骤。**

1.需要安装activex插件：weboffice（在线编辑器）（重点）

涉及到插件的问题，就会设计到浏览器兼容与版本问题。在ccflow中，公文插件已经兼容ie8以上版本，chrome，firebox，基于其他浏览器内核的360浏览器或其他浏览器。

**安装步骤：**

第一步：执行weboffice.exe安装文件。

第二步：安装成功后，执行ffactivex-setup-r39.exe文件。

如果以上两步完成之后，不能在除IE浏览器之外的浏览器中打开，再执行第三步：注册WebOffice.ocx文件。

注意：使用在线编辑公文时，要保证机器的Misoffice安装正确。如果每次打开word都需要重新注册计算机，会浪费很多时间（解决方法问百度）。如果安装不正确，会造成公文打开失败。

**2.设计流程**

根据需要，设计符合自身逻辑的流程，这个不再赘述。

**3.表单设计（重点）**

其他表单的内容在这不去一一说明了，只讲怎么设置公文与公文附件相关。

**a.公文相关：**

打开节点属性—》按钮权限

选择启用公文，有两种方式：tab标签页和按钮。

图片地址：https://static.oschina.net/uploads/img/201512/29184000\_hlfy.png

节点属性—》公文按钮

选择可以对公文进行的操作。上传公文的节点必须有：打开模版或者打开本地 中的一个权限。

模版存放位置：（将做好的模版放入这个位置，打开模版时，可以直接选择，概述中，有相关图例）

ccflowvalueaddedCCFlowDataUserOfficeTemplate

图片地址：https://static.oschina.net/uploads/img/201512/29184017\_jUBX.png

**b.附件相关：**

第一种情况：需要附件显示在表单中   
  
 在表单添加一个多附件组件或者单附件组件即可。  
  
 第二种情况：附件不显示在表单中，只在公文中显示。  
  
 在表单的最下方（建议）或其他地方添加一个多附件组件或单附件组件。右键编辑，选择隐藏组件。如下图进行设置：

图片地址：https://static.oschina.net/uploads/img/201512/29184113\_L2vp.png

**附：**

1.如果不需要附件，不添加就可以。

2.上传公文节点的后面的节点，也需要设置启用公文（tab页或按钮方式）。如果有附件，后面的节点表单也需要有附件组件。

公文按钮属性配置

本章节详细讲述每个属性的功能，如下图：

图片地址：https://static.oschina.net/uploads/img/201506/22233206\_V1N0.png

**属性列表**:

|  |  |  |
| --- | --- | --- |
| **功能选项** | **解释** | **备注** |
| 打开本地 | 是否需要打开本地的公文模版文件？ | 打开本地的word文件 |
| 打开模版 | 是否允许打开服务上传的公文模版文件？ | 模板是在流程设计器中的系统维护 中上传的文件 |
| 保存 | 是否启用保存按钮？ | 将文件保存到服务器 |
| 接受修订 | 对上一次的修改是否接受 | 接受用户对文档进行的内容更改 |
| 拒绝修订 | 拒绝对文档的留痕修订 |  |
| 套红 | 在发文之前，需要在word的页眉增加一个图片。 | 一般是一个红头文件 |
| 是否查看留痕 | 查看对文档留痕的痕迹(接受修订和拒绝修订后痕迹会消除) |  |
| 是否只读 | 文档只读，不能对文档进行任何操作 |  |
| 打印 | 调用word的打印功能。 | 打印当前的word文档 |
| 签章 | 在当前的光标位置，插入选择的公章图片文件。 | 此签章不等于电子签章，他是图片 签章，电子签章需要第三方的软 件支持。 |
| 插入流程 | 是否需要在当前光标位置插入当前的流程图片。 |  |
| 是否记录节点信息 | 会在文档底部记录当前节点的名称操作人时间格式： 操作人节点名称时间 |  |
| 是否自动保存为PDF | 保存为pdf格式文件。文档保存后会保存为pdf格式的文档 |  |
| 下载 | 是否允许当前用户下载该文件 |  |
| 是否使用父流程文档 | 对父子流程有用。子流程延用父流程的公文 |  |
| 是否自动套红 | 是否在套红节点根据规则自动套红，主要是根据表单的发文类型来确定红头文件 |  |
| 自动套红文档 | 此文档是xml文件，需要配置表单字段具体值以及相关的套红文件 |  |
| 是否进入留痕模式 | 对当前文档的操作保留word的痕迹,当修订之后才会对文档进行实际修改 |  |
| 指定文档的模板 | 该模板必须是已经存在的模板，当用户未上传正文时会自动打开该模板 |  |

以上功能列表，不一定都要记住，但一定要了解，正如我们所讲的一样ccflow有1千个功能你用到了就六七十个，但是你要熟悉他们，不理解的可以查手册帮助。

考核属性

概述

图片地址：https://static.oschina.net/uploads/img/201506/22233851\_AFiy.png

以上是对考核信息的设置，因为考核来说各个单位规则都不相同，模式也不想同。所以ccBPM只记录此部分的设置。具体的时效考核与质量考核的开发由二次开发人员来决定。

时效基本设置

一个任务分配下去后，它有三个状态，分别是：正常、预警、逾期。

警告期限，就是提前多少天预警。限期，就是该工作需要多少天完成。

**比如**：设置限期3天完成，警告期限是1天。

那么，周1接受到的任务，应该是周4完成，如果周3还没有完成，就成警告状态了。

超时处理

超时处理就是对一个操作员的工作超过指定的时间后如何去处理？ccBPM为我们提供如下规则去处理

|  |  |  |
| --- | --- | --- |
| **规则名称** | **解释** | **表达式配置** |
| 不处理 | 不做任何处理仍然在当前节点上，这是默认的选择项 | 不配置 |
| 自动向下运动 | 让其自动的运行到下一个节点上。比如：逾期没有审批，让其自动审批。 | 不配置 |
| 自动跳转指定的节点 | 发送给指定的节点指定的人员， 在处理内容里个配置格式为： 节点编号+“，”+处理人编号。 比如：102,zhoupeng | 例如: 105,zhoupeng 意思是：跳转到节点105， 让zhoupeng处理。 |
| 自动移交给指定的人员 | 当逾期时，就移交给指定的人员处理。 | 格式为:用户编号 比如： zhoupeng 注意只能配置一个操作人员. |
| 向指定的人员发消息 | 向指定的人员发消息。 | 格式为:用户编号1, 用户编号2 比如：zhangsan,lisi |
| 删除流程 | 彻底的删除该流程。 | 无配置 |
| 执行SQL | 执行sql，处理应用业务逻辑。 | 配置一个sql语句支持ccflow的表达式。 |

**超时处理测试用例在demo中**

图片地址：https://static.oschina.net/uploads/img/201506/22234228\_uoXR.png

CcBPM如何来判断超时的:

根据工作时效的设置，ccBPM就会自动计算出来应该完成的日期，这个日期在流程引擎注册表里WF\_GenerWorkfLow 字段: SDTOfNode，如果您设置了节假日，那么ccflow就会自动扣除节假日。

处理超时工作需要ccBPM的服务支持，如下图：

图片地址：https://static.oschina.net/uploads/img/201506/22234242\_S3t9.png

在系统管理里有节假日设置功能界面：

工作质量考核

**关键字**：CCBPM工作质量考核时效考核

**需求背景**:

我们把在工作流程引擎中的考核分为两种模式，一种是时效考核、另外一种是质量考核。

时效考核就是对用户的操作进行时间点的记录，然后按照一定的规则进行计算出来该人员提前、超时、逾期完成工作的情况进行量化的评分，这种考核模式完全依靠计算机来自动准确无误的完成。应用此模式能够对工作效率进行完整的监督评价、也可以进行对数据分析、及时地调整岗位、及时地抽调人员、让单位的人员布局更加合理。

工作质量考核就是对相关人员对当前参与的以前人员的工作的质量（做的好坏、认真、符合要求）进行工作质量评价，这种考核模式需要人员主观的认定、检查、评价、评估。

这里仅仅研究工作质量考核，对于时效考核清参考其它的章节。

工作质量考核的原则与需求：

1，工作质量考核首先是一个后面的工作节点考核前一个工作节点、不能跳跃考核其他的节点。

2，我们把考核人对应的节点叫做考核节点、被考核人考核的节点叫被考核点。

3，根据上述的结论、所以考核节点与被考核节点是相联结的、并且考核节点在后。

4，工作质量考核需要在4个字段的输入：被考核人的编号、被考核人名称、被考核人得分、评分说明。

5，根据节点的运行类型不同，如下两个组合的考核点才有效。

a 被考核点是普通节点，考核点是普通节点  
  
 b 被考核点是普通节点，考核点是分流点节点  
  
 C被考核点是子线程节点，考核点是合流点节点  
  
 d 被考核点是合流节点，考核点是普通节点

6，分析以上的组合类型得出如下两种考核模式:

考核点的数据输入在主表上(a,b,d模式)，考核点的数据输入在从表上(c模式)。

7，如果流程撤销发送，该节点上的质量考核数据要删除，退回则没有影响。

8，如果流程被物理删除，此流程上所有的质量考核数据都要根据workid删除。

**CcBPM的设计**:

CCBPM创建了一个表WF\_CHEval，来存储这些质量考核数据、因为涉及到质量考核各个单位的应用就很难抽象，所以CCBPM目前还没有对此表的数据展现、利用进行更高级的挖掘与利用，如何利用此表数据进行为单位服务是二次开发的事情，ccBPM仅仅负责把数据的输入填充到该表里。

此表的结构如下:

图片地址：https://static.oschina.net/uploads/img/201506/22234657\_lrCI.png

CcBPM的实现、配置:

现在以一个demo流程事例来说明如何配置质量考核:

图片地址：https://static.oschina.net/uploads/img/201506/22235123\_CYdq.png

第一步：把考核节点在节点属性里配置：

图片地址：https://static.oschina.net/uploads/img/201506/22235135\_CmfR.png

第二步：在合流节点的明细表汇总表里增加系统工作质量考核字段:

建立一个明细表后，插入列组：

图片地址：https://static.oschina.net/uploads/img/201506/22235239\_jaZX.png

插入后效果：

图片地址：https://static.oschina.net/uploads/img/201506/22235252\_XiJT.png

第三步：在最后一个节点上设置质量考核的字段，并在节点属性里标记该节点是质量考核节点。

图片地址：https://static.oschina.net/uploads/img/201506/22235305\_hu1H.png

选择后系统就会自动增加如下4个质量考核数据采集字段，其中被评估人员编号与名称都是只读的，流程在运转过程中因为，您已经标记了该节点是质量考核节点所以ccBPM就会把上一个节点操作人员的信息放在这里。

图片地址：https://static.oschina.net/uploads/img/201506/22235321\_0ESe.png

当您发送给此节点后，ccBPM就会想WF\_CHEval 表里插入一条数据, 如果您撤销了发送，此数据将会被删除，如果流程物理删除则会根据workid删除。

第四步：测试并运行流程

这里只看第3，4两个考核点。

图片地址：https://static.oschina.net/uploads/img/201506/22235346\_vxDx.png

最后节点的工作质量考核数据采集

图片地址：https://static.oschina.net/uploads/img/201506/22235402\_EgJh.png

第五步骤：查看质量考核数据知否被正确的写入WF\_CHEval 表。

图片地址：https://static.oschina.net/uploads/img/201506/22235424\_1Q6L.png

质量考核数据与实效考核数据时做好考核系统的基础。

扣分得分设置

因为考核是非常复杂的工作，这里只做配置使用，并不体现在考核上，考核的统计分析，执行，需要开发人员读取轨迹表来完成。

相关功能

概述

图片地址：https://static.oschina.net/uploads/img/201507/08111012\_GG54.png

相关功能：请看左边的菜单栏目，以下是分别介绍与当前节点有关系的相关功能。

节点绑定岗位

节点岗位是维护节点与岗位的对应关系，它用于节点权限访问规则。

图片地址：https://static.oschina.net/uploads/img/201507/08111031\_60I7.png

用来控制可以访问以节点的工作岗位，那些岗位可以访问此节点处理此节点的工作任务。如果此节点是开始节点，那就是那些人员可以发起工作。

注意：当投递规则中，设置按岗位投递才有效。

节点绑定部门

节点部门是维护节点与部门的对应关系，它用于节点权限访问规则。

图片地址：https://static.oschina.net/uploads/img/201507/08111117\_7JYh.png

用来控制可以访问以节点的部门，那些部门可以访问此节点处理此节点的工作任务。如果此节点是开始节点，那就是那些人员可以发起工作。

注意：当投递规则中，设置按部门投递才有效。

节点绑定接受人员

用来控制可以访问以节点的人员，那些人员可以访问此节点处理此节点的工作任务。如果此节点是开始节点，那就是那些人员可以发起工作。

图片地址：https://static.oschina.net/uploads/img/201506/22235757\_fHaZ.png

注意：当投递规则中，设置按人员投递才有效。

消息收听

什么是消息收听

图片地址：https://static.oschina.net/uploads/img/201507/08111142\_rmZi.png

消息收听，在流程运行时，向指定的节点处理人发送消息。收听消息的点叫做侦听点.被收听的点叫发送点。消息接受有三种方式：短信、邮件、邮件+短信.

比如：财务报销流程中, 发起流程人要在第一时间知道部门经理审批内容、总经理审批内容.如果财务审批通过了，我就可以第一时间到出纳那里领取报销款。

流程引擎中贯穿了这样的消息机制，就可以把每个岗位上的人员有机的连通起来，协调起来，使整个单位的整体工作效率更高。

此消息收听机制就可以解决此问题。

如何设置消息收听

1, 选择侦听节点，找节点属性（在节点属性里，点消息收听按钮，系统弹出消息收听设置的对话框。

2，点设置消息收听。

3，设置收听内容格式.

4, 在当前节点设置收听的节点，意义是当前人员收听指定节点发送来的信息。

消息设置内部支持系统参数，类似于单据设置。

比如：

标题设置：您的借款申请已经被总经理@Rec审批通过,核准借款金额为@JieKuanJier。

@Rec， @JieKuanJier 就是变量。

消息收听对应的表

表结构：

图片地址：https://static.oschina.net/uploads/img/201506/22235939\_RLcd.png

表内容：

图片地址：https://static.oschina.net/uploads/img/201506/22235950\_bvjt.png

表单库

概述

**说明**：本章节仅仅对驰骋表单的做一些概要的说明，具体的表单设计请参考《驰骋工作流引擎-表单设计器操作手册.docx》。

**表单库**：顾名思义就是表单的仓库，他是存放表单的地方。我们把流程比喻成火车，那表单就是车厢。火车的用途就是运输旅客货物，那末流程的作用就是运输数据。因为运输旅客、不同的货物，需要不同的车厢，同样流程在运输数据时也需要不同的表单来盛放这些数据。火车有客车车厢、火车车厢、油罐车车厢，那么表单的类型有傻瓜表单、自由表单、自定义表单、Excel表单与Word表单。开发人员可以根据不用的客户需求选择不同的表单。

火车车厢是有工厂制造出来的，那么表单是有表单设计器设计出来的。

不同的火车车厢有不同的制造厂，那么不同的表单也有不同的表单设计器。

**表单设计器**：表单设计器是设计表单的工具，如何设计一个表单，请查看驰骋表单设计器文件。

**表单库的设计**：有两个重要的表来展现与存储表单，如下图

图片地址：https://static.oschina.net/uploads/img/201506/23000133\_w38O.png

表单的调用：请参考《驰骋工作流引擎-表单设计器操作手册.docx》

表单类型

概述

不同的货物需要不同的车厢去运输，同样不同的数据也需要不同的表单去盛放，驰骋表单引擎与驰骋工作流程引擎珠联璧合，既保持各自的独立性也保持紧密的藕荷功能，才能有效的发挥流程引擎的能力。

傻瓜表单

**优点**：界面清晰、简洁、紧凑。

**缺点**：格式固定。

图片地址：https://static.oschina.net/uploads/img/201506/23000243\_9hGx.png

自由表单

**优点**：界面元素可以根据用户的需要任意排列。

**缺点**：样式不能调整。

图片地址：https://static.oschina.net/uploads/img/201506/23000313\_XvCA.png

自定义表单

如果傻瓜表单与自由表单都满足不了用户的需求，那就使用第三方的工具比如vs自己定义一个表单，然后把url配置里里面去。我们成为自定义表单。

Office表单

这里说的office表单是说的word与excel表单。

他是借用office在线编辑的功能实现对office文件存储，更多的信息请参考驰骋表单设计手册。

创建表单

概述

第一步：在表单库中增加点新建表单，如下图。

图片地址：https://static.oschina.net/uploads/img/201506/23000434\_OsKF.png

第二步：选择类别与表单类型，点保存以后，其它的都与ccform设计一样了，下图：

图片地址：https://static.oschina.net/uploads/img/201506/23000447\_CtkS.png

创建一个自由表单

创建自由表单很简单了，点创建，就可以根据自己的需要画出表单

创建一个Excel表单

**1.在表单库中创建一个excel类型的表单：**

图片地址：https://static.oschina.net/uploads/img/201507/08181002\_pErI.png

在Excel表单编辑界面，通过拖拽添加主表字段，这些字段都是需要在Excel表单模板中进行收集数据的，不需要在此Excel表单设置中进行排版，只需要增加字段以及设置字段信息【目前只支持文本字段】；也可增加明细表，设置明细表的字段信息，保存该Excel表单。

**2. 在Excel表单编辑界面**，点击“扩展设置”工具栏按钮，打开此表单的扩展设置界面，选择左侧“装载填充”，打开装载填充设置，如图所示：

图片地址：https://static.oschina.net/uploads/img/201507/08181029\_vDAu.png

图片地址：https://static.oschina.net/uploads/img/201507/08181036\_fRGM.png

该项设置在Excel表单中自动填充的数据获取SQL语句，分别有主表SQL设置和明细表SQL设置。在主表SQL设置中，可以用@OID来获取当前工作(WorkID)的记录。此处设置的SQL语句，在Excel表单加载时，会直接将通过SQL语句获取的数据填充到Excel模板文件中各自对应的单元格中，修改这些单元格中的值，保存，即可以将Excel中更改的数据保存到数据库中，主表数据保存到Excel表单的物理表中，明细表数据保存到Excel表单的明细表中，明细表物理表名格式为：Dtl+Excel表单编号+序号(从0开始)。

**3. 设置excele表单模版。**

Excel表单需要设置相应的模板，模板文件即为Excel文件，在模板文件中，需要设置主表字段以及明细表字段的填充单元格信息等，这些设置是与第2步Excel表单设置中的主表、明细表信息一致的。Excel表单模板文件编辑完成后，通过Excel表单界面“扩展设置”中的“Excel表单属性”界面中的上传功能与Excel表单进行绑定，请参见上一步中的说明。

新建一个Excel文件，编辑排版成自己想要的格式，原则上是利于用户进行录入，便于数据采集。如下图所示：

图片地址：https://static.oschina.net/uploads/img/201507/08181117\_1MZw.png

主表设置：

在需要录入主表数据的单元格处，点击右键，选择“命名单元格区域”【此版本为Excel 2007】，如图所示：

图片地址：https://static.oschina.net/uploads/img/201507/08181131\_NYmh.png

打开新建名称对话框，如图所示：

图片地址：https://static.oschina.net/uploads/img/201507/08181143\_Fjpc.png

填写“名称”，此名称与对应的主表字段名称一致，“范围”选择当前Sheet的名称【此处Sheet的名称需要说明一下，一个Excel表单模板中可以包含该表单的多个Sheet，名称以Excel表单编号+"\_"+序号（从1开始）】，“引用位置”就是当前选中的单元格【此处如果在其他版本的Excel中的右键没有“命名单元格区域”项，可以找到“名称管理器”，打开后点击“新建”，来创建命名区域与主表字段进行关联】，单击确定即可建立这一个与主表字段关联的命名区域，之后的所有主表字段建立与此类似，“名称管理器”如图所示：

图片地址：https://static.oschina.net/uploads/img/201507/08181159\_8Vb3.png

图片地址：https://static.oschina.net/uploads/img/201507/08181207\_PY5x.png

明细表字段设置：

明细表填充创建字段关联，首先选择整个明细表区域，单击右键，选择“命名单元格区域”，弹出新建名称对话框，“名称”组成为：dtl\_+明细表编号，“范围”选择当前Sheet的名称，点击确定即可，这是先定义了明细表的范围，然后进行明细表字段的定义。

每个明细表的第一行为标题行，目前只支持定义一行标题，不支持多行标题。明细表字段也是定义在标题行内的，选中标题行的一个单元格，单击右键，选择“命名单元格区域”，弹出新建名称对话框，“名称”组成为：dtl*+明细表编号+*+当前单元格代表的明细表字段名[+\_+列汇总函数（如：COUNT,COUNTA,SUM,AVERAGE,MAX,MIN等，原则上只要是excel中定义的列函数都可以应用，根据实际情况进行设置）]，“范围”选择当前Sheet的名称，点击确定，则当前单元格所在列就会填充明细表的指定字段。按照以上步骤，设置所有需要的明细表字段在excel表单模板中的关联即可。

如果明细表字段名称中填写的字段为rowid，则系统会自动在该列填充行序号。

需要说明的是，一般明细表有多行，如果定义的明细表填充区域的行数小于“明细表SQL”查询出的行数，则系统会自动增加缺少的行，如果某个明细表字段列定义有汇总函数，系统也会自动增加汇总行。

一个Excel文件包含多个表单模板

目前的规则是，可以在一个excel文件中设置多个excel表单模板，前提是这多个excel表单是绑定在同一个流程节点上的。之所以这样设置，是因为exc el本身的运行机制决定的，在用activex控件操作excel时，因为其本身是单进程多文档的应用程序，所以控件只能同时操作一个excel进程，但我们的流程节点是可以绑定多个excel表单的，所以只能做此设置，将绑定到该节点的多个excel表单模板放于一个excel文件中，这样就实现了在一个excel进程中进行操作，在excel表单界面使用程序来进行控制切换这多个表单。

一个excel文件中，可以增加多个sheet页，新建一个文件，默认会有3个sheet页，名称分别为Sheet1、Sheet2、Sheet3，如果在Sheet1上定义的是编号为ExcelCSBD的excel表单模板，则需要将此sheet的名称Sheet1改为ExcelCSBD*1，即Excel表单编号+"*"+序号（从1开始），如果这个Excel表单有要定义多个sheet页，则每个sheet页的名称都要改成这种格式，后面的序号从1开始叠加。sheet页缺少，则手动增加新sheet页，并将名称改为规定格式，如图所示：

图片地址：https://static.oschina.net/uploads/img/201507/08181237\_vedR.png

不符合规定格式的sheet页不作为excel表单模板使用。

**4.选择模版在表单中上传：**

选择左侧的“Excel表单属性”项，打开Excel表单属性设置界面，如图所示：

图片地址：https://static.oschina.net/uploads/img/201507/08181301\_7lWi.png

该界面可以上传Excel表单模板文件，单击“浏览...”按钮选择模板文件，文件扩展名必须为.xls或.xlsx，选择后，单击保存进行上传保存模板文件，此文件在服务器上的保存路径为：ccflow网站所在目录DataUserFrmOfficeTemplate；保存文件名为Excel表单编号。此界面中单击打开“Excel配置项”打开的界面与“表单权限与显示顺序”界面中单击“ToolbarExcel”打开的界面是一样的。

需要注意的是：如果某节点绑定了多个excel表单，编辑好excel表单模板文件后（将这多个excel表单的模板放于一个excel文件中），要分别在这绑定的多个excel表单的扩展设置中的“excel表单属性”页面，分别上传这同一个excel表单模板，比如：这个节点绑定了3个excel表单，则要将这个共同的excel模板文件分别在这3个excel表单的扩展设置中，上传这个模板文件，否则使用时会报错。

**5.绑定节点表单：**

在流程设计器中，打开要绑定Excel表单的流程，在要绑定Excel表单的节点上，单击右键，选择“绑定流程表单”。

打开“流程表单绑定”界面，从其中的表单列表中选择该节点要绑定的Excel表单，可以多选，选中后单击“Save”，如图所示：

图片地址：https://static.oschina.net/uploads/img/201507/08181332\_87dU.png

**6.设置表单权限（可选）**

进入“表单权限与显示顺序”设置界面，在该界面列出刚才选择绑定到节点的所有表单，每个绑定表单都有一些设置项，可以根据实际情况进行设置【当前版本中，有一些设置项还没有生效，比如“是否可编辑”、“是否可打印”等都没有生效】，如图所示：

图片地址：https://static.oschina.net/uploads/img/201507/08181352\_hJRe.png

需要说明的是“ToolbarExcel”项，该项设置每个Excel表单开放的功能权限，比如保存、下载等，设置完毕，单击“Save”按钮保存修改。

创建一个Word表单

多表单节点流程

概要

本章节讲述表单库、节点绑定流程表单、流程表单在不同的节点的权限控制、表单如何被其他程序调用。

关于表单的设计基础知识请参考《驰骋工作流引擎-表单设计器操作手册.docx》。

**表单库**:简单的说就是表单的仓库，它是用来存放设计好的表单模版的，在与流程的节点绑定时或者需要单独调用时，才能被实例化并被用于到数据的采集与展现。

表单就是为数据采集与数据展现所用的。

表单被实例化的方式有两种：一是被节点绑定被调用，二是被其他的应用程序单独调用。

图片地址：https://static.oschina.net/uploads/img/201506/24091742\_UTde.png

在用户点【绑定流程表单】时，系统弹出表单库，让用户选择绑定。

节点表单与流程表单的概念

这两个概念比较容易混淆，我们用如下定义来描述两个的区别与联系。

1,在表单库中的表单就称为流程表单。

2,一个节点可以绑定多个流程表单。

3,一个节点必须有一个节点表单，但是这个节点表单可以隐藏起来，不使用它。

4,方向条件的判断是根据节点表单进行，不依据流程表单元素。

5,节点表单就是火车头，流程表单就是火车的车厢，一列火车必须有一个火车头。火车头+车厢才组成一列火车，一个车厢可以与其它的火车头挂接。这就是节点表单与流程表单的关系。

6,每个流程表单对应一个表或视图，也可以多个流程表单对应一个表或视图。

7,流程表单的主键是OID，它是一个int类型的由BP框架控制的增长的数值，您不用考虑它的重性问题，如果把流程表单绑定到节点上，这个ID就是WorkID。

节点绑定流程表单

概述

**应用场景**：在很多场合下一个节点上需要绑定多个表单的，我们把这样的节点叫多表单节点。一个节点绑定多个表单，他们之间的关系就是连接关系，流程表单的数据该怎么存储也怎么存储，与流程的运动无关系。我们用如下图来展示多表单流程的运行方式。

第1步绑定表单设置相关属性

**第一步，绑定多个表单，打开流程设计器，选择一个节点右键，选择绑定表单，如下图：**

图片地址：https://static.oschina.net/uploads/img/201506/24092052\_mi2j.png

**进入表单绑定操作功能界面，选择要绑定的表单，如下图：**

图片地址：https://static.oschina.net/uploads/img/201506/24092207\_630k.png

**点保存，进入下一步，在点保存，至于各个权限的操作，我们会在下个章节里进行。**

图片地址：https://static.oschina.net/uploads/img/201506/24092214\_FRrA.png

在这里可以调整顺序执行相关的操作

第2步设置节点属性

**第二步，打开节点属性，找到标签页，选择节点表单方案为自由表单，或者表单树。**

图片地址：https://static.oschina.net/uploads/img/201506/24092413\_FYSd.png

**说明表单树，与标签页表单是两种不同的多表单展现方式。**

多表单的展现方式

标签页模式

**说明：**使用标签页的方式实现对流程表单在该节点的排列，如果当前表单的数据有变动，则ccform会在标签页角上显示\*号，如果标签页切换了，该表单自动保存并且星号自动消失。

**粗体**说明：如果在自由表单下，该节点上绑定了多个表单，系统就默认为按照标签页处理。

图片地址：https://static.oschina.net/uploads/img/201506/24092616\_uQMq.png

表单树模式

**设置方式：**与标签页模式不同的是，在节点属性选择使用“表单树”。

图片地址：https://static.oschina.net/uploads/img/201506/24092754\_VpzS.png

**表单树的展示效果**

图片地址：https://static.oschina.net/uploads/img/201506/24092905\_hxcn.png

**使用说明：**树形表单工作界面以传统的界面布置方式展现，以头部流程工具栏，左边部分表单树，右边是表单展现区域。如果打开多个表单系统以标签页的模式展现，如果表单内容发生变化了，在标签页上会以\*号显示，在切换标签页是系统会自动保存。

节点与流程表单权限控制

概述

**应用背景：**一个流程有多个节点，一个节点绑定多个表单，同样一个表单可能需要在不同的节点上绑定，在每个节点上有不同的人员去处理，相对于同一个表单来说，字段权限、附件权限、从表权限需要在不同的节点上进行控制。

**应用案例：**发起一个企业资质审批流程，受理节点需要填写多个表单，对一个表单来说，开始节点能填写，在第2个节点上预审人员可以修改部分字段，修改上传附件，或者增加附件。第三个节点完全不能修改，只能查看。

**权限控制分类：**不同的节点对一个表单的权限控制有三部分，字段权限，附件权限，明细表权限。本章节就这三部分讲解如何设置与控制。

设计步骤

**步骤1：**增加一个流程表单，拖放一些字段控件，一个多附件，一个明细表，让其都可以编辑，如下图：

图片地址：https://static.oschina.net/uploads/img/201506/24094710\_7h7t.png

**步骤2：**设计一个两个节点的流程，把这个表单绑定到两个节点上。

图片地址：https://static.oschina.net/uploads/img/201506/24094726\_gsFd.png

在节点上右键，选择绑定流程表单菜单，进入下一步骤，设定绑定。

图片地址：https://static.oschina.net/uploads/img/201506/24094739\_SpW1.png

**步骤3：**在节点2上设置字段、多附件、从表的权限，请参下面的[节点流程表单权限配置]章节设置配置表单元素权限。

**步骤4：**执行流程检查并运行流程。

运行过程演示

**打开流程设计器，我们点击运行按钮，如下图：**

图片地址：https://static.oschina.net/uploads/img/201506/24095044\_BPWc.png

**在开始节点上填写必要的字段，上传必要的文件，填写必要的明细表。**

图片地址：https://static.oschina.net/uploads/img/201506/24095101\_pJ07.png

**发送到下一个节点上去，打开下一个节点信息，如下图：**

图片地址：https://static.oschina.net/uploads/img/201506/24095133\_lexa.png

我们发现，在这个节点上，附件不能上传与下载了，不能删除了。 从表，部分列不能编辑了，还可以删除。部分字段不可以编辑了，说明我们的权限控制成功了。

节点流程表单权限配置

表单权限控制

**下图是节点与节点表单属性控制：**

图片地址：https://static.oschina.net/uploads/img/201506/24095405\_fPpK.png

**是否可以编辑？：**如果设置不能编辑，那么这个表单绑定当前节点上后，就不能编辑了。不能执行保存操作，这种方式不能对局部的表单元素进行控制，如果需要局部控制，需要使用权限控制方案的自定义方案。

**是否可以打印？：**如果可以打印，表单的有上角就多出一个打印按钮，您可以配置一个rtf打印模版，操作员就可以通过这个打印按钮把当前表单的信息按照rtf模版的格式打印出来，如何配置请参考本节点属性=》按钮权限=》打印章节。

**权限控制方案：**这里有两个选择可以选择，默认方案就是表单设计的表单元素权限，这个时候设置为字段、附件、从表设置权限就没有意义。自定义方案，就是把表单元素的权限控制与该节点联系起来，只有选择自定义方案时自定义方案的设置才有效。

**谁是主键？：**这个功能应用于分合流的节点里。

图片地址：https://static.oschina.net/uploads/img/201506/24095635\_5kGs.png

**工作ID是主键：**这是默认的选择项。

流程ID是主键，当当前的节点为子线程时有效，如果填写表单的节点在干流程上，而当前节点是子线程节点，就需要设置为流程ID是主键，才能的打开该表单数据。

**父流程ID是主键：**如果当前流程为子流程，并且该表单的数据采集是从父流程上采集来的，那么就要设置父流程ID是主键，才能把父流程采集的数据打开。

**延续流程ID是主键：**省略，这个场景应用的较少。

**顺序：**一个节点上有可能绑定多个流程表单，用户需要排序，这个顺序就是调整表单的显示的顺序，如下图是调整好的顺序表单。

图片地址：https://static.oschina.net/uploads/img/201506/24095733\_mbvB.png

字段权限控制

对于同一个表单，在不同的节点上需要不同的权限控制时，需要点击自定义方案设置栏目下的字段连接。

图片地址：https://static.oschina.net/uploads/img/201506/24100035\_9H8v.png

字段连接系统弹出一个窗口让你配置参数。

**可见：**就是该字段是否在当前界面可见。

**可用：**就是该字段是否只读。

**是否数字签名：**如果是数字签名，怎该字段就会显示图片或者ca签名。

**检查必填：**如果是必填项，发送之前系统就会判断为空，则不能发送。

**正则表达式：**为该控件设置正则表达式以支持个性化的输入校验控制。

**写入流程数据表？：**相应的节点表单里也要有与字段名称一样的字段，选择后系统就会自动写入NDxxRpt表，依据是该表的主键与NDxxRpt表是一致的。

**[Save]就是保存当前的设置。**

**[Delete All]就是删除全部设置。**

**[Copy From Node] 就是从其他的节点上copy配置**，比如：流程有 A B C D , 四个节点，对于同一个表单BCD 三个字段权限设置方案是一样的，所以我们可以copy过来，省的在一个个的设置了。

附件权限控制

对于同一个表单，在不同的节点上需要不同的权限控制时，需要点击自定义方案设置栏目下的附件连接。

图片地址：https://static.oschina.net/uploads/img/201506/24100616\_MS0P.png

当您点击附件连接，打开一个新界面时候，系统就会把该表单下面所有的附件都列出来，这里只有一个附件。在原始属性栏里，点原始属性，系统就会弹出该附件的原始属性窗口，如果点【编辑】连接系统就会copy一份原始属性让您编辑，如下图所示：

图片地址：https://static.oschina.net/uploads/img/201506/24100645\_7lLC.png

我们把上面图片的底部红色圆圈的标注部分都设置只读的，不可以下载，不可以删除，不可以上传，那么这个附件的在这个节点的权限就会按照当前设置来计算。

当您点击保存关闭后，第2级页面将会被刷新，并且多出一个【删除】连接，如果点击删除，该权限设置的副本将被删除，就会按照原始属性计算，如下图：

图片地址：https://static.oschina.net/uploads/img/201506/24100701\_4j0O.png

这个图表示着我们已经做了权限设置并且生效了。

从表控制权限

对于同一个表单，在不同的节点上需要不同的权限控制时，需要点击自定义方案设置栏目下的从表连接。

图片地址：https://static.oschina.net/uploads/img/201506/24100802\_DFmJ.png

**列表：**列表就会把该表单所有的从表都列出来，目前该表单只有一个从表。

**原始属性：**打开它就可以弹出来从表的属性，在该表单的属性窗口。

**编辑：**就是要编辑当前从表，在当前节点的权限功能控制。如果是第一次编辑，系统将会复制一份明细表的基本属性信息，与从表的字段信息让您编辑。如下图：

图片地址：https://static.oschina.net/uploads/img/201506/24101144\_uwOk.png

当您保存后，系统就会在删除的栏目上多出一个删除连接，就是删除该副本。

图片地址：https://static.oschina.net/uploads/img/201506/24101158\_9HWx.png

**说明：**

第1：权限副本删除后，系统就会按照从表现有的属性计算，否则按照设置的属性计算，上面的图中，已经表示出来权限设置是生效的。

第2：在设置了从表的副本权限后，如果从表增加、删除的列，系统就不能反映到副本中，这个时候需要删除副本在重新建立。

流程数据报表设计

概述

**关键字：**数据查询统计考核报表

使用流程数据报表，首先是您使用了ccBPM的节点表单与流程表单，而非使用sdk表单，这是因为流程报表的设计都是基于流程数据表的，而流程数据表，就是该流程的运行结果存储表。

一条流程从发起到结束，经历发起、运行、退出三个阶段，在流程运行过程中都要对流程数据进行查询统计与分析。

**流程数据是如何产生的？**

在您设计流程时，系统就会自动创建一个流程数据表，这个表来存储流程数据运行的结果，流程的统计分析都是基于这个表进行的，关于流程数据表的描述如下：

1,该表是在创建流程时自动创建的。

2,该表的主键是工作ID，一个流程实例启动起来后，系统就会向这个表里增加一条记录。

3,该表的字段集合是流程系统字段+业务字段组成的。

4,业务字段就是流程设计人员在各个节点设计的节点表单的合集，比如：请假人、请假时间、请假类型。它是所有节点表单字段的合集。

流程系统字段，有如下：工作ID、发起人、发起时间、停留节点、流程参与人、流程状态等等。

流程数据表里有一流程状态字段WFState, 这个字段是系统识别的字段，他也有一个WFSta 字段是查询统计分析用的字段，如下图。

图片地址：https://static.oschina.net/uploads/img/201506/24101457\_vVaF.png

流程数据是流程运行的副产品，是在整个管理活动中最后的结果。我们知道所有的流程数据在流程运行的过程中，都会把数据copy到流程数据表即：NDxxxRpt , (xxx 为转化成int类型的流程编号)。关于运行的表机制结构请参相关章节，报表的查询也是基于这个表的进行设计与查询的。

当一条流程启动起来以后他就产生一个WorkID, 这个int类型的自增长的，全局唯一的。这个WorkID贯穿整个生命周期过程。

报表的设计步骤

进入流程设计器有三个入口，分别是工具栏、画布的右键菜单、与流程的相关属性，如下图

图片地址：https://static.oschina.net/uploads/img/201506/24101608\_ZFr1.png

进入流程报表设计器，如下图：

点击保存并继续按照向导，就可以设计出来一个报表。

图片地址：https://static.oschina.net/uploads/img/201506/24101626\_kASL.png

在下面的章节里，我们一步步的介绍每个设计步骤的属性。

基本信息

如上图所示，首先系统就会生成一个表单ID，他的格式为: ND+转化为int类型的流程编号+MyRpt。比如：流程为001的，那么他的报表编号为 ND1MyRpt，在您设计该流程时，系统就会从流程数据表里按照您的选择重新组合一个新的表单描述，但是数据源仍然是流程数据表。

点保存并继续，进入查询列表选择器。

查询列表字段筛选

选择要呈现报表相关的列，在需要显示的列上打勾，灰色的是必须选择的项目。

图片地址：https://static.oschina.net/uploads/img/201506/24101740\_ppcL.png

**字段区域:**，按照系统字段、枚举字段、普通字段、外键字段(由于该案例中没有外键字段所以就没有显示出来)四部分组成。系统字段显示了所有流程运行相关的字段。其他三类的字段是业务字段，是所有节点表单字段的合集，是ccBPM自动计算出来的，选择要显示字段，然后点保存按钮，进入列的顺序调整阶段。

列表字段显示顺序

在执行查询字段筛选后，系统就会让您开始执行列的显示顺序，如下图。

图片地址：https://static.oschina.net/uploads/img/201506/24101812\_tBbg.png

**移动方向箭头，就可以改变顺序，点保存并继续。**

查询条件设计

查询条件，就是在查询与分析时的条件，凡是外键、枚举、Boolen类型的字段都可以做为查询条件，如果不设置系统默认为部门与年月，为查询条件。

图片地址：https://static.oschina.net/uploads/img/201506/24101846\_EjOR.png

**选中字段，点保存按钮。**

重置默认值

清除以上设计，恢复默认值。恢复默认值后，查询条件为:部门与年月，查询列为流程的基本属性列（流程状态标题操作员部门发起人发起时间参与人跨度(天) 结束人结束时间年月单据文号）。

数据展现的方式调用API

CcBPM对流程运行的数据处理分为流程查询，统计分析，对比分析3大固定的通用的操作模式。这三个操作模式是对以上设计报表的展示。

以下的组件是都是可以配置到被集成系统的菜单里面的，相关的配置连接如下。

|  |  |  |
| --- | --- | --- |
| **组件名称** | **连接** | **说明** |
| 查询组件 | 格式:/WF/Rpt/Search.aspx?RptNo=ND”+int类型的流程编号+”MyRpt&FK\_Flow=流程编号 Demo: /WF/Rpt/Search.aspx?RptNo=ND2MyRpt&FK\_Flow=002 |  |
| 高级查询 | 格式:/WF/Rpt/ SearchAdv.aspx?RptNo=ND”+int类型的流程编号+”MyRpt&FK\_Flow=流程编号 Demo: /WF/Rpt/ SearchAdv.aspx?RptNo=ND2MyRpt&FK\_Flow=002 |  |
| 分组分析 | 格式:/WF/Rpt/Group.aspx?RptNo=ND”+int类型的流程编号+”MyRpt&FK\_Flow=流程编号 Demo: /WF/Rpt/ Group.aspx?RptNo=ND2MyRpt&FK\_Flow=002 |  |
| 交叉报表 | 格式:/WF/Rpt/ D3.aspx?RptNo=ND”+int类型的流程编号+”MyRpt&FK\_Flow=流程编号 Demo: /WF/Rpt/ D3.aspx?RptNo=ND2MyRpt&FK\_Flow=002 | 功能不是很稳定 |
| 对比分析 | 格式:/WF/Rpt/ Contrast.aspx?RptNo=ND”+int类型的流程编号+”MyRpt&FK\_Flow=流程编号 Demo: /WF/Rpt/ Contrast.aspx?RptNo=ND2MyRpt&FK\_Flow=002 | 功能不是很稳定 |

查询表格展现

**说明：**关键字查询条件与外键枚举查询条件是报表设计的。

图片地址：https://static.oschina.net/uploads/img/201506/24102525\_paPI.png

**此组件是对数据结果的查询，是通用的查询方式，有三大部分组成。查询条件，显示区域，翻页区域。**

点击标题列，系统就打开该工作的查看器。

图片地址：https://static.oschina.net/uploads/img/201506/24102536\_ZDBp.png

高级查询

**高级查询**也叫自定义条件查询，用户可以选择指定的字段 ，进行高级的组合条件查询，一个复杂的查询条件被组合后，系统可以保存一个查询模版保存起来，可方便下一次的查询，如下图：

图片地址：https://static.oschina.net/uploads/img/201506/24102740\_cRb1.png

**可以保存多个查询条件模版，这对经常查询分析的应用很有用。**

通过条件的增加与删除按钮，就可以增加删除查询条件，根据不同类型的字段，系统会让您选择不同的操作符号，不同的查询值，如下图。

图片地址：https://static.oschina.net/uploads/img/201506/24102800\_6wnl.png

**导出：**把查询的结果导出到excel.

数据分组分析

数据表格分组分析

数据表格的分组分析，有如下几部分组成。

**查询条件**：在报表设计器中定义的，按照关键字、与指定的外键枚举查询。

**显示内容**：只要是外键或者枚举类型的字段，都自动排列在显示内容里。

**分析项目**：只要是数值类型的数据，都可以显示在分析项目里，对与一个分析项目可以求合计、求最大、求最小。

**图表**：显示图表的选项，只有显示内容选择一个时候，图表才可有效。

图片地址：https://static.oschina.net/uploads/img/201506/24102946\_ZHYh.png

点详细信息就是对数据的挖掘分析，比如我要看到每个数值背后的详细信息，就需要打开【详细】按钮查看，如下图：

图片地址：https://static.oschina.net/uploads/img/201506/24102958\_fASb.png

如上图所示，要看到请假了两天，要知道是那个请假流程请的假，就可以挖掘出来。

点数值列连接，就可以对这个列进行排序，如下图：

图片地址：https://static.oschina.net/uploads/img/201506/24103022\_Jc2Q.png

特别说明只有分析内容选择一个维度的时候，图表才有意义，如下图：

图片地址：https://static.oschina.net/uploads/img/201506/24103035\_Levt.png

下面章节展示了，三种常用的图形分析效果。

**导出:** 系统就导出到Excel.

图片地址：https://static.oschina.net/uploads/img/201506/24103057\_S7hW.png

柱状图数据分析

图片地址：https://static.oschina.net/uploads/img/201506/24103221\_53Os.png

此类图形形象的多用于分析数据的多少。

饼状图分析

图片地址：https://static.oschina.net/uploads/img/201506/24103312\_rK9h.png

此类图形形象的多用于分析数据的组成部分。

折线图分析

图片地址：https://static.oschina.net/uploads/img/201506/24103359\_pQGr.png

此类图形形象的多用于分析数据增长变化以及数据的波动情况。

交叉报表

**定义：** 就是在一定的查询条件下，选择两个分析维度，这两个分析维度就是报表的外键或者字段，选择一个数值类型的分析项目，对这个数值进行求合，求平均，求最大的函数计算，如下图：

图片地址：https://static.oschina.net/uploads/img/201506/24103436\_HHyr.png

**说明：**选项功能的，显示百分比、显示图形，没有开发完成。

对比分析

省略-此部分没有经过严格测试。

流程设计

概述

**流程设计：**零代码模式开发，也叫配置模式开发，它是通过配置的方式实现流程的开发。它相对的开发是sdk模式的开发，也叫api模式开发，它是为高级开发人员准备的。

本章节将讲解零代码流程开发的基础知识。讲解线性、同表单分合流、异表单分合流的模式开发。零代码模式开发，是我们所提倡的一种开发模式。流程的设计人员不需要代码，只需要配置的模式进行开发。它需要用到ccform与驰骋工作流的丰富的节点属性与流程属性来实现。

我们已经在流程引擎的功能说明章节里详细的阐述了流程属性与节点属性，但是要成为两个真正的骋工作流开发高手还需要，把这些属性灵活的运用。把复杂的客户应用灵活的变通用骋工作流实现。

只有记住这些功能，多练、多想、多总结，你才能成为骋工作流高手。

请假流程设计

使用自由表单设计

**步骤1：画流程**：

图片地址：https://static.oschina.net/uploads/img/201506/24104521\_TvCH.png

**步骤2：绑定岗位**

图片地址：https://static.oschina.net/uploads/img/201506/24104544\_N64E.png

**步骤3：设计表单**

**填写申请单：**

图片地址：https://static.oschina.net/uploads/img/201507/06160427\_4raC.png

**部门经理审批：**

图片地址：https://static.oschina.net/uploads/img/201507/06160445\_9pbH.png

**总经理审批：**

图片地址：https://static.oschina.net/uploads/img/201507/06160500\_UKc7.png

**步骤4： 检查并运行**

**流程设计完成以后，点击检查按钮：**

图片地址：https://static.oschina.net/uploads/img/201507/06160519\_X5Do.png

**步骤5：设计报表统计分析**

**点击 流程属性---》选择报表统计分析**

图片地址：https://static.oschina.net/uploads/img/201507/06160542\_yMFN.png

**根据步骤进行报表设计：**

图片地址：https://static.oschina.net/uploads/img/201507/06160600\_6JXy.png

使用审核组件实现

**步骤1：设计流程**

同上

**步骤2：绑定人员、岗位、部门**

同上

**步骤3：设计表单，在表单中添加审核组件**

图片地址：https://static.oschina.net/uploads/img/201507/06160758\_XBRz.png

**步骤4：在其他流程节点中，设置审核组件为开启**

效果图：

图片地址：https://static.oschina.net/uploads/img/201506/24105440\_OU5I.png

审核后效果：

图片地址：https://static.oschina.net/uploads/img/201507/06160826\_e9ea.png

**审核组件也可以使用在SDK表单、自定义表单中.**

使用自定义表单实现

**定义概述：**一个已经做好的表单需要绑定到节点上。

**自定义表单工作模式：**流程控制按钮区域是ccflow来完成，表单区域是放在控制区域下面的框架里。

图片地址：https://static.oschina.net/uploads/img/201512/29183122\_aKyB.png

**注意:**框架里面的界面是一个url，这个就是自定义表单。

**应用背景：**这种类型是解决我自己已经有了一个成熟稳定的产品或者项目，里面的表单已经做好了，我只想使用流程引擎负责流转功能，业务数据的存储不变化。这中类型有如下特点。

A,数据的存取有自己开发人员来确定。

B,仅仅使用ccflow的流程引擎。

C,最大限度的减少程序开发人员的开发。

D,最大完整的保障自己的业务独立与业务完整。

**如果要适应这种方式需要在把自己的表单做如下改造：**

A,把自己的表单主表增加一个int类型的workid列用于做主键或者用程序保障他的唯一性。

B,自己开发的表单，要接受来自ccflow的参数，参数格式如下。比如：您输入的是 http://localhost/MySelf.aspx?aa=xxx 系统在调用时就会是http://localhost/MySelf.aspx?aa=xxx&WorkID=123&FK\_Node=101&UserNo=zhoupeng&SID=xxxxx，这些参数是系统自定传递的，它表示那个流程实例、到达了那个节点、那个流程，与当前的登录人员，SID 是验证该人员是否合法，请参考其他章节。开发人员可以根据节点编号灵活控制权限。

如果您了解了ccflow的数据存储结构（请参考流程属性章节），我们建议您把该流程的流程属性，数据存储表名与您的业务表单表名一致。

图片地址：https://static.oschina.net/uploads/img/201512/29183226\_IwS7.png

**这样做的好处有**：

1,不用担心数据存储主键，不用关心流程的业务逻辑。

2,系统会给该业务字段自动创建其他的流程字段，用户可以根据这些字段信息扩展更多的业务逻辑编写。

**实现步骤：**

通过编写代码完成表单页面，表单中必须有一个保存函数Save()，在执行发送或指定保存时，ccflow会自动调用这个方法。在用户点击保存与发送按钮后，系统就会自动触发了框架里面的save方法，完成框架内的数据保存，如果框架内的函数返回false,标识发送失败，系统就停止发送。

**1.图例：**

图片地址：https://static.oschina.net/uploads/img/201512/29183242\_BX5T.png

**2.配置节点属性，在节点属性表单栏选择中，选择自定义表单。**

图片地址：https://static.oschina.net/uploads/img/201512/29183257\_lXSX.png

**3.启用调试运行**

成功后，出现信息提示。

图片地址：https://static.oschina.net/uploads/img/201512/29183309\_e19f.png

多人会签流程设计

线性流程设置

在ccflow中，我们可以通过配置，就可以实现会签或者多人审批等操作。

首先，要在需要会签或者多人审批的节点设置工作处理人。

图片地址：https://static.oschina.net/uploads/img/201512/29182553\_zJMR.png

然后，再给这个节点设置运行模式（或者成为节点处理模式），通过节点属性—运行

图片地址：https://static.oschina.net/uploads/img/201512/29182617\_xePv.png

模式，选择协作模式或者队列模式即可。

由此两步，即可实现多人会签或者多人审批。

分合流（多人、多部门分工处理）的设计

**分合流**：顾名思义，想长江一样，有主流河流和分支河流，有起点，有终点。

**适用范围**：多人、多角色、多岗位、多部门等相互独立审批工作、处理工作，同时又可在某个步骤中查看工作的汇总、或者过程、或者处理结果等。

**作用**：可以实现复杂的业务逻辑审批、工作处理，提高工作效率，提高多部门的协同办公的能力。

**分合流设置的两种方式：**

同表单分合流和异表单分合流。

**同表单分合流**：操作同一个表，一个工作多人处理。

图片地址：https://static.oschina.net/uploads/img/201512/29182736\_oIOq.png

上图中，成员填写计划、修改计划为子线程节点，下达填写计划为分流节点，汇总整理计划、计划汇总是合流节点。

**通过以下方式设置节点类型。如果使用分合流，节点类型必须设置。**

图片地址：https://static.oschina.net/uploads/img/201512/29182756\_2B4N.png

子线程节点中，根据绑定的人员和设置的节点运行规则决定有多少人处理。

合流节点，根据绑定的人员、设置的节点运行规则和运行模式决定有多少人处理。

**异表单分合流：可以操作不通的数据表，不同的工作。**

图片地址：https://static.oschina.net/uploads/img/201512/29182815\_3kAW.png

**除了分流节点、合流节点记忆结束节点，其他的子线程节点都表单一个单独的数据表（工作任务）。**

图片地址：https://static.oschina.net/uploads/img/201512/29182834\_91GU.png

同表单分合流程的设计

**关键字：**同表单分合流.

**定义：**同表单分合流就是，多人可以同时处理一个工作，这些人处理的工作内容是相同的，也就是表单是相同的，我们就成为同表单的分合流。在它的流程引擎里，也叫会签。

**我们以月销售总结为例，来说明什么是同表单的分合流。**

**需求描述：**一个销售经理，需要本部门的销售人员，填写汇报上月的销售工作。它有三个步骤，下达任务，各个销售人员填写，销售经理汇总查看。

**说明：**汇总上来本月销售情况，每个销售人员填写的销售单据都是相同的，每个销售人员都是并行填写的。我们把这样的模式，成为同表单的分合流。

**第1步骤, 画出流程图**

图片地址：https://static.oschina.net/uploads/img/201506/24111207\_jfTu.png

**设置步骤：在节点1设置节点类型为分流，节点3设置节点类型为合流。**

图片地址：https://static.oschina.net/uploads/img/201507/08142511\_Y0W3.png

**第2步骤,把子线程的数据汇总到合流节点表单中去**

图片地址：https://static.oschina.net/uploads/img/201506/24111255\_zpQb.png

对于分合流程来说，一般的模式就是分流–子线程–合流。通常是分流点与合流点是同一个人处理。比如：事例中的部门负责人启动分流让本部门的成员填写同一格式的工作日志，然后汇总到部门经理身上。

现实的业务需求是合流点要看到每个子线程的工作内容，如何处理把子线程的数据汇总到合流节点表单中去？

**CCBPM为您提供如下两种方案：**

1，使用ccBPM内部的约定规则：在合流点上创建一个从表，从表的字段与子线程节点表单字段对应，这样才确保可以复制，在合流节点表单上的从表的属性里设置是分流节点汇总表. 这样的话每一个字线程完成后就向这个从表中插入一条数据，并把子线程节点表单的数据按照字段对应关系copy到从表的一行记录中。(在案例库中有这个事例)

2, 利用子线程的节点事件：在了线程发送成功与撤消的两个节点事件中向指定的表写入数据，一般是向合流节点表单写入，以满足您的特殊要求。在发送成功时插入数据，在撤消时删除数据。

**第一种方案适用于大众情况，第二种方案需要写存储过程可以处理复杂的业务逻辑。**

**第3步骤,把子线程从表数据汇总到合流从表数据上去**

**关键字**：CCBPM合流数据汇总合流从表数据汇总。

**应用场景**：标准制修订流程，步骤说明：请见流程图。

图片地址：https://static.oschina.net/uploads/img/201506/24111357\_MFLA.png

子线程从表属性默认值不需要特殊的设置

图片地址：https://static.oschina.net/uploads/img/201506/24111418\_aMtQ.png

**第4步骤,关于合流点退回、终止、查看进度的问题处理办法**

**需求背景**：

在分合流流程中，对于合流点或者分合流点，操作员要看到上一个子线程进展情况，可以根据需要退回、或者终止指定的子线程。

**实现方法1**：在合流节点上启用子线程按钮，就可以。

**实现方法2**:对于自由表单使用者来说，您可以在表单上增加一个连接或者，一个按钮如下图。

图片地址：https://static.oschina.net/uploads/img/201506/24111502\_AVY8.png

只要在表单上进入： HeLiuDtl.aspx?WorkID=xxxxxxxx 功能界面，就有对该合流节点上的子线程的操作。或者打开流程：**D:ccflowtrunkCCFlowDataFlowDemoFlow02.同表单分合流同表单分合流(二).xml**模板装载上在第3个节点表单中可以看到。在节点属性中，如果您启用了强制删除子流程功能，那么在这个合流点上会出现终止子线程的功能。

异表单流程的设计

**画出流程图：**

图片地址：https://static.oschina.net/uploads/img/201506/24111636\_nuIL.png

设计步骤大多与同表单分合流相同。

父子流程的设计

定义

一个流程A的一个节点，因工作的需要调起另外的流程B，A就叫父流程，B就叫子流程。如果流程B的一个节点在调用流程C,那么B相对于C来说就叫父流程，C叫子流程。所以是子流程还是父流程来说都是相对的。

一个流程A在一个节点上可以调用n个子流程。

在父流程的吊起子流程的节点，我们称之为吊起节点。它所打开的第一个界面就是开始节点的第一个节点表单界面，通常需要把吊起节点表单所有的数据copy到子流程的第一个节点表单上去。

停留在父流程节点上的工作人员，必须具备发起子流程的权限。

如果一个流程不能被单独发起，只能被其它流程所调用发起，可以在流程属性里设置，让其不显示在发起流程列表里。

图片地址：https://static.oschina.net/uploads/img/201506/24111756\_SNcl.png

调用子流程方法

手工设置连接：在表单中增加一个Link ，指向工作发起界面,事例：

MyFlow.aspx?FK\_Flow=012&PWorkID=@OID&PNodeID=1103&PFlowNo=011

以上事例说明了，在节点1103上启动流程012。父流程的ID用@OID 获取。

**调用子流程的参数列表：**

|  |  |  |
| --- | --- | --- |
| **参数名** | **说明** | **备注** |
| PFlowNo | 当前流程编号 | 必须 |
| PNodeID | 当前工作节点 | 必须 |
| PWorkID | 当前节点的工作ID | 必须 |
| FK\_Flow | 被调用的流程编号 | 必须 |
| FK\_Node | 被调用的流程的开始节点 | 可选 |
| JumpToNode | 要跳转到的节点（如果需要跳转到指定的节点上的话） | 可选 |
| JumpToEmp | 要转到的人员（如果需要跳转到指定的人员的话） | 可选 |
| 其他字段参数 | 如果要传递其它的字段参数可以用&字段名=字段值的格式传递。 | 可选 |

图片地址：https://static.oschina.net/uploads/img/201506/24112228\_ll7R.png

重要的属性

**流程属性：**

当最后一个子流程完成后，用户希望父流程的调用节点可以自动向下运动。

图片地址：https://static.oschina.net/uploads/img/201507/08142840\_BVm8.png

该属性启用后，最后一个子流程完成后，系统就会让父流程的吊起节点自动运动到下一步骤上去。

开发注意事项

1, 唤起节点的表单数据可以copy 到被唤起流程的第一个节点上.

2, 如果以前已经唤起过的流程没有运行下去,那么这个调用就会把上次没有发起的子流程掉起来.

3, 如果没有的话就可以看到已经发起的流程列出来.

如果要实现自动转向，或者带有条件的转向，子流程您需要查看节点属性，如何配置它，请参考节点属性。

4，如果您需要对数据进行二次处理，或者调用子流程进行其它的处理，您可以做一个中间处理页面，把处理的业务逻辑放在这个页面上。

流程事件和节点事件

概述

**关键字：**流程事件，节点事件。

**描述：**流程在运动过程中，它是按照一定的时序进行的，所以ccBPM根据每个时序定义了一些事件，比如节点发送前、发送成功时、发送失败时、节点退回前、退回后。比如：流程删除前、删除后、结束前、结束后，流程开发人员可以在相关的事件里填写自己的业务逻辑。

**打开节点属性点击事件标签：**

图片地址：https://static.oschina.net/uploads/img/201507/08142938\_Of6R.png

**实现业务逻辑的方法：**ccBPM通过两种方式实现事件，分别是：配置的方法与事件基类实体。

配置的方法比较简单，可以执行多种方式，适合维护人员。

事件基类适合程序开发人员，编写事件基类可以非常灵活、执行效率非常高效。对于有开发能里的朋友我们推荐使用后者，在下面章节，我们将详细讲述如何使用它们实现您的业务逻辑，实现与您的系统交互。

编写流程事件实体

**关键字：**ccBPM 流程事件实体 FEE

**描述：**流程事件实体基类(BP.FlowEvent.Base)是ccBPM为二次开发人员提供的可以重写方法的基类。二次开发人员需要继承该基类并实现基类的业务逻辑就可以实现复杂的业务处理。

**应用案例1：**我们打开ccBPM的demo流程中的财务报销流程，在表单里我需要对明细表求和，并把明细表的合计放入主表的一个字段里，如下图：

图片地址：https://static.oschina.net/uploads/img/201506/24112720\_4Xp0.png

**第一步：**我们编写一个FEE类，此类从FlowEventBase继承下来，重写SaveAfter方法。

图片地址：https://static.oschina.net/uploads/img/201506/24112732\_t4CK.png

业务逻辑方法如下：

图片地址：https://static.oschina.net/uploads/img/201506/24112743\_7l1g.png

**第2步：**执行类的编译。这里特别说明，你编写的类库一定要以BP.\*开头，否则无效。

**第3步：**把流程标记填写到流程属性的流程标记里。

图片地址：https://static.oschina.net/uploads/img/201507/08143051\_GMwc.png

其他：注意流程基类，已经为开发者提供丰富的属性，您可以根据这些属性灵活的实现您的业务逻辑。

配置ccBPM事件

**配置ccBPM的事件，我们知道配置ccBPM事件是实现ccBPM复杂的业务逻辑**

|  |  |
| --- | --- |
| **配置项** | **说明** |
| 禁用 | 不实用，无配置，让配置的不起作用。 |
| 使用SQL | 编写SQL语句，请在执行内容里填写SQL，注意表达式支持变量。 如果有多条sql ，可以用@GO 分开。比如： DELETE WF\_Emp WHERE 1=2 @GO DELETE WF\_Emp WHERE 2=3 如果要执行存储过程就要写: EXEC YourProName @OID |
| 执行存储过程 | 例如：EXEC YourProName @OID |

请在执行内容里填写一个存储过程名称，注意表达式支持变量。

执行SQL:

@OID 是一个当前的workid 参数。

如下图：

图片地址：https://static.oschina.net/uploads/img/201507/08143145\_hew4.png

通过使用事件基类来书写业务逻辑

我们将要取消这个方法，已经使用起来的

我们知道ccBPM的流程引擎是用事件与外部系统进行交互的，如何使用ccBPM的base基类来满足业务流程的逻辑实现。

**应用案例：**计算明细表的一列合计，并把它放在表单的主表一个字段里，这个用表单保存后事件来完成。请参考ccBPM的demo流程001,财务报销流程。

**第一步：**编写一个事件类，这个类从BP.Sys.EventBase 集成下来，实现Do 方法。

图片地址：https://static.oschina.net/uploads/img/201506/24113235\_5TMf.png

**第二步**：在开始节点表单里点击傻瓜表单，点事件。

图片地址：https://static.oschina.net/uploads/img/201506/24113249\_g8kQ.png

打开事件编辑窗口：

图片地址：https://static.oschina.net/uploads/img/201507/08143243\_meua.png

在执行内容里填写事件类名：BP.FlowEvent.F001。

**第三步实现效果：**

图片地址：https://static.oschina.net/uploads/img/201506/24113326\_BlQV.png

**关于事件基类的详细说明：**

固定参数与系统参数，请参考：

D:ccflowtrunkComponentsBP.En30EventBaseEventBase.cs

**其他的参数包括两部分:**

1：来自表单里面的所有字段数据。

2：如果是url就是所有通过url传递过来的参数。

3: 在类里面这些参数都可以通过this.GetValInt, or this.GetValStr 获得。

4: 总之可以在Do方法里获得丰富的参数。

5: 通过事件类型与，FK\_Node 可以把多个类的方法集成在一起，没有必要，每个事件都做一个class文件。

ccBPM的事件里执行URL与执行类的区别

1，URL 编写简单方便，可以支持多语言。

2，url 执行效率底下，不允许使用cookies，与session.

3, url 不允许访问webuser.\* 的变量。

4, url 在表单装载前事件里，如果效率底会导致表单加载错误.

5，使用Url如果执行时间长，会导致超时的错误。

使用类的优点:

1, 执行效率高，编写容易，可以访问webuser.\*的变量，与session,cookies.

2, 该事件所必须放入BP.\*.dll类库才能执行.

3, 不会超时，并执行效率高。

流程变更处理方案与对策

**很多的朋友都问这个问题，为了方便解决此疑问，面对以上的变化,ccBPM采取的应对方案如下.**

**1,增加节点.**

RE:解决方案:对流程没有影响.  
  
1.对原来与现有的流程没有什么影响，历史的工作报告还能打开.流程轨迹图可能会出现不连续的情况,但是不影响查看.  
  
2,在途工作会按照新的设计去运行.

**2,减少节点.**

**影响范围:**

1,历史数据会不能正常打开，被删除的节点数据反映不出来.  
  
2,历史轨迹图不能被正确的显示.  
  
3,在途工作会出现错误.

RE:解决方案.

在原来的基础上改造:\*\*  
  
 1,如果历史数据不多，不是很重要.可以考虑,在原来基础上修改.  
  
 2,采用节点的生命周期的模式处理.将要禁用的节点.也就是说节点不删除,而是改变它的生命周期.让其达到不可用的目的.  
  
 \*\*废止当前流程,新建新的流程发布:\*\*  
  
 1,复制当前的流程，做为新的流程发布.  
  
 2,禁用当前流程.  
  
 \*\* 建议采取后者方案. \*\*

**3,节点工作人员发生变化.**

RE:对流程运行没有影响，对历史数据没有影响.

**4,表单内容增加.**

RE:对流程运行没有影响，对历史数据没有影响.

**5,表单内容减少.**

RE:把减少的字段隐藏起来. 对流程运行没有影响，对历史数据没有影响.

**6,方向条件变化.**

RE:对流程运行没有影响，对历史数据没有影响.

事件类处理业务逻辑的命名规则约定

在应用程序类库里面创建一个FlowEvent目录。

类的命名空间为 BP.FlowEvent.Fxxxx , xxx 标识流程编号。

一个流程的业务逻辑写入到一个类里，不同的节点，不同事件的逻辑用节点编号与事件的标记区分出来。

详细请参考。D:ccflowvalue-addedDemoAndTestingBP.DemoEventBaseF001.cs

图片地址：https://static.oschina.net/uploads/img/201506/24113847\_VXqb.png

如何设置邮件模板格式

请打开D:ccflowtrunkCCFlowDataUserEmailTemplete

模版文件里您可以定义字段变量。比如: @+字段名.

如何使用驰骋工作流程引擎在分合流中汇总附件数据

**应用背景：**

在一个工程中，经理下发任务后，需要每个接到任务的人，写一份报告上传，报告给经理。这种情况下，就需要将所有上传的附件，汇总呈现给经理阅览。

**同表单分合流的设计方法：**

设置前提：子线程节点与合流节点中，必须有上传附件的控件。

在合流节点中，设置附件属性中的权限控制，如下图：

图片地址：https://static.oschina.net/uploads/img/201510/22174308\_WGUC.png

**异表单分合流的设计方法：**

只需要在子线程节点与合流节点中添加附件控件，控件名称相同即可实现。

**附件汇总的效果：**

图片地址：https://static.oschina.net/uploads/img/201510/22174321\_SVGl.png

高级开发SDK模式开发

概述

**关键字：**sdk模式开发

**说明：**Sdk模式开发是ccBPM开发者秘笈武器，它可以实现复杂的业务需求，实现各式各样的呈现风格。Sdk模式开发也叫api开发。驰骋工作流引擎开发团队将所有与流程引擎的交互，都通过api提供的。比如：菜单类的api，获取发起流程列表、获取待办列表、获取未完成流程列表、获取抄送列表。功能类的：比如发送、退回、移交、加签。

在零代码开发模式下，流程引擎就是调用这些api来实现用户的需求，您可以跟踪这些页面代码可以看到他们。

学习sdk模式的开发是很简单的，驰骋开发团队为您提供了丰富的demo。CCBPM的sdk模式开发概要介绍。

**CcBPM的所有的功能基本上都是调用ccBPM的api来完成的。**

菜单部分：获取发起流程列表、待办、在途、抄送。这些菜单的形成是通过ccBPM的api获得一个数据源然后把他们呈现出来。

在流程的操作过程中：流程的发送、退回、转发、移交、删除、撤销、加签也是通过一定的参数调用ccBPM的api完成的。

**这些api都存放在:BP.WF. Dev2Interface里面。**

图片地址：https://static.oschina.net/uploads/img/201506/24140301\_hqez.png

如果开发者想摆脱ccBPM已经开发好的模式依赖，灵活的实现客户的需求，那就需要自己编写代码，调用底层的api实现，这就是ccBPM的sdk开发模式也称为api的开发模式。

**什么情况下用户sdk模式的开发？**

如下情况可用到sdk的开发模式：

  表单超级复杂ccform满足不了用户的应用需求，也不能使用嵌入方式的表单实现。

  业务逻辑超级复杂，ccBPM的事件也不能满足用户的需要。

功能调用

说明概要

1, 您的系统与ccBPM交互是通过Url 与参数约定的，比如：要在您的系统里登陆ccBPM里面的某些功能。就需要按照一定的约定规则执行。

2，参数分为基本参数、必用参数与可用参数，基本参数有三个 UserNo,(用户编号) SID(登陆ID号). DoWhat(执行标记). 比用参数是根据DoWhat 来确定的，比如您要启动一个流程就需要FK\_Flow这个参数。可用参数是根据。

3，根据您需要的功能，查询参数列表。

访问位置: http://yourServerIP/ccFlow/WF/Port.aspx .

集成案例效果

**更多的集成案例效果图，请参考: D:ccflowdoc图片介绍集成效果图**

图片地址：https://static.oschina.net/uploads/img/201506/24140519\_W2UW.png

图片地址：https://static.oschina.net/uploads/img/201506/24140526\_Ov8s.png

功能列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **名称** | **可选参数** | **说明** | **内容说明** |
| Login | 登录 | 无 | Port.aspx?UserNo=001 &SID=123323 &DoWhat=Login | 系统登录 |
| LoginSmall | 精简界面 | 无 | Port.aspx?UserNo=001 &SID=123323 &DoWhat=LoginSmall | 同上 |
| Start | 发起 | 无 | Port.aspx?UserNo=001 &SID=123 &DoWhat=Start | 调用发起工作界面，显示当前用户能够发起的工作列表 |
| StartSmall | 发起指定的流程 | FK\_Flow | Port.aspx?UserNo=001 &SID=123 &DoWhat= StartSmall   &FK\_Flow=001 | 调用简洁的流程发起界面，没有头部与尾部。 |
| Emps | 通讯录 | 无 | Port.aspx?UserNo=001 &SID=123323 &DoWhat=Emp | 调用通讯录 |
| EmpsSmall | 精简界面 | 无 | Port.aspx?UserNo=001 &SID=123323 &DoWhat=EmpSmall | 调用通讯录 |
| EmpWorks | 待办工作 | FK\_Flow（可选） | Port.aspx?UserNo=001 &SID=123 &DoWhat=EmpWorks | 打开我的工作列表。 没有FK\_Flow列出全部，有就列出指定的流程。 |
| EmpWorksSmall | 待办工作（精简界面） | 同上 | Port.aspx?UserNo=001 &SID=123 &DoWhat=EmpWorksSmall | 同上 |
| Runing | 在途工作 | FK\_Flow（可选） | Port.aspx?UserNo=001 &SID=123 &DoWhat=Runing | 打开我的在途工作列表。 没有FK\_Flow列出全部，有就列出指定的流程 |
| RuningSmall | 在途工作（精简界面） | 同上 | 同上 | 同上 |
| FlowSearch | 流程查询 | FK\_Flow (可选) Key查询关键字(可选). 流程表里外键、枚举字段(可选)。 |  | 调用流程数据查询功能。 如果FK\_Flow为空就调用流程查询主界面。 否则调用指定流程编号的流程。 在调用指定流程查询功能时， 可以传入按照关键字查的参数Key, 与外键，枚举值指定的参数，作为查询条件。 |
| FlowSearchSmall | 精简界面 | 同上 | 同上标记修改成FlowSearchSmall | 同上 |
| FlowFX | 流程分析 | 同上 | 同上 | 同上 |
| DealWork | 处理指定的工作 | FK\_Flow WorkID 字段值(可选) | Port.aspx?UserNo=001 &SID=123 &DoWhat=DealWork &FK\_Flow=001 &WorkID=10001 &字段名称=字段值 | 处理工作ID为10001的工作。 字段值参数是可选的。 |
| DealWorkInSmall | 简洁窗口 | 同上 | 同上 | 同上 |

常用的测试案例

**--调用待办工作.**

http://127.0.0.1/ccFlow/WF/Port.aspx?UserNo=hq&SID=123&DoWhat=EmpWorks&FK\_Flow=006

**--调用发起工作主要界面.**

http://127.0.0.1/ccFlow/WF/Port.aspx?UserNo=hq&SID=123&DoWhat=Start

**--发起指定的流程. 没有参数.**

http://127.0.0.1/ccFlow/WF/Port.aspx?UserNo=hq&SID=123&DoWhat=Start&FK\_Flow=006

**--发起指定的流程. 并且传入参数，比如目的地字段. 也可以是合同编号......**

http://127.0.0.1/ccFlow/WF/Port.aspx?UserNo=hq&SID=123&DoWhat=Start&FK\_Flow=006&mudedi=shandongjinan

**--发起指定的流程(简洁页面).**

http://127.0.0.1/ccFlow/WF/Port.aspx?UserNo=hq&SID=123&DoWhat=StartSmall&FK\_Flow=006

**--发起指定的流程(简洁页面). 并且传入参数，比如目的地字段. 也可以是合同编号。**

http://127.0.0.1/ccFlow/WF/Port.aspx?UserNo=hq&SID=123&DoWhat=StartSmall&FK\_Flow=006&mudedi=shandongjinan

**--调用流程查询无参数。**

http://127.0.0.1/ccFlow/WF/Port.aspx?UserNo=hq&SID=123&FK\_Flow=006&DoWhat=FlowSearch

**--调用流程查询带有关键字的。**

http://127.0.0.1/ccFlow/WF/Port.aspx?UserNo=hq&SID=123&FK\_Flow=006&DoWhat=FlowSearch&Key=11233

**--调用流程分析.**

http://127.0.0.1/ccFlow/WF/Port.aspx?UserNo=hq&SID=123&DoWhat=FlowFX&FK\_Flow=016

**--处理指定的工作，**

http://127.0.0.1/ccFlow/WF/Port.aspx?UserNo=hq&SID=123&DoWhat=DealWork&FK\_Flow=006&WorkID=48

**--处理指定的工作，带有参数**

http://127.0.0.1/ccFlow/WF/Port.aspx?UserNo=hq&SID=123&DoWhat=DealWork&FK\_Flow=006&WorkID=48&mudedi=jinan

**--处理指定的工作(利用简洁界面)，**

http://127.0.0.1/ccFlow/WF/Port.aspx?UserNo=hq&SID=123&DoWhat=DealWorkInSmall&FK\_Flow=006&WorkID=48

**--处理指定的工作(利用简洁界面)，带有参数**

http://127.0.0.1/ccFlow/WF/Port.aspx?UserNo=hq&SID=123&DoWhat=DealWorkInSmall&FK\_Flow=006&WorkID=48&mudedi=jinan

调用ccBPM的功能API

**Fk\_flow:流程编号**

**userNo、BP.Web.WebUser.No:登录帐号**

**Fk\_node:节点编号**

**Workid:工作ID**

**FID：父流程ID**

**1.系统登录**

a)前台登录：BP.WF.Dev2Interface.Port\_Login(userNo);

b)登录流程设计器：/WF/Admin/XAP/Designer.aspx

c)自动登录流程设计器：/WF/Admin/XAP/Designer.aspx？UserNo=admin&SID=1234556.

SID需要读取数据库中，对应admin的SID。

**2.当前登陆人可以发起的流程（发起）**

Dt=BP.WF.Dev2Interface.DB\_GenerCanStartFlowsOfDataTable(BP.Web.WebUser.No);

**3.当前登陆人可以处理的工作（待办）**

Dt=BP.WF.Dev2Interface.DB*GenerEmpWorksOfDataTable(userNo,FK*Flow);

Dt=BP.WF.Dev2Interface.DB\_GenerEmpWorksOfDataTable();

**4.当前登陆人处理过的流程（在途）**

Dt=BP.WF.Dev2Interface.DB*GenerRuning(userNo,FK*Flow);

Dt=BP.WF.Dev2Interface.DB\_GenerRuning();

**5.当前人收到的抄送信息（阅知件）**

Dt=BP.WF.Dev2Interface.DB\_CCList(userNo);

**6.创建空白的WorkID:**

Int64 workid=BP.WF.Dev2Interface.Node*CreateBlankWork(fk*flow);

CCBPM的工作ID是一个int64位的证书，始终是按照顺序号+1产生的。该值全局唯一，并没有重复性，该信息记录到Sys\_Serial表中,Workid的生成冲100开始。

**7.执行流程发送：**

//执行发送

BP.WF.SendReturnObjs objs=BP.WF.Dev2Interface.Node*SendWork(fk*flow,workid);

//流程是否结束

Bool isFlowOver=objs.IsStopFlow;

//下一步节点ID

Int toNodeID=objs.VarToNodeID;

//下一步节点名称

String toNodeName=objs.VarToNodeName;

//下一步处理人ID

String toEmpID=objs.VarAcceptersID;

//下一步处理人名称

String toEmpName=objs.VarAcceptersName;

//发送后信息

String msgInfo=objs.ToMsgHtml();

发送说明：发送就是让节点向下运动。调用接口执行发送后，返回一个执行结果的对象SendReturnObjs，该对象是流程引擎执行过程中的编号，解析该变量，可以检查流程是否完成、晕倒到哪一个节点、一下节点由谁来执行工作。他的流向和接受人是根据流程设计的规则而定的。

Objs=BP.WF.Dev2Interface.Node*SendWork(fk*flow,workid,fk\_node,userNo);

如果程序员知道下一步要送给谁、到达哪一步节点，可以调用上面的接口，该接口就会摆脱流程引擎的节点运行规则，自定义运行。

**8.撤销发送：**

String msg=BP.WF.Dev2Interface.Flow*DoUnSend(fk*flow,workid);

撤销是发送的逆向操作。能否被撤销，是根据当前活动节点的撤销规则决定的。

**9.流程回滚：**

String msg=BP.WF.Dev2Interface.Flow*DoRebackWorkFlow(fk*flow,worjid,fk\_node,’回滚’);

回滚，是在流程结束后需要重新在指定的节点，让指定的人员重新向下运行。与撤销不同的是，回滚是在流程完成以后的操作，并且由管理员操作。

**10.退回：**

msg=BP.WF.Dev2Interface.Node*ReturnWork(fk*flow,workid,FID,fk\_Node,toNodeID,toemp,’原因’,’是否原路返回’);

**11.移交：**

BP.WF.Dev2Interface.Node*Shift(fk*flow,fk\_node,workid,FID,toemp,’原因’);

移交就是把自己所要做的工作交给其他人处理。

**12.加签：**

//A将工作加签给B后，B处理完成后，会发送给A，由A发送到下一步。

msg=BP.WF.Dev2Interface.Node\_Askfor(workid,BP.WF.AskforHelpSta.AfterDealSendByWorder,toemp,’原因’);

//A将工作加签给B后，B处理完成后，直接发送到下一步。

Msg=BP.WF.Dev2Interface.Node\_Askfor(workid,BP.WF.AskforHelpSta.AskDealSend,toemp,’原因’);

//A将工作加签给B后，B回复加签

Msg=BP.WF.Dev2Interface.Node*AskforReply(fj*flow,fk\_node,workid,fid,’原因’);

**加签说明：**

加签就是把自己所要做的工作参考其他人意见，或者让其他人处理。它分为两种模式：1.加签后，由加签人发送到下一步。2.加签后，由加签人发给当前人，由当前人发送到下一步。

**13.结束流程：**

//点击结束按钮，结束流程，这种方式会记录日志

Info=BP.WF.Dev2Interface.Flow*DoFlowOver(fk*flow,workid,’原因’);

//按照标记删除流程

Info=BP.WF.Dev2Interface.Flow*DoDeleteFlowByFlag(fk*flow,workid,’原因’，true);

//彻底的删除流程，无日志记录

Info=BP.WF.Dev2Interface.Flow*DoDeleteFlowByReal(fk*flow,workid,’原因’,true);

//彻底删除流程，有日志记录

Info=BP.WF.Dev2Interface.Flow*DoDeleteFlowByWriteLog(fk*flow,workid,’原因’,true);

**结束流程的三种方式说明：**

第一种，走到最后一个节点，自动结束。

第二种，在指定的节点上，设置流程结束条件。流程运行到当前节点时，符合条件就执行结束，如果不符合，继续向下运行，一直到自动结束。

第三种，在指定的节点上，启用结束流程按钮，强制结束流程。

SDK开发环境设置

BS模式下环境设置

**第1步：引用类库**

引用BP.En30, BP.WF, BP.Web.Controls30类库,

这些类库位于D:ccflowtrunkComponents

引用必要的dll 位于D:ccflowtrunkDLL 在您编译时缺少什么就引用什么。

图片地址：https://static.oschina.net/uploads/img/201506/24142704\_XnBc.png

**第2步：整理前台资源**

把CCFlow Web工程下的重要文件目录copy您的项目中去。

D:ccflowtrunkCCFlowClientBin

D:ccflowtrunkCCFlowWF

D:ccflowtrunkCCFlowDataUser

**第3步：整合配置文件**

把web.config 中的<appsetting>节点之间的部分copy到您的 web.config中去。

CS模式下环境设置

**第1步：引用类库**

引用BP.En30, BP.WF, 类库,这些类库位于D:ccflowtrunkComponents

**第2步：整合配置文件**

把web.config 中的<appsetting>节点之间的部分copy到您的项目配置文件中去。

**第3步：装载配置文件**

参考D:ccflowtrunkCCFlowServicesGlo.cs 方法把配置信息调入到ccflow系统中去。

获取ccBPM数据（SDK）

**说明：接口在源程序(BP.WF. Dev2Interface.cs)上有详尽的描述.**

获取数据是如何根据您自己的需要，通过ccBPM的接口获取想要的数据。

比如：发起流程，待办工作，在途工作。

**类名：BP.WF. Dev2Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **静态方法** | **参数** | **返回值** |
| 获取当前操作员可以发起的流程集合 | DB\_GenerCanStartFlowsOfEntities | 无 | BP.WF.Flows |
| 同上 | DB\_GenerCanStartFlowsOfDataTable | 无 | DataTable |
| 获取当前操作员的待办工作 | DB\_GenerEmpWorksOfDataTable | Fk\_flow 流程编号 | DataTable |
| 获取当前操作员的在途工作 | DB\_GenerRuningOfEntities | 无 | GenerWorkFlowExts |
| 同上 | DB\_GenerRuningOfDataTable | 无 | DataTable |

流程发起参数传递

**流程发起节界面（MyFlow.aspx）会经常被如下功能调用。**

一个流程被集成的系统调用。一个流程结束后，唤起另外一个流程。

**调用方式传递：**

发起一个流程 MyFlow.aspx?FK*Flow=流程编号&FK*Node=开始节点编号。

如果向这个流程的节点表单的文本框内，传递一些预订的数据就按如下格式。

MyFlow.aspx?FK*Flow=流程编号&FK*Node=开始节点编号&Fields=值。

**比如：**从项目列表中增加一个连接发起项目施工流程就按如下约定。

MyFlow.aspx?FK*Flow=001&FK*Node=101&PrjNo=GC001&PrjName=我的项目。

这样用户在发起这个流程时就会把项目编号，项目名称传递到用户的表单上去，您可以解决主表的参数传递问题，但是ccBPM没有解决如何把数据传递到从表，如果需要从表的数据预制就需要根据表单事件来完成。

**其它：**

用户在第一个节点保存后，ccBPM把记录表单采集的数据到草稿里，每个人，每条流程只有一个草稿，如果您要想删除这个草稿，就如下规则。

MyFlow.aspx?FK*Flow=001&FK*Node=101&IsDeleteDraft=1

如果您想删除草稿还想传递参数，按照如下格式。

MyFlow.aspx?FK*Flow=001&FK*Node=101&IsDeleteDraft=1&PrjNo=GC023&PrjName=我的项目。

SDK发开案例

概述

我们通过如下两个案例来说明如何使用sdk开发流程，第一个案例使用经典模式开发，可以掌握ccBPM基础的用法。第2个案例是利用ccBPM的sdk组件来完成开发。

我们推荐充分利用ccflow的组件来完成应用开发，只要你掌握了这些规则你就可以灵活的使用sdk满足各式各样的客户需求。

图片地址：https://static.oschina.net/uploads/img/201506/24150849\_8Ajd.png

请假流程案例1（经典SDK案例）

在您安装ccBPM的demo数据库后，您就可以看到该案例。

它的业务很简单，填写请假申请单到部门经理审批，如果天数大于10，就到总经理审批，小于等于10天就让人力资源部备案，在下面章节里我们详细讲述如何实现。

**第1步：画出流程图**

按照上述的业务逻辑画出流程图，如下图：

图片地址：https://static.oschina.net/uploads/img/201506/24150936\_AcPO.png

**第2步：编写表单**

这是一个很经典的sdk开发模式案例，我们首先编写表单，这些表单分别是:

|  |  |
| --- | --- |
| **表单名称** | **作用** |
| S1\_TianxieShenqingDan.aspx | 填写请假申请单 |
| S2\_BumenJingliShenpi.aspx | 部门经理审批 |
| S3\_ZongJingLiShenpi.aspx | 总经理审批 |
| S4\_RenShi.aspx | 人力资源备案 |

**详细请参考：**

图片地址：https://static.oschina.net/uploads/img/201506/24151125\_9BPf.png

这四个界面，建议写一个模版，然后他们都引用这个模版文件，达到风格一致。

**第2步：把表单url配置到节点属性里.**

在流程设计器里，点右键，找到节点属性，把节点表单方案设置为sdk表单，配置表单url。

图片地址：https://static.oschina.net/uploads/img/201507/08144746\_DmAi.png

点击保存按钮确定。

**第3步：创建流程数据表**

创建一个实体表，用于存放流程运转数据,这里我们用bp框架的实体类来完成的

BP.Demo.QingJia. 字段解构如下。

图片地址：https://static.oschina.net/uploads/img/201506/24151235\_BAMn.png

**这个表必须有一个int类型的字段，我们可以称为workid , 它是一个主键，每一个流程实例，系统就会分配一个workid。**

**第4步：接受4大参数，编写业务逻辑.**

每个功能界面都可以接受4大参数，**FK*Flow,FK*Node,WorkID,FID**, 然后处理相关的业务逻辑，如下图所示。

图片地址：https://static.oschina.net/uploads/img/201506/24151310\_gNZs.png

**传入SDK表单的参数：**

|  |  |  |
| --- | --- | --- |
| **参数名** | **参数值（事例）** | **解释** |
| FK\_Flow | 001 | 流程编号 |
| FK\_Node | 101 | 当前节点编号 |
| WorkID | 80810 | 工作ID,流程实例ID, 他是一个int64位的数值。 |
| FID | 80811 | 流程ID，如果该节点非子线程节点，它始终等于0， FID只有在分合流流程才有意义。 |
| UserNo | Fuhui | 当前登录者编号 |
| SID | Xwe2423sdsdwewe | 当前登录者安全校验码,用户在登录成功后，系统会直接写入一个安全校验码， 写入Port\_Emp表里的SID这一列中，如果ccflow访问一个网页时，需要校验当前操作员的合法身份， 就需要使用userNo与SID是否相等进行校验。 |
| 。。。 | 。。。 | 其他参数 |

**实现发送代码：**

图片地址：https://static.oschina.net/uploads/img/201506/24151642\_I7WG.png

**实现转向代码发送：**

图片地址：https://static.oschina.net/uploads/img/201506/24151704\_yD5e.png

**实现退回代码：**

图片地址：https://static.oschina.net/uploads/img/201506/24151719\_fkuU.png

请假流程案例2（组件模式）

使用ccBPM的sdk开发组件开发一个请假流程，使用组件开发流程界面少，开发速度快，组件功能稳定，功能灵活丰富，对公文类的流程、业务类的流程都有很好的解决方案。

**第1步：画出流程图**

图片地址：https://static.oschina.net/uploads/img/201506/24151826\_lflw.png

**第2步：设计表单**，首先建立一个表单：S11001.aspx, 把两个组件拖入界面里，然后把表单构造好。

图片地址：https://static.oschina.net/uploads/img/201506/24151852\_P1DP.png

**第3步：设置toolbar组件**，它的功能控制是有节点属性来决定的，在那个节点启用什么按钮，可以在节点属性里设置。为每个节点设置好toolbar组件。

图片地址：https://static.oschina.net/uploads/img/201507/08144645\_eQwU.png

**第4步：审核组件FrmCheck**，审核组件的属性也是在节点属性里配置，每个属性的意义请参考节点属性章节。

图片地址：https://static.oschina.net/uploads/img/201507/08144654\_YA5z.png

**第5步：把表单URL配置到节点属性里.**在节点表单方案里设置为sdk表单，在表单url填写对应的url ,该demo 可以在源文件中找到。

图片地址：https://static.oschina.net/uploads/img/201507/08144704\_CQIM.png

**注意：因为多个节点可以同享一个表单，所以其他的两个节点也是如此配置。**

**第6步：增加两个按钮发送与保存**，拖入两个按钮保存与发送，把他们放入div 里面，把他们隐藏起来。

图片地址：https://static.oschina.net/uploads/img/201506/24152052\_EBv1.png

**为按钮事件写入保存与发送代码：**

图片地址：https://static.oschina.net/uploads/img/201506/24152106\_8fxm.png

**第7步：重写编写代码**，使用javascript重写Save Send方法。

图片地址：https://static.oschina.net/uploads/img/201506/24152129\_pfHZ.png

**第8步：在Page\_Load里写入界面初始化代码**，在初始化里填写如下代码，把给控件赋值:

图片地址：https://static.oschina.net/uploads/img/201506/24152156\_E68k.png

**第9步：运行并测试流程**，启动流程：填写请假申请单.

图片地址：https://static.oschina.net/uploads/img/201506/24152222\_6Muz.png

在开始节点，把审核组件被禁用了，所有看不到审核组件

**提示发送给下一个节点：**

图片地址：https://static.oschina.net/uploads/img/201506/24152242\_HeXh.png

**部门经理审核.**

图片地址：https://static.oschina.net/uploads/img/201506/24152355\_EH6v.png

**在这个节点里审核组件被启用了，部门经理审核后发送到下一个节点。**

图片地址：https://static.oschina.net/uploads/img/201506/24152408\_Tyfq.png

**总经理审核**

图片地址：https://static.oschina.net/uploads/img/201506/24152428\_BooO.png

**该步骤审核组件被启用起来.**

图片地址：https://static.oschina.net/uploads/img/201506/24152444\_9RMG.png

菜单相关5大API

概述

从一个demo上了解sdk模式的ccBPM开发.

我们为用户准备了一个简单版本的demo，从这个demo里面您可以看到发起、待办、在途、抄送、已完成5大功能的基本调用方法。

图片地址：https://static.oschina.net/uploads/img/201506/24152710\_IvbJ.png

登录

**登录的作用就是让系统知道谁进入该系统了，系统根据当前登录者的身份，给予相应的操作权限。**

一般来说cCBPM是集成对方系统进行工作的，对方系统也有登录方法。首先对方系统需要检查密码用户名，执行自己系统的登录方法，然后调用ccBPM的登录api，就完成登录，如下图：

图片地址：https://static.oschina.net/uploads/img/201506/24152737\_Fq8Q.png

发起

**说明**：一个流程能被那些人发起是有该流程的开始节点的节点访问规则确定的，同样一个人能够发起的流程是有该人员的岗位权限部门权限决定的。发起流程的api就是返回一个结果集合。

如下图：

图片地址：https://static.oschina.net/uploads/img/201506/24152814\_ffo3.png

**展示效果：**

图片地址：https://static.oschina.net/uploads/img/201506/24152827\_LaOF.png

待办

**当前操作人员需要处理的工作，就成为待办。**

图片地址：https://static.oschina.net/uploads/img/201506/24152902\_2TfF.png

**展示效果：**

图片地址：https://static.oschina.net/uploads/img/201506/24152916\_hfpA.png

未完成

**未完成，也叫在途，它是一个人参与的流程，当前流程没有完成，并且当前待办不再该人手上。**

图片地址：https://static.oschina.net/uploads/img/201506/24152946\_jQdc.png

**展示效果：**

图片地址：https://static.oschina.net/uploads/img/201506/24152958\_e672.png

**其他：**

在途的每条工作，都可以执行如下3中功能操作。

**[撤销发送]：**撤销就是把待办放入当前人员处理的最近节点上来。比如：a,b,c,d,e,f节点，张三处理了b节点，当前已经运动到f节点上了。张三要撤销后，待办就到了b节点，张三在执行向下发送，就按照步骤进行。一个节点是否可以被撤销，能否被成功撤销是在当前停留的待办节点来决定的。

**[催办]**：就是催促当前节点处理人处理。

**[Copy发起]**：启动一个新流程，以当前流程数据为基础。

已完成

**已经完成的流程数据**，我们不建议您使用接口获取已经完成的数据，您可以直接访问数据表获取WF\_GenerWorkFlow ，根据流程状态来获取已经完成的数据，待办数据。

**流程状态字段： WFSta , 它是一个枚举值。**

图片地址：https://static.oschina.net/uploads/img/201506/24153104\_H7Z2.png

**系统的枚举值都是从这个表中获得。**

常用的SDK方法

发送方法

**发送：**发送方法是sdk最常用的方法，注意没有之一，哈哈。

**发送方法名称是**: BP.WF.Dev2Interface.Node\_SendWork他有很多的方法重写，最重要的一个方法就是。

图片地址：https://static.oschina.net/uploads/img/201506/24153225\_9pVY.png

仔细的阅读每个参数的作用，灵活的根据应用场景应用。

此方法可能返回异常信息，所以在调用该方法时需要try{ }catch{} 捕获异常，把异常信息提示给用户,请参考如下图片案例。

图片地址：https://static.oschina.net/uploads/img/201506/24153241\_0ggp.png

退回方法

**退回是比较常用的方法之一，退回方法的api是BP.WF.Dev2Interface.Node\_ReturnWork。**

仔细的看看参数，就知道如何调用该退回方法了。

图片地址：https://static.oschina.net/uploads/img/201506/24153335\_iHDQ.png

我们不建议用户直接调用api，而建议调用ccflow的这个工作部件，这个工作部件调用很简单。详细请参考：BP.WF.Dev2Interface.UI*Window*Return

图片地址：https://static.oschina.net/uploads/img/201506/24153348\_41oc.png

图片地址：https://static.oschina.net/uploads/img/201506/24153356\_mQP9.png

由于在BP.WF.Dev2Interface这个类库里，已经很清晰的描述了各个api的作用，由于同步与变更的关系，这里不再赘述。

CCBPM提供的通用组件

为了简化流程的开发，ccBPM提供了一些组件，这些组件稳定成熟，可以快速的完成我们的开发，因为组件说明书在代码文档里已经有了详尽的描述，为了升级与文档不能同步，请参考程序文档，打开如下目录查看：

图片地址：https://static.oschina.net/uploads/img/201506/24153647\_REW7.png

**包含：**

**工具栏组件**

**审核组件**

**多附件组件**

**单附件组件**

常见问题（在ccBPM的sdk开发时注意登录的问题）

如果您是采用cs开发ccBPM应用程序，在您使用CS的Form封装IE控件时，需要注意的问题是，在CS里面的IE，不能在打开新的IE,否则就会出现session丢失的情况，解决的办法就是让IE在同一个界面完成。

**比如：** CS 程序打开了一个cs的from 的IE的窗口A, A这个窗口不能弹出IE窗口B, 如果要处理B，就需要在CS放入from转入B。

系统维护和流程服务

系统设置

全局设置

概述

CCBPM的全局设置都在web.config里完成的，在每个配置项，我们都做了详细的说明，在这里不在赘述。

在里面包含了如下方面的设置：

1，  数据库连接。

2，  表单与流程全局配置。

3，  发送邮件配置。

4，  上传ftp配置。

5，  用户风格配置。

6，  运行的模式配置。

7，  其他配置。

在这里我们把重点的配置说明做特别说明。

数据相关

**数据库配置：**

图片地址：https://static.oschina.net/uploads/img/201506/24154255\_EAsA.png

流程相关

表单相关

系统枚举值

图片地址：https://static.oschina.net/uploads/img/201506/24154409\_2r6M.png

**全局的枚举值维护.**

编码表

图片地址：https://static.oschina.net/uploads/img/201506/24154449\_dVhT.png

**系统中的编码表，也称为字典表，是用户在设计表单中维护的。**

JS校验库

图片地址：https://static.oschina.net/uploads/img/201507/08145318\_55VO.png

全局变量

**用于表单中的全局变量。**

图片地址：https://static.oschina.net/uploads/img/201506/24154611\_2TS6.png

功能执行

图片地址：https://static.oschina.net/uploads/img/201506/24154625\_PZaK.png

**独立执行一个功能，每个功能后都有详细的功能执行说明，仔细的阅读这些功能说明它会对您的流程设计有帮助。**

组织结构

概述

**主要是对部门、岗位、人员、人员与部门、人员与岗位之间的关系维护。**

这些维护是在ccBPM独立运行的模式下才可以使用的。在非独立模式下，这些用户、部门、岗位都来自于OA,ERP,GRP…. 系统，这些维护功能都在主系统中。

部门维护

图片地址：https://static.oschina.net/uploads/img/201506/24154734\_uaXc.png

岗位维护

图片地址：https://static.oschina.net/uploads/img/201506/24154749\_ZxM2.png

人员维护

图片地址：https://static.oschina.net/uploads/img/201506/24154805\_OhRP.png

图片地址：https://static.oschina.net/uploads/img/201506/24154810\_ljvs.png

功能性执行

如何把数据批量的调度到流程数据中

**应用背景:**

在系统实施中，已经有了一批流程已经在机器外走完了。如何把这些流程记录到系统中去以方便统计与查询，如果让用户在走一遍流程显然是不合理的。这是就要用户流程导入功能。

这种数据分为两种情况：已经走完的流程与指定走到一个节点的流程。流程导入以上excel数据为载体导入到ccBPM系统中去。

高级开发（深度了解ccBPM）

类设计与数据库设计命名规则

**总体规则**

1，以英文命名，每一个字母大写，提倡用简写。比如Emp 人员 Dept 部门 Station 岗位。

2，如果英文名称太长，或者不常用可考虑用中文命名，建议用中文的缩写。比如管理机关GLJG。

3，对于类与表的命名必须有前缀，比如BP.Port.Dept 类名，表名为Port\_Dept.。

4，前缀命名尽量简单。比如考核命名空间英文名称为Check 空间可以命名为CH。共用的类英文名称为Public 可简写为 Pub.

**类库设计规则：**

1，格式为 BP+前缀+类名。比如：BP.Port.Emp 人员类。

2，属性命名与字段命名相同。特殊属性可以遵守总体命名规则。

3，类库的倒数第一个命名空间要与要与物理表的前缀一致。比如：物理表考核项目。

BP.CH.CheckIiems。物理表为 CH\_CheckItem。

4，如果一个属性是外部实体或者枚举类型，以His开头。比如在类人员中的部门类HisDept.

5，方法命名规则，遵守总体命名规则的第1，2条。

6，变量的命名规则，字母小写，以下划线开头。比如： age 年龄。addr 地址。要求每一行都要有注释。

**数据库设计命名规则:**

1, 物理表命名格式前缀+“*”+对应的类名。部门操作员 Port*Emp, Port*Dept ,CH*CheckItem. WF\_Flow

2，视图命名格式。“V*” + 前缀 + “*” +对应的类名。V*CH*CheckItemExt。

3, 存储过程命名规则“Pro*” + 前缀 + “*”+过程名称. 比如Pro*Sys*Init.

4，函数命名规则“Fun” + + 前缀 + “*”+函数名. 比如：Fun*Sys\_GenerOID.

5, 外键字段命名规则“FK\_” +对应的物理表类名.

比如：在人员表(Port\_Emp)中表示部门的字段 FK\_Dept. 同事一看就知道，这个字段是外键，并且它的对应到Dept这个类的物理表上。

表结构与流程运行机制

**1, CCBPM有自动修复数据表功能**, 所以表的字段的变化不需要用户干预由ccBPM自动完成. 所以如果您看到sql的错误,在执行一般就可以解决，如果解决不了，就执行一次数据库修复工具。

**2, CCBPM有自动增加字段备注功能**，所以每个字段的中文名称都已经增加了数据表的字段的备注属性上了,请注意对照，以方便您阅读ccBPM.

**3, 了解CCBPM的表结构是您二次开发的基础**，所以ccBPM的高级用户需要熟悉每个表，掌握必要的表结构. 但是没有必要全部掌握。

**4, CCBPM表命名规则是前缀+"\_"+表名**. 大多以英文缩写或者，拼音大写组成. 比如: Port*Emp 人员表. WF*Flow 流程表。

**CCBPM表分为框架表、流程规则描述表、流程数据运行表、公共表4大类:**

**BP框架表.**

通常以Sys *与 Port* 开头. 也可以称为ccBPM系统表. 它是存储系统运行的基础信息. 没有必要多了解它们.

**列举如下:**

Sys\_MapData: 映射主表  
  
Sys\_MapAttr: 字段表  
  
Sys\_MapDtl: 表单映射从表.  
  
Sys\_MapExt : 表单映射扩展信息表  
  
Sys\_MapFrame :表单框架表  
  
Sys\_MapM2M: 多对多的关系表  
  
Sys\_FrmAttachment: 表单附件表  
  
Sys\_FrmImg: 表单图片表  
  
Sys\_FrmImgAth: 图片附件表.  
  
Sys\_FrmLab: 表单标签表  
  
Sys\_FrmLine: 表单线表.  
  
Sys\_FrmLink: 表单连接.  
  
Sys\_FrmRB: 表单单选按钮.  
  
Sys\_GloVer: 全局变量表.  
  
Sys\_GroupField: 表单字段分组表.   
  
Sys\_Log:日志表  
  
Sys\_Serial : 序列号生成记录表.  
  
Sys\_Enum: 枚举信息表. SELECT\*FROM SYS\_ENUM WHERE ENUMKEY='WFState'  
  
Sys\_EnumMain: 枚举信息主表.  
  
Sys\_SFTable :自定义表  
  
Sys\_UserRegedit : 用户注册信息表.  
  
Sys\_CField: 查询字段选择信息存储表.   
  
Sys\_Config: 系统配置表.   
  
Sys\_Contrast: 统计分析表:  
  
Sys\_DefVal: 默认值表,快速填写大块文本时临时信息存储.

**组织结构表**

Port\_Emp: 操作员表.  
  
Port\_Station: 岗位表  
  
Port\_Dept: 部门表  
  
Port\_EmpStation: 人员岗位表.  
  
Port\_EmpDept: 人员部门对应表.  
  
Port\_Unit: 集团公司表,对于集团类的用户有效.

**ccBPM流程规则描述表（流程模版表）.**

**用来存储流程配置的基础信息。以 WF\_ 开头.**

**列举如下:**

WF\_FlowSort : 流程类别表.  
  
WF\_Flow : 流程表.   
  
WF\_FlowEmp : 可控制流程的人员表.  
  
WF\_FlowNode : 流程的节点表.  
  
WF\_Node : 节点表  
  
WF\_NodeStation : 节点岗位(能够访问该节点的岗位.)  
  
WF\_NodeDept : 节点部门表(能够访问该节点的部门)  
  
WF\_NodeEmp : 节点人员表(能够访问该节点的人员)  
  
WF\_NodeFlow: 节点了流程表(在此节点上能够调起的子流程)  
  
WF\_NodeReturn : 可退回到的节点(一个节点能够退回的节点,多对多的关系.)  
  
WF\_BillTemplate 单据表  
  
WF\_BillType 单据类型表  
  
WF\_Cond :条件表 (方向条件，流程完成条件，节点完成条件)  
  
WF\_Direction :方向表  
  
WF\_Event : 事件表(节点事件，流程事件)  
  
WF\_FAppSet : 应用配置表(可以配置调用第三方的数据)  
  
WF\_Frm 流程表单.  
  
WF\_FrmNode : 流程表单与节点对应关系表.   
  
WF\_LabNote : 标签用在流程设计器的标签.   
  
WF\_Listen : 消息收听配置表. 详细请 baidu "CCFlow消息收听"  
  
WF\_Rpt : 流程报表设计  
  
WF\_RptAttr : 流程报表属性  
  
WF\_RptEmp : 报表可访问的人员  
  
WF\_RptStation : 报表可访问的人员

**流程运行控制表(流程引擎控制表).**

WF\_GenerFH : 分合流信息注册表，对于是分合流的流程才有效，发起一条分合流，就多一条数据在此。  
  
WF\_GenerWorkFlow : 流程注册表。  
  
WF\_GenerWorkerlist : 节点工作人员表.(流程运行完成后此记录就被删除了)   
  
WF\_Bill : 单据信息存储表.  
  
WF\_CHOfFlow : 流程考核信息表.  
  
WF\_CHOfNode : 节点考核信息表.  
  
WF\_Emp : 流程人员配置信息表  
  
WF\_ForwardWork : 转发信息存储表.  
  
WF\_RememberMe : 投递路径记忆表, 请baidu "CCFlow智能投递"  
  
WF\_ReturnWork : 退回信息存储表  
  
WF\_M2M : 一对多的关系存储表.  
  
WF\_SelectAccper : 选择接受人

**ccBPM流程数据存储表与视图**

1, 此类的表是以ND开头的，由CCBPM自动创建自动删除的。

2, 每一个节点对应一个节点表单，每个节点表单对应一个表。

3, 节点表单的物理表的规则是”ND”+int.Prase(流程编号)+”两位的顺序号”, 比如流程编号为010 则开始节点表单编号为 ND1001。

4, CCBPM取所有节点表单字段的合集，形成一个表以ND+int.Prase(流程编号)+’Rpt’。

5, 流程轨迹表 NDxxxTrack，用来记录流程运行轨迹的数据表，也叫流程日志表，可以查看流程轨迹图，流程轨迹表。

**公共字典与Demo表.**

大多以Pub , CN开头，以Demo\_ 开头的是演示数据。

比如: 城市，省份，片区，年月，工作日. 这里不介绍了.

如果您对以上的表结构在大概的认识，您可以阅读如下文件了.

了解ccBPM数据表运行机制，为自己项目做高级的查询与二次开发(开发进阶篇).

流程实例生命周期

创建流程时

**流程表 WF*Flow, 节点WF*Node.**

**第1步：**流程自动生成一个三数数的流程编号. 比如: 001.

**第2步：**ccBPM自动生成开始与结束两个节点. 节点编号为 101 与 199. 如果在有创建节点时，节点编号就为 102, 103 ....

**第3步：**CCBPM会为每个节点生成一个物理表，这个节点的信息存储这个表里,格式为“ND” + 节点编号，比如:ND101,ND199.

**每个节点表都有相同的字段.**

**比如:** OID工作ID, RDT记录日期, Rec记录人,CDT,完成时间. NodeState 节点状态.

开始节点比较特殊, 它多一个 WFState 字段.就是流程状态,用来记录流程运行的状态.

**第4步**：建立一个流程 WF*Flow,插入一条流程数据. WF*Node 插入n条节点数据.

**第5步**： ccBPM会产生一个流程数据表,系统产生一个"ND" + 流程编号+"Rpt" 命名的表, 比如:ND1Rpt 它是流程数据表。

它把所有节点字段的汇总集合形成的一个表.流程在启动后，运行中，完成后，这个表的数据一直跟随流程运转而更新.所以您可以查询到该流程的任何数据在这个流程表中。

这个流程的扩展字段有。

|  |  |  |
| --- | --- | --- |
| **字段英文名称** | **中文名** | **说明** |
| OID | 工作ID |  |
| FID | 流程ID |  |
| Title | 标题 |  |
| BillNo | 单据编号 | 一个流程表单编号 |
| WFState | 流程状态 | 流程状态 |
| RDT | 当前日期 |  |
| Emps | 当前可操作人员 |  |
| FK\_Dept | 发起人部门 |  |
| FK\_NY | 年月 |  |
| FlowDaySpan | 流程事件跨度 | 从开始发起到流程结束 |
| FlowEmps | 参与流程的人员 | 参与真格流程上的人员，人员编号用逗号分开 |
| FlowEnder | 结束人 | 最后处理人 |
| FlowEnderRDT | 结束时间 | 流程结束的时间 |
| FlowStartRDT | 发起流程日期 | 发起流程的日期 |
| MyNum | 个数 | 恒等于1，用于分析 |

**第6步：CCBPM会产生一个节点视图，V+流程编号**. 比如:V001. 这个视图就是把本流程中的各个节点数据，相同的字段都汇总出来。

**事例如下:**

CREATE VIEW [dbo].[V001] / *WorkFlow:测试流程 Date:2011-07-30* /

AS   
  
 SELECT '101'+'\_'+CAST(OID AS varchar(10)) +'\_'+CAST(FID AS VARCHAR(10)) AS MyPK,

'101' AS FK\_Node,OID,FID,RDT,substring(RDT,1,7) AS

FK\_NY,CDT,Rec,Emps,NodeState,FK\_Dept, 1 AS MyNum FROM ND101   
  
 UNION   
  
 SELECT '199'+'\_'+CAST(OID AS varchar(10)) +'\_'+CAST(FID AS VARCHAR(10)) AS MyPK,

'199' AS FK\_Node,OID,FID,RDT,substring(RDT,1,7) AS

FK\_NY,CDT,Rec,Emps,NodeState,FK\_Dept, 1 AS MyNum FROM ND199   
  
 GO

对于开发时效考核的朋友大用用处，可以统计每个节点用的天数，规定完成的天数，完成的工作量。

**第7步：ccBPM会修复流程数据视图。**V\_FlowData。这个视图就是union 所有的流程数据表，形成的视图，如果要对全体流程执行查询，统计与分析用这个视图非常方便。

流程运行时

**分合流流程注册表:** WF\_GenerFH

**流程注册表:** WF\_GenerWorkFlow

**工作人员表:** WF\_GenerWorkerlist

1. 启动一个流程后，ccBPM自动产生一个WorkID, 一个流程一个WorkID, 并且处开始节点表(ND101)中插入一条数据, WFState=0 流程状态=0 表示运行中.

2. 启动一个流程后，就产生一条数据，放在WF\_GenerWorkFlow

3. 发送一个步骤后,就把接受的工作人员放入WF\_GenerWorkerlist .

4. 流程运行完成后，就把两个表中的这个 WorkID 的数据清除.

5, 在流程数据表里(ND1Rpt)中, 每个节点有新采集的数据就会copy到此表中.

**所以结合流程数据表,与当前的这两个表，您可以查询出来想要的数据满足二次开发.**

流程完成后（数据归档）

**流程完成后，就与 WF*GenerWorkFlow ,WF*GenerWorkerlist 没有关系了。**

所以查询流程数据从，流程数据表里，或者从节点数据表里查询. 比如: ND101, ND102, ND199, NDRpt. V001 。

**开始节点的流程状态=1 , 表示此流程已经完成。**

数据查询

我们知道了ccBPM的数据产生规则，以下是经常用到的ccBPM数据查询方法

要想二次开发您需要熟悉如下查询方法。

**查询待办一个人的待办工作:**

SELECT \* FROM WF*GenerWorkerList WHERE FK*Emp='zhoupeng' and IsPass=0 AND IsEnable=1

**查询在途工作:**

SELECT \* FROM WF*GenerWorkerList WHERE FK*Emp='zhoupeng' and IsPass=1 AND IsEnable=1

**查询一个人的一个流程的历史处理工作。**

SELECT \* FROM V\_FlowData WHERE FlowEmps LIKE ',zhoupeng,' and WFState=1

**如果对于整个流程的查询那就查询视图。**

SELECT \* FROM V\_FlowData

利用ccBPM流程数据，为客户开发一个高级流程查询页面。

如果通用查询界面满足不了您的要求，您需要自己开发一个查询界面来完成用户的查询需要，下面讲解如何开发一个高级个性化的查询界面。阅读如下内容，需要您了解了ccBPM的数据表结构，流程数据存储结构基础之上。

如果您是独立运行模式，使用的ccBPM的整体界面，那就修改ccBPM主菜单文件D:ccflowtrunkCCFlowDataUserXMLBarOfTop.xml

图片地址：https://static.oschina.net/uploads/img/201506/24171822\_BkVv.png

**设置Url=””, 在OnClick=””增加一段js代码让其转入您的查询界面中去。**

在做您个性化的查询时，您可能需要调用如下接口：

**流程轨迹图：**

http://youserverIP/ccflow/WF/Chart.aspx?WorkID=943&FK\_Flow=008&FID=0

**流程工作报告:**

http://youserverIP/ccflow/WF/WFRpt.aspx?WorkID=943&FK\_Flow=008&FID=0

**表单附件**

如何打开表单附件需要您查询了解表单设计器，多个附件的存储。在这里简述如下：

首先要明确要查询那个节点上的附件，您能明确节点ID。

查询物理表SELECT \* FROM Sys*FrmAttachmentDB WHERE RefPKVal=工作IDANDFK*MapData='ND'+节点编号

图片地址：https://static.oschina.net/uploads/img/201506/24172147\_pGco.png

总结性技术文摘

综合类

ccBPM如何提高工作流程管理系统运行效率

**关键字：工作流程管理系统,workflow,bpm,执行效率,缓存.**

CCBPM通过如下方提高业务流程系统执行效率。

**缓存机制**

**1, 什么情况下用到缓存**?有以下几条缓存数据的规则。

1, 数据被频繁利用，但是不经常更新,数据量不太大. 比如人员、岗位、部门。

2, 数据在同一个过程中经常利用,比如:流程对一个实体进行运算. 它经常被各个方法所使用.比如纳税人流程中: 纳税人的信息 土地流程中的土地信息.

3, 用户的个人信息，比如他的岗位集合，部门集合.....

**2, 缓存分类.**

全局级的: 在application 中应用. 比如:部门、岗位、人员、持久化存储的映射.

用户级的. 在session 中应用. 一个操作员的个人信息.

临时级的. 临时应用，在各个过程中应用，一个操作中消亡。

**3, 缓存怎么在CCBPM工作流引擎中表现?**

1, 系统启动时，自动的把一个常用的数据调入到全局级的缓存区。

2, 用户登陆后，把该用户的岗位、部门集合放入到用户级的缓存区.

3, 系统在运算时,大部分数据已经被加工处理好，大大减轻了服务器的负担。

**事先计算:**

事先计算就是把能够计算得到的属性在流程设计时提前计算出来，比如：节点的位置，它是一个枚举类型，开始节点、中间节点、结束节点。如果在运行中计算会消耗服务器的资源，如果在设计阶段计算好，就会节省这笔开销。

这些事先计算包含多个方面，比如：表单是否有明细、是否有框架、是否有扩展应用。。。。。

**XML 数据的缓存**

系统启动起来后就会把xml数据调入到缓存处理,节省了大量的IO 操作。

**参数执行：**

多达95%以上的sql 都是用参数执行的，您可以通过监控工具捕获到它们。CcBPM是运行在bp框架上的，所有的实体类，都是放在SQLCash 这个类中，所有的update,delete,insert,select 操作都是采用的参数执行，生成的sql放入缓存里。

关于ccBPM事件与其它表达公式的参数

在整个流程设计过程中，经常用到表达式，有表达式都要有参数，这里总结ccBPM的中的参数格式与参数约定。

**CcBPM的参数有三种类型：全局参数、表单字段参数、传递参数。**

**全局参数：**

@WebUser.No 当前操作员编号，

@WebUser.Name 当前操作员名称，

@WebUser.FK\_Dept 当前操作员部门，

@WebUser.FK\_DeptName 当前操作员部门名称。

**表单字段参数：**

获取方式是: @+字段名，比如:@WenJianMingCheng 获取文件名称。

**传递参数：**

获取方式是: @+参数用

**注意：目前不支持exe与webservices 的调用，但是可以支持url，您可以在url里调用webservices.**

**关于ccBPM对URL的执行支持与注意的开发事项：**

**1, CCBPM提供一个内部变量，@AppPath** , 它是http://+”host’+虚拟目录. 比如：

您按如下设置：@AppPath/App/DoUrl.aspx

ccBPM就会把@AppPath解析为http://127.0.0.1/CCFlow/。

**2, CCBPM支持URL外部参数，**一般是在是这种格式的执行

http://localhost/WF/MyFlow.aspx?WorkID=1111&FK*Node=12101&FID=0&FK*Flow=121.

CcBPM就会把RowURL ？符号后面的数据看作为参数。

在执行您指定的URL里面不能使用BP.Web.WebUser.\* 的参数，所调用的类也不能有关于此参数的引用，因为这些信息记录在cookies里面，ccBPM的执行是通过webrequest类来执行的，所以就获取不到这些数据。

在执行url是ccBPM会自动增加4个系统参数，它们分别是

UserNo,SID,OID,FK*Unit,FK*Dept. UserNo与SID是为您的URL的执行提供安全检查。

**同样URL的表达式与支持@WebUser.\*的参数对象。**

以子流程结束的工作流程有什么特点？设计时应注意什么事项

我们知道ccBPM有同表单分合流，与异表单分合流，与这之相对应的以子流程结束的流程也是有两种：以**子流程结束的同表单分合流，与以子流程结束的异表单分合流程。**

图片地址：https://static.oschina.net/uploads/img/201506/25113054\_mP2B.png

**对于这两类流程做如下特别说明：**

都是有多个线程，与一个主线程的流程。

每个子线程在流程结束后就要检查，当前的子线程是不是最后一个？如果是最后一个就在结束主流程。主流程完成的时间跨度是按所有的子流程都完成来计算。

为什么提倡采用瀑布方式的设计流程？

**瀑布的方式就是一个主流向分出多个细流，这种方式也叫树形的方式。**

例如：http://bbs.ccflow.org/showtopic-156.aspx

所说有瀑布的方式设计流程就是尽可能的减少合并分支，或者不用合并。因为每个路径所填写的表单是不同的，在每个节点上如果有单据打印，把打印的内容也是不同的。

**优点：**条理清晰，每个节点表单填写内容明确，在节点单据打印时数据清晰。

**缺点：**节点太多、让人感到流程复杂。

瀑布方式的流程设计是规范的流程设计，流程的复杂与否与节点多少没有直接关系。

ccBPM流程设计技巧大全(总结篇)

1, 采用瀑布方式的设计。

2, 尽量采用自由表单。

3, 先设计最后一个表单里所有的字段，然后向前copy。

关于工作流程引擎中的岗位的设置的问题

近来有朋友反馈这样的问题，一个节点需要所有的岗位上的都可以执行，就做了一个岗位全选，此时流程运行是可以的。但是过了一段时间后，岗位与人员都做了变化，尤其是岗位增加了。但是此岗位下的人员就不能访问改节点了，需要重新设置一下节点岗位才可以。这样的话所有流程的节点凡是所有岗位都可以访问的都要重新设置一次，这样就给管理人员增加了麻烦。

**为此特提出如下解决方案:**

1, 增加一个岗位编号为 00, 员工岗，它是一个通用岗位.  
  
2, 让所有的人员都必须有这一个岗位.  
  
3, 在节点设置时只设置这一个岗位就可以了.

关于流程数据的年度结转方案

关键字: 流程数据年度结转工作流引擎工作流程管理系统 ccBPM

**客户需求**

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

1, 需要减轻数据库的压力.

2, 历史数据可以被查询.

3, 横跨年度的数据与流程可以正常使用。

**实现方案:**

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

1, 为每个年度建立一套应用程序与数据库.

2, 在每年的12月31号执行数据结转.

3, 新年度的数据库里包含横跨年度的数据与流程,但是要删除已经完成的历史数据.

4, 删除所有历史年度的待办工作,因为它的待办工作都是结转到新年度的数据库里了.

**实现步骤:**

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

1，在每年的12月31号,执行结转。

2，结转之前进行数据库级的备份.

比如备份文件: ccflow2012.bak.

3, 建立结转年度数据库

比如: ccflow2012, 然后恢复ccflow2012.bak

4, 建立ccflow应用程序.

比如: ccflow2012, http://localhost/ccflow2012/ ,并修改web.config 数据库连接到 ccflow2012数据库.

5, 删除历史年度的查询机上的待办工作:

在 http://localhost/ccflow2012/ 用admin管理员进入执行 [删除所有上年度的待办工作] 功能.

6, 删除当前年度的已经完成的所有工作:

在 http://localhost/ccflow/ 用admin管理员进入执行 [删除所有本年度的待办工作] 功能.

7, 分别进入两个系统进行测试, 完成.

关于工作流引擎取回审批的设计方案与实现过程

**需求背景**

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

0, 本文中尽量避免审批概念，因为在我们认为没有审批节点或者审核节点，任何节点都是数据采集.

1, 取回处理，也可以称为提前处理，是对于拥有绝对审批权限的人，为了工作的需要直接的从下属工作人员中取回工作而直接处理。

2，这种做法是违反规范的，但是在一些场景下是需要的。比如:

1, 下属长期不处理。

2, 下属在外地.

3, 特殊的情况下不需要让特定下属知道此事件.

**可能遇到的情况:**

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

1, 被取回的点有多个人处理？

2, 什么样的数据范围可以被取回？怎么去控制？

3, 工作流程引擎是不完全计算下几步骤的处理人员的.

4, 特殊权利的节点需要，需要指定,并且访问规则必须是按人员，或者按岗位.

**处理方案:**

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

1, 此功能需要后台设置与前台处理两大部分.

2, 后台设置，要选择特殊权限的节点，此节点不能是开始节点也不能是第二个节点.

3, 以特殊权限的节点，确定可以取回工作的节点，一个特殊的节点可以对应多个可以取回工作的节点。

**CCBPM工作流引擎处理案例.**

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

1, 增加了一个类D:ccflowtrunkComponentsBP.WFWFPersonalizeGetTask.cs

此类映射到WF\_Node物理表.

2, 在流程设计器里增加一个工作取回规则设置.

http://localhost/ccflow/WF/Admin/Personalize/GetTask.aspx ，用来设置取回处理的工作规则.

3, 在前台增加了取回处理工作界面

http://localhost/ccflow/WF/GetTask.aspx

**它的操作步骤：**

1，首先选择要处理的流程.

2, 系统就会列出来可以取回的点，根据系统设置的规则，如果是空的，则此人没有此权限.

3, 取回处理就类于跳转发送，直接放在当前操作人中的待办工作里.

为ccBPM增加禁用用户立刻生效功能

**需求背景:**

一个用户被禁用了，就必须立刻生效，这就意味着他不能操作任何事情，也许是这个用户正在处理事情。

**实现方式:**

1, 在WF\_Emp 表中增加一个字段UseSta 默认为1, 0标识禁用，1启用.  
  
2, 在 /WF/UC/MyFlow.ascx 的Page\_Loag中增加判断，如果被禁用了就立刻停止改用户的使用。  
  
3, 在web.config 中增加一个配置, 如果没有此配置项，就默认为false.

<!-- 是否启用检查用户的状态 -->

<add key="IsEnableCheckUseSta" value="1" />

**其它:**

1, 此功能已经发布.

2, 如果没有必要就不要启用此功能.

关于工作流引擎授权问题的需求变更

各位ccBPM爱好者，如果您在使用ccBPM的授权功能，请仔细阅读。

**新需求点说明:**

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

1, 同一个时间点一个人只能授权给一个人。

2，需要按流程的范围授权，或者全部授权。

3, 需要为授权增加一个日期.

4, 被授权人登录后的每一步操作，都要需要记录实际处理人，所以需要在日志中扩充一个列来记录实际执行人。

**变更内容:**

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

1，原来的 WF\_Emp 表中的 AuthorIsOK 取消了.

2, 新增加了 AuthorWay 0, 未授权， 1，全部授权，2按指定的流程授权.

如果此枚举值，有变化ccflow会另行通知。

3, 新增加了 AuthFlows 字段，用来存储授权的流程编号，流程编号用逗号分开.

4, 新增加了 AuthorToDate 授权到日期, 如果逾期登录会提示出错，但是ccBPM不会自动收回授权。

5, 在 NDxxxTrack 表中增加了一个Exer 列，来记录与显示实际执行人。

**etc:**

此类文件位于D:ccflowtrunkComponentsBP.WFPortWFEmp.cs

**变更对被授权登录的影响:**

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

1, 发起,待办,在途,抄送. 在原来授权人的基础上增加了被授权的流程范围过滤.

2, 其它未影响到的未知。

如何为ccBPM工作流引擎增加一个优先级PRI

对于一条流程的优先级可分为低，中，高三个级别，用这个状态来标识这条流程的紧急程度。(说明:这个紧急程度是可以被自定义的,请打开select \* from sys\_enum where enumkey=PRI ,如果您要修改CCFlow的默认枚举值，同时也要修改每个枚举值对应的图片,它位于D:ccflowtrunkCCFlowWFImgPRI .)

以以前版本的ccBPM中是以节点表单的标志来处理的，为了统一管理现在纳入了整个工程的属性里。

**设计与实现:**

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

0, 流程的优先级是一个全局属性，如果启用了任何一个流程都要有优先级别。

1, 在开始节点表单中增加一个PRI字段，枚举类型的. 0低，1中，2高默认为1。非开始节点表单可以没有PRI字段。

2，如果不需要PRI字段就可以把改字段隐藏了.

3, 在流程注册表中WF\_GenerWorkFlow 也增加PRI字段。

4，在待办工作界面里可以按此字段进行分组。

5, 如果流程在中途需要修改流程优先级，就要可以在所在节点表单中改变PRI这个值，一般的情况下PRI是发起人确定的。

6, 如果采用SDK表单的开发模式，需要有参数的方式把PRI的值传递到节点表单里。

7, 如果您要修改CCBPM的默认枚举值，同时也要修改每个枚举值对应的图片,它位于D:ccflowtrunkCCFlowWFImgPRI .

**配置步骤:**

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

1, 在web.config 文件中增加 IsEnablePRI 设置 0 不启用， 1 启用。默认不启用。

<!-- 是否为流程增加一个优先级 -->

<add key="IsEnablePRI" value="1" />

2，执行一次全局的流程体检操作路径流程设计器-》系统维护-》系统设置-》功能执行->修复节点表单。

**兼容问题:**

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

1, 对于老用户来说开始节点没有此字段，在执行流程体检后，ccBPM会自动增加上.

2, 为每一个开始节点表单调整一下PRI 字段的位置.

3, 如果您的系统没有对流程进行区分优先级的需求，就不必升级。

关于工作流引擎流程轨迹表的效率提高改进内容发布

关于工作流引擎流程轨迹表的效率提高改进内容发布

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

**关键字: 工作流引擎工作流管理系统CCBPM流程轨迹 wf\_track**

**什么是流程轨迹表?**

是CCBPM用于存储整个流程运动过程生命周期中的每一步的操作的轨迹数据表, 它存储在 WF\_Track 表里.

**为什么要变更?**

1，访问频繁.

2, 插入的数据量大而频繁.

3, 给流程的执行效率带来很大的问题，尤其是流程运行二三年后，更为突出.

**如何进行变更,变更方案的方案是什么?**

1, 创建了一个WF\_TrackTemp 表，此表用于临时存储轨迹数据.

2, 在调度里定期的把WF*TrackTemp数据表的数据copy到WF*Track 同时删除已经copy过去的数据.

3, 调度放在每天夜晚执行，由D:ccflowtrunkCCFlowServicesbinReleaseCCFlowServices.exe 完成.

如果分布在两个表里 ccBPM如何来完成查看流程的历史轨迹，会不会对历史数据造成影响？

用union 的方式来完成，对历史的数据不会造成影响。

**其它:**

1，流程轨迹分布在两个数据表里，ccBPM是按照union 的方式查询的.

2, 此次的变更不会对历史数据造成影响.

3, 对说业务量较小，使用时间不长，没有遇到效率问题的用户不必去更新.

**更新内容:**

全部.

项目工程类的流程在ccBPM中应用与设计说明书

项目工程类的流程在ccBPM中应用与设计说明书

**说明:**

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

1, Prj为项目工程所用的一些扩展应用,项目流程也可以称为工程流程，是为了建造一个跨期比较长的工程在每个环节上需要流程控制。

2, 项目流程权限控制比较特殊，比如一个人在a项目中是项目经理，在b项目中是工程师。这类的流程在节点权限控制时会遇到局限在本项目中流转。

3，一个项目会有资料树的概念，每一个有关项目的流程具有一些文档，这些文档会于项目关联。

4，项目流程ccBPM在表单设计中有一个约定。每个节点表单中（流程表单除外）都必须有PrjNo,PrjName两个属性.

**在流程设计时必须指明此流程为项目类型的流程。**

**环境搭建**

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

1. D:ccflowExpandingApplicationPrjPrj 复制到D:ccflowtrunkCCFlow 中去.

2. 在CCFlowwebapplication 中引用 D:ccflowExpandingApplicationPrjComponentsBP.PRJBP.PRJ.csproj

3. 在目录: D:ccflowtrunkCCFlowDataUser 下创建一个Prj.Data 目录. 这个目录将会放资料树。

D:ccflowtrunkCCFlowWFDataUserPrjDataData 资料树数据．

　　D:ccflowtrunkCCFlowDataUserPrjDataTemplete　资料树模板数据.

**编译该项目.**

**创建您的第一个工程项目:**

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

1, 在流程设计器中建立一个项目, 该项目的编号为 0001.

2, 为该项目设计项目树模板.

2.1 创建目录: D:ccflowtrunkCCFlowDataUserPrjDataTemplete001

2.2 在以上目录下创建目录模板, 系统中读取文件名.文件里面可以没有任何内容. (为了方便期间您可以把 D:ccflowExpandingApplicationPrjTemplete\*.\* copy 里面测试)

**目前只支持一级的资料树.**

2.3 说明:在您上传资料后，系统就会自动的把文件上传到D:ccflowtrunkCCFlowDataUserPrjDataData001 下面。

1. 为该项目设计岗位与人员.设计资料树节点访问规则.

2. 新建一个流程，在流程属性中设置该流程为项目类型的流程.

3. 为该流程的每个节点表单都要增加一个PrjNo,PrjName字段.

6, 在表单上，增加一个超连接连接地址为 ./Prj/DocTree.aspx?FK*Prj=@PrjNo&FK*Node=当前节点ID. 系统就会打开项目树了.

如何获取ccBPM的待办工作列表处理工作到达的自动提醒

**第一种方式：您可以利用ccBPM的sdk去获取。**

这个方法是 BP.WF.Dev2Interface.DB\_GenerEmpWorksOfDataTable(); 这个静态的方法可以返回一个datatable.

**第二种方式：您可以直接查询数据库。**

datatable= BP.DA.DBAccess.RunSQLReturnTable("SELECT \* FROM WF\_EmpWorks WHERE FK\_Emp='" + fk\_emp + "' ORDER BY ADT DESC ");

获得返回来的table ,您就可以分析里面的数据进行工作提醒了。

**列的详细描述如下:**

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

WorkID - 工作ID

Starter -发起人

StarterName -发起人名称

FK\_Flow -流程编号

FlowName -流程名称

FK\_Node - 节点

NodeName -停留节点名称。

Title -标题

RDT -记录时间

ADT -发送时间

SDT - 应该完成日期

FK\_Emp -操作员编号

FK\_EmpText -操作员名称

FID -流程ID

FK\_FlowSort -流程类别

ccBPM工作流引擎中如何在特定的一个步骤，调用起另外一条流程

**需求描述:**

1, 操作员在操作最后一个节点时，点发送按钮，希望结束当前的流程并且调用起来另外一个流程，让它在停留在该流程第一个节点的输入界面上。  
  
2, 操作员在操作最后一个节点时，点发送按钮，希望结束当前的流程并且调用起来另外一个流程，让它在停留在该流程制定的节点的输入界面上，并且让第一个节点的工作保持完成。

**需求描述的扩展说明:**

1，不仅仅是最后一个节点，还可以认为是中间的节点, 也就是说，中间的节点在发送后，吊起特定的流程，并且父流程也要向下运动.  
  
2, 操作节点是中间点, 发送后，结束当前的流程，并吊起子流程.

**解决办法:**

**需要如下两方面的知识:**

1, 父子流程调用.  
  
2, 节点属性-》基本属性-》发送后转向.

**以上两个知识点都在流程设计操作说明书里.**

**在发送后转向，直接按照一定的规则转向这个特定的子流程url就可以了。**

关于发送接口的变更与ccBPM的发送后返回的SendReturnObjs对象

**各位ccBPM的API开发者：**

**根据不同的设备获取发送后的信息需要，现在ccBPM的发送接口变更如下。**

1，  取消了WorkNode 类中的AfterNodeSave 现修改为NodeSend.

2，  原来此方法返回的是一个string 串，来表示执行的信息，现在返回一个SendReturnObjs一个对象。此对象的.ToMsgHtml() 的方法与原来的返回string 对象一致。

3，  原来在BP.WF.Dev2Interface中的Node\_Send 方法现在都返回了SendReturnObjs对象，如果您要想获取原来发送后的信息就调用ToMsgHtml()方法。如果您想把信息输出到其它的设备上可以调用ToMsgText()方法。比如您要想把发送后的信息放到手机上，或者silverlight 程序上，系统就无法解析html中的链接信息。

4，  对于想用新版本的用用户来说，您只是做少量的代码修改就可以了。

**比如:** 原来的

string nodeSendMsg=BP.WF.Dev2Interface.Node\_SendWork(‘001’,203,null);

**现在修改成:**

string nodeSendMsg=BP.WF.Dev2Interface.Node\_SendWork(‘001’,203,null). ToMsgOfHtml();

就可以了。

**关于ccBPM的SendReturnObjs对象：**

此类位于: D:ccflowtrunkComponentsBP.WFWF5.0SendReturnObj.cs

此对象是调用ccBPM的API发送接口返回来的一个发送结果对象，此对象是从集合类上继承下来的，里面存储了发送过程中的各种信息，每一种信息都有一条标识。这种标识是恒定不变的，api的开发者在获取这个发送后返回对象后，就可以获取发送过程中产生的各种新信息，比如到达了那一个节点，下一步的接受人是谁，是否发送成功。

**一条消息有4个属性：**

**消息标识:** MsgFlag

**消息类型：**HisSendReturnMsgTypeInfo消息，SystemInfo系统消息

**粗体**文本消息:MsgOfText

**Html消息:**MsgOfHtml 此种消息可以输出超链接。

此外，此对象也有助于开发者判断错误。

如何让ccBPM实现邮寄模式的工作流程设计

**邮寄流程模式的定义:**

现用形象的实例说明，在一个小区里有一个邮箱，小区的居民都向这个邮箱由投递要发送的邮件(每个人可以看做一个子线程), 邮寄员定期的或者不定期的(自动或者手动)来取出这些信件，交给邮局，然后邮局就开始执行下一步的送达工作.

**我们知道ccBPM的节点类型有普通节点、分流点、合流点、分合流点、子线程点。如果出现分合流的情况，必须有一个分流点在前。**

上面所描述的流程模式目前ccBPM是不支持的，因为ccBPM每启动一个流程必须首先分配一个WorkID, 如何使用变通的办法来解决这个问题？请按如下步骤去做。

1，建立一个邮件表，存放社区成员的投递信息，让其每个投递(也就是每个子线程)就产生一条数据.

这个数据采集的界面自己去开发。

2, 建立一个流程，参考流程设计器操作手册中的 3.9.3章节如何设置流程的自动发起?

解决思路: 让子线程单独在流程之外处理。

ccBPM工作流引擎,如下方式完成对流程的结束大总结

**1, 正常完成:**

流程运行到最后的一个环节.

**2, 非正常结束:**

流程运行到指定的节点，满足的指定的流程结束条件，自动结束流程. 这种模式需要在节点的流程完成条件里设置。

**3, 强制结束:**

流程运行到特定的节点后，因当前操作员的需要强制删除该流程. CCBPM为我们提供了结束流程的功能按钮.

**4，以删除的方式结束流程:**

谈起这种模式首先要掌握ccBPM的删除规则:

**这种模式CCBPM有相应的api 与节点功能按钮，可以在节点属性按钮部分的流程删除规则，以删除的模式结束流程有如下几种规则。**

@0=不能删除@1=逻辑删除@2=记录日志方式删除@3=彻底删除@4=让用户决定删除方式  
  
@0=不能删除: 该流程不允许用户删除.   
  
@1=逻辑删除：仅仅做上删除标记, 数据仍然在节点表单与流程报表里面.  
  
@2=记录日志方式删除 : 删除节点表数据，流程报表数据，并记录到 wf\_workflowDeleteLog 表里作备案.  
  
@3=彻底删除 : 彻底清除该流程的所有数据.  
  
@4=让用户决定删除方式 : 显示出对话框，让用户选择上面的删除方式.

说明:

**以 1,2,3 模式与逻辑删除删除四种情况下结束的流程，可以会滚，其他模式结束的流程就不能会滚。**

为什么驰骋工作流引擎提倡用事件来处理完整性检查业务的完整性？

为什么驰骋工作流引擎提倡用事件来处理完整性检查业务的完整性？

**关键字: 驰骋工作流引擎表单引擎表单提交安全性检查**

**说明:**

我们知道，表单提交前的安全性检查是很有必要的，很多的开发者首先要考虑的是javascript检查，就是在发送的按钮里写一个事件,用这个事件来执行检查用户的输入是否合法。

在这里ccBPM要提倡的这种检查放在事件里进行，有如下原因。

1，事件里容易表达业务逻辑。  
  
2，事件里支持多种表单的检查，目前ccBPM提供了aspx表单的解析执行器，正在完善着slilver的表单解析执行性，与正在开发的android表单解析执行器。所以你在编写js的脚本在这些设备下不能很好的应用。

基于以上两个原因，所以提倡你使用事件做安全性检查，熟练的使用ccBPM的事件是你在驰骋工作流引擎扩展开发的基础。

把消息写入驰骋即时通讯CCIM

**不建议新手研究CCIM**

在ccBPM运行过程中会产生很多的消息，比如：待办、退回、转发、流程侦听、抄送。这些消息需要通过一定的方式传(比如：邮件、短信、内部消息、移动设备、即时通讯)递给相关人同处理。

本章节介绍如何把ccBPM的消息写入驰骋即时通讯CCIM。

**步骤如下：**

第一步：要把ccBPM集成到GPM中，请参考上一章节。

第二步：启动流程服务，把写入CCIM的开关打开。

图片地址：https://static.oschina.net/uploads/img/201506/25155043\_CIdM.png

流程类

关于:/WF/FlowFormTree/FlowFormTreeEdit.aspx 功能界面使用说明

关于:/WF/FlowFormTree/FlowFormTreeEdit.aspx 功能界面使用说明

**此页面是功能调用页面，它需要两个参数. FK\_Flow,WorkID.**

是用来解决流程表单树的查看与编辑，他对应的是流程编号而非节点编号.

**主要应用到如下场景:**

1, 需要查看流程信息，而非指定特定的节点。  
  
 2，需要查看流程信息，并且可以编辑表单信息，应用于数据采集，即时该流程已经走完。  
  
 3, 在该功能工具栏上，仅具有，保存，轨迹，关闭三个按钮。

**相关：**

1, 正常流程节点表单树的信息绑定，实在节点属性，节点表单中设置。  
  
 2, 节点绑定的表单在节点属性设置. 流程绑定的表单在流程属性里设置.  
  
 3, 节点绑定流程表单的功能是页面是 FlowFormTreeView.aspx 需要的参数是 FK\_Flow,WorkID,FK\_Node,FID