

TASKCTL

敏捷批量调度开拓者，开启批量调度工具化时代

敏捷调度技术平台 v7

Designer 设计 IDE 环境

成都塔斯克信息技术有限公司

产品网站：www.taskctl.com

1 前言	5
1.1 文档目的.....	5
1.2 读者对象.....	5
1.3 版本修订.....	5
2 认识开发环境	6
2.1 组织架构.....	6
2.2 登陆.....	7
2.2.1 本地登录.....	7
2.2.2 服务器登录.....	8
2.3 主界面.....	8
2.4 窗口布局.....	9
2.5 创建一个简单流程.....	10
3 开发环境详解	12
3.1 目录菜单.....	12
3.2 工具栏.....	14
3.3 状态栏.....	15
3.4 资源管理窗口.....	15
3.4.1 工程节点.....	16
3.4.2 工程变量节点.....	17
3.4.3 流程节点.....	17
3.4.4 流程变量节点.....	18
3.4.5 模块节点.....	19
3.4.6 资源节点搜索框.....	19
3.5 模块设计窗口.....	20
3.6 流程节点工具箱.....	20
3.7 属性窗口.....	21
3.8 消息窗口.....	22
3.8.1 错误输出窗口.....	22

3.8.2 流程编译输出窗口.....	23
3.8.3 查询结果输出窗口.....	23
3.9 查找与替换.....	23
4 流程开发设计.....	25
4.1 流程签入与签出.....	25
4.2 流程创建与删除.....	25
4.3 流程编辑.....	26
4.3.1 流程基本属性编辑.....	26
4.3.2 变量管理.....	27
4.3.3 模块新增、修改、删除.....	27
4.3.4 模块代码编辑.....	29
4.3.5 模块图形编辑.....	29
4.3.6 串并关系结构.....	30
4.3.7 节点属性编辑.....	32
4.3.8 保存模块.....	43
4.4 流程编译.....	44
4.5 流程发布及归档.....	44
4.6 一些快捷操作技巧.....	45
4.6.1 如何快速在图形设计器中定位指定作业.....	45
4.6.2 如何快速在图形设计器中定位指定模块.....	45
4.6.3 如何快速通过图形增删一个作业.....	46
4.6.4 如何快速将大模块拆分更多的小模块.....	46
4.7 流程相关备注信息的重要性.....	46
5 定时器开发设计.....	47
5.1 定时计划 TIMINGPLAN.....	47
5.2 其他.....	48
6 流程运行与监控.....	49
7 快捷键速查表.....	50

8 常见问题 QA.....51

1 前言

1.1 文档目的

桌面软件 Designer 是 TASKCTL 调度平台客户端工具软件之一，本文旨在介绍该软件的功能与操作，以便帮助相关人员对该软件的使用。

1.2 读者对象

《TASKCTL Designer 设计 IDE 环境》主要适合以下读者对象

- ✓ 技术开发人员
- ✓ 项目实施人员

1.3 版本修订

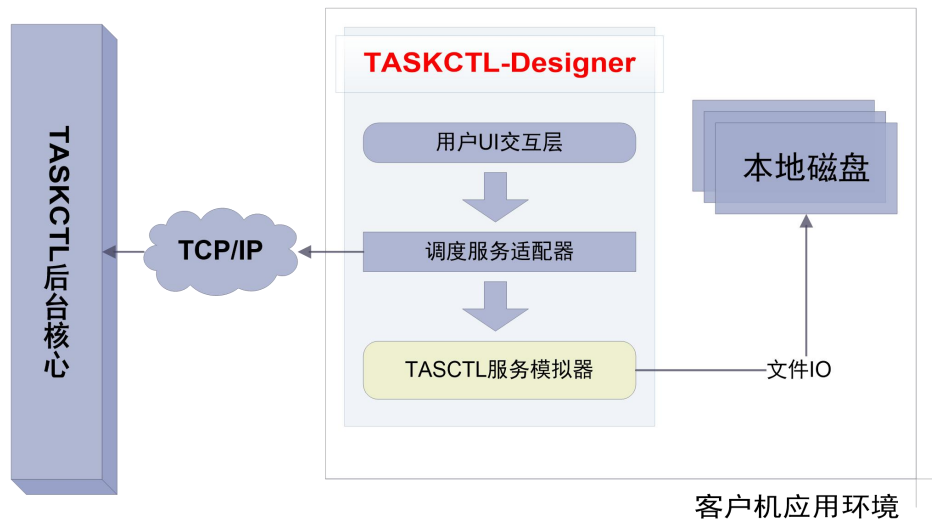
为了支持不同版本（4.1、5.0、5.1、6.0、7.0）的文档说明，在文档中会用如下标识来标注版本支持性：

- v6.0+ ：表示从 6.0 版本开始支持
- v7.0- ：表示从 7.0 版本不再支持

2 认识开发环境

流程配置信息是调度的主要信息来源,掌握流程配置是掌握调度的主要方式。流程配置的主要思路都是对作业进行节点化抽象,并从作业之间的关系描述入手。TASKCTL 借鉴了现代程序语言的开发概念,以具有一定语法的代码编辑为基础,推出了一套支持代码/图形开发的设计集成环境-Designer。该平台不仅使调度流程设计更灵活,也更便捷直观。摆脱了很多传统产品工具面对大量作业流程设计时,不得不抛弃本身基本的表单配置功能,而借助诸如 Excel 等第三方工具的窘境。同时集成了流程编译环境,使流程开发设计过程中具有可调试性及较强的指导意义。

2.1 组织架构



通过上图了解到,本开发平台除了能通过调度服务器适配器以 TCP/IP 协议的方式与 TASKCTL 后台核心服务器通讯外,还集成了 TASKCTL 服务模拟器组件,直接进行本地开发。

2.2 登陆

Designer 目前支持两种登录方式：“本地登录”和“服务器登录”。“本地登录”方式一般是为了模拟本地调度流程开发的环境，以及某些场合的流程调度演示。在正式的生产环境中，采用“服务器登录”方式登录到 TASKCTL 服务器。

2.2.1 本地登录

不需要登录 TASKCTL 服务器，使调度流程脱机开发变为了可能。



如上图所示：选择登录方式为“本地”，不需要输入用户登录信息直接点击“登录”按钮，就可以快速登录到调度服务模拟器进行本地开发。

2.2.2 服务器登录

该方式需要登录 TASKCTL 服务器，通常正式的开发都是通过这种方式进行流程设计。

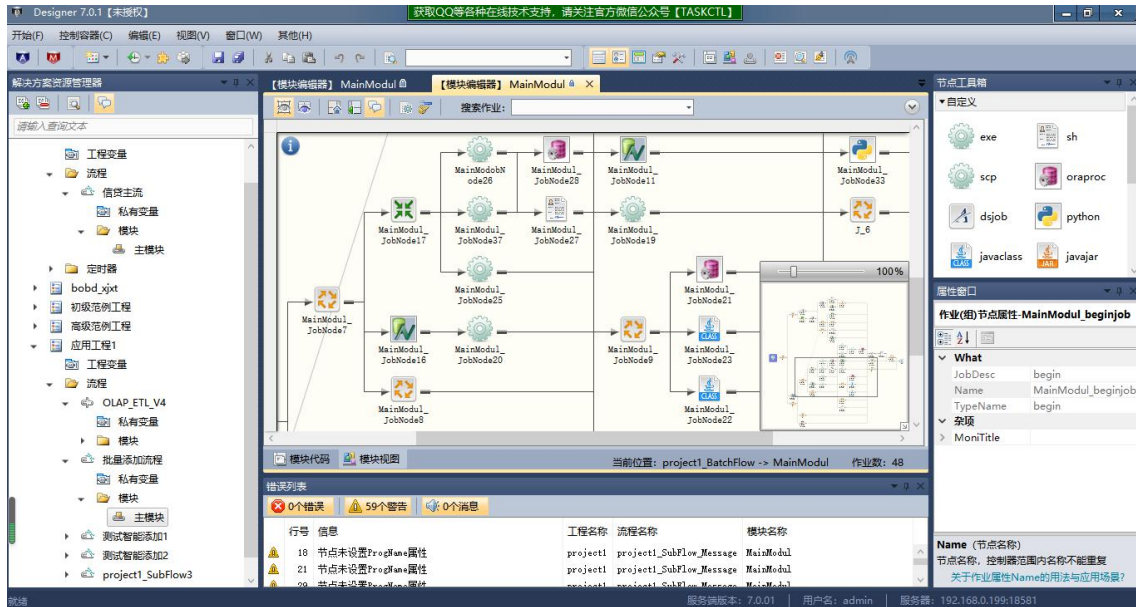


如上图所示：选择登录方式为“服务器”，输入正确的用户名、密码及服务器地址点击“登录”按钮，登录到指定的 TASKCTL 服务器。该服务器地址为服务器部署时所设置的“核心服务节点”地址。登录成功后，Designer 将根据登录用户进行系统进行一系列的初始化设置。

说明： 系统自动记录了每次成功登录的用户信息，可选择进行快速登录。

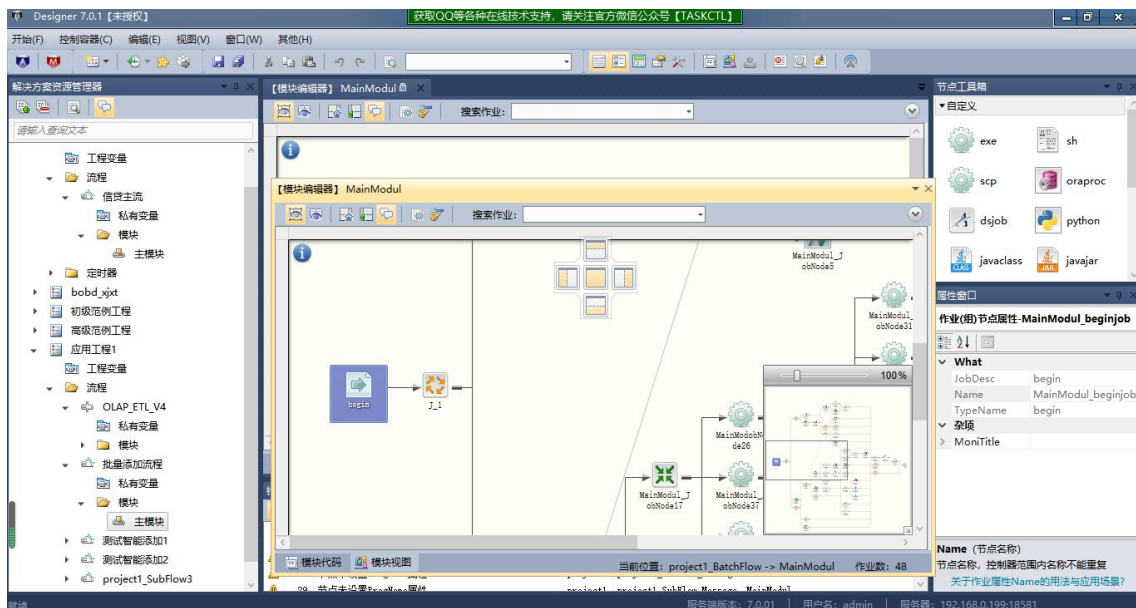
2.3 主界面

以下图示是成功登录后的主界面，可以较直观的看到 Designer 具备了传统桌面应用程序的“目录菜单”，“工具栏”，“状态栏”以及“工作区”。工作区又细分为“资源区”，“开发设计区”，“信息输出区”，“工具箱”，“属性窗口”等窗口。



2.4 窗口布局

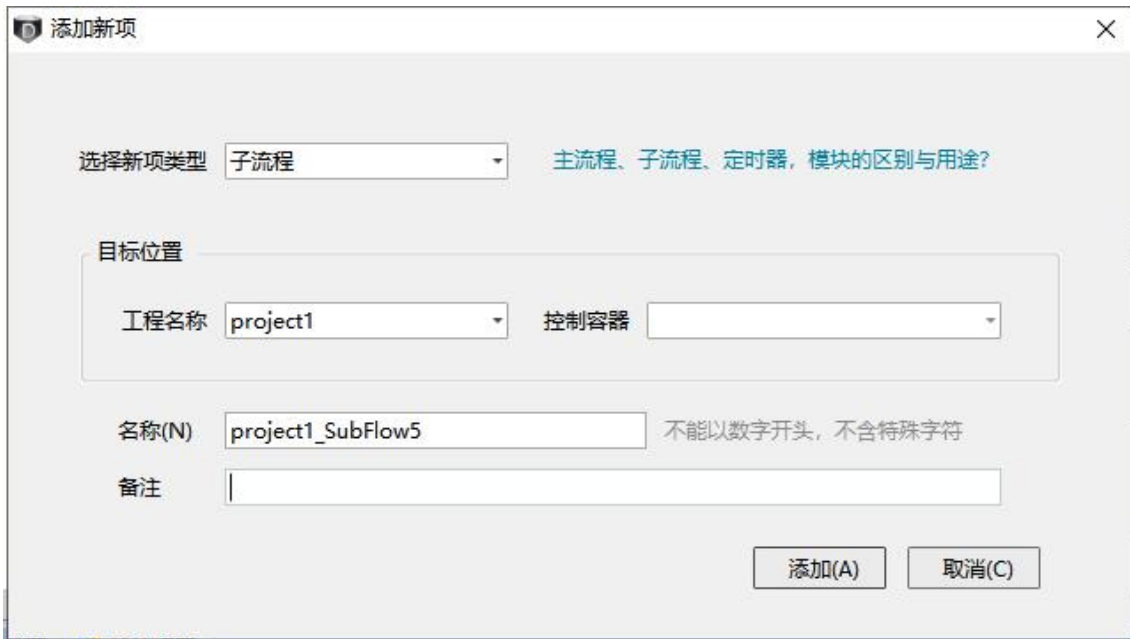
Designer 采用了时下多数 IDE 工具“多文档及浮动窗口”的布局模式。您可以同时打开多个开发设计窗口。另外，通过对工作区窗口拖拽及浮动，可以自定义个性化的开发环境。如下图所示：



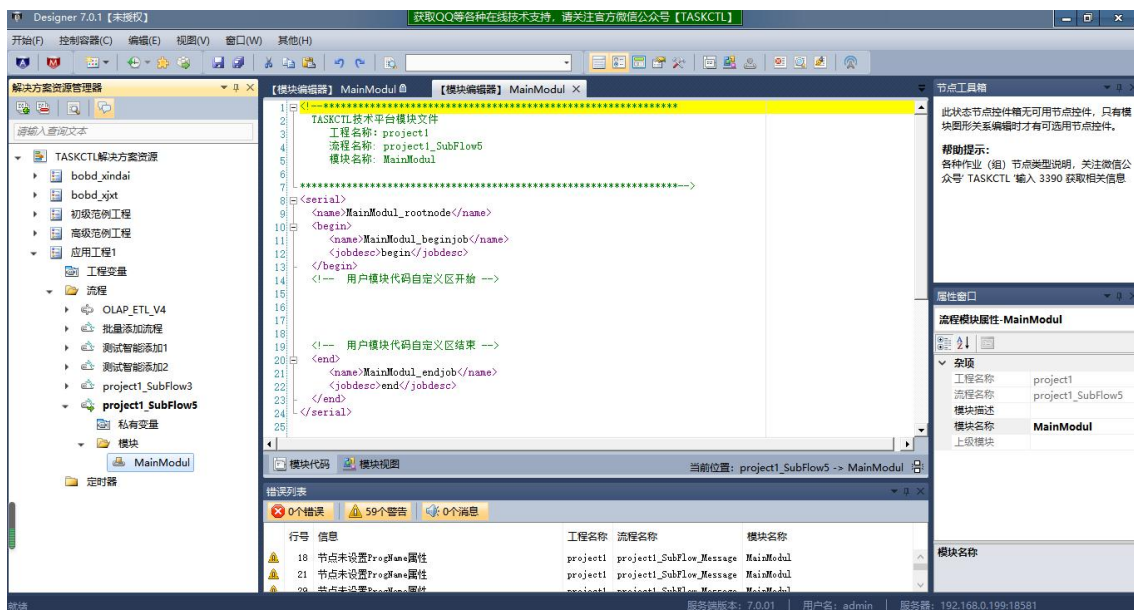
2.5 创建一个简单流程

在详细介绍开发环境之前，下面将通过几个步骤在 Designer 创建一个简单的流程。

1. 右键单击“资源管理器”里的工程节点“应用工程1”，新建控制容器子流程。弹出控制容器创建窗口，如下图：



2. 点击“添加”菜单项，系统将自动创建一个子流程控制容器。



通过上图可以看到，系统自动为新流程命名为“project1_SubFlow5”，同时为

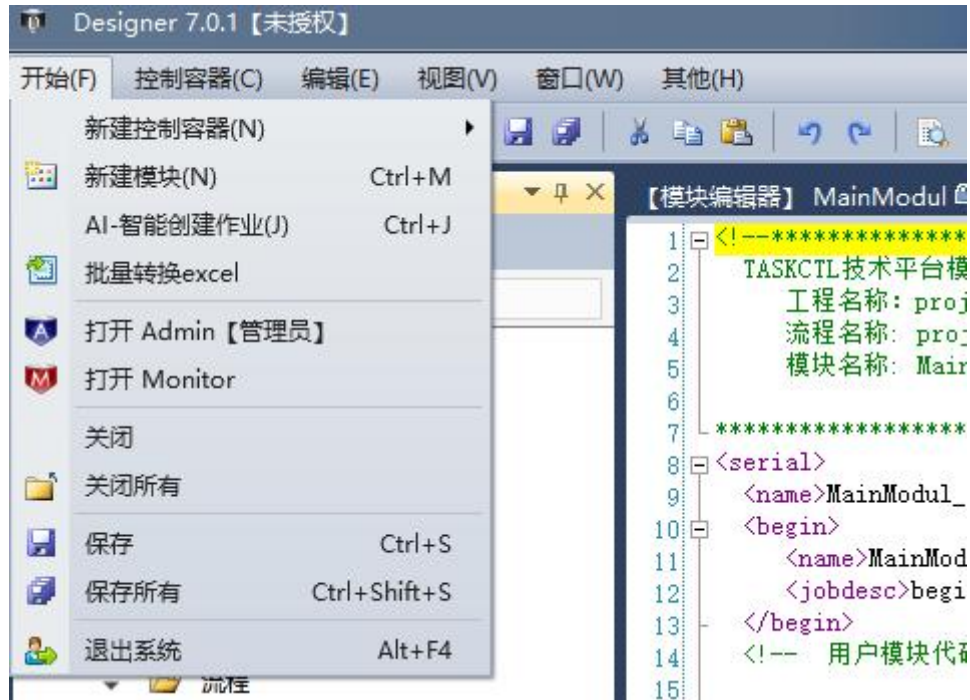
其建立了主执行模块“MainModul”，以及创建了一个基本的模块代码结构。接下来您可以在“用户模块代码自定义区”内增加作业了。

完成好流程设计之后，按快捷键 **F6** 编译该流程。编译通过并发布后，就可以用“监控维护管理平台-Monitor”来运行该流程。

3 开发环境详解

3.1 目录菜单

目录菜单区按照流程开发应用场景的不同,分为了“开始”、“控制容器”、“编辑”、“视图”、“窗口”、“其他”六大子目录。如下图所示:



下表展示了平台目录菜单结构以及菜单项对应的快捷键和说明:

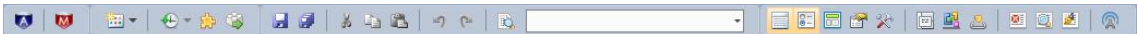
子目录	菜单项	快捷键	说明
文件		Alt+F	
	新建控制容器		该菜单项还拥有子菜单项
	新建模块		
	AI-智能创建作业	Ctrl+J	
	批量转换 excel		
	打开 Admin		当前用户需要管理员权限
	打开 Monitor		
	关闭		
	关闭所有		对模块设计器窗口进行操作
	保存	Ctrl+S	
	保存所有	Ctrl+Shift+S	
退出系统	Alt+F4		
控制容器		Alt+C	
	删除容器		资源树选定流程节点

	删除模块		资源树选定模块节点
	装载归档		
	编译	F6	
	发布		
	签入		资源树选定流程节点
	签出		
编辑		Alt+E	
	撤销	Ctrl+Z	当前设计器为代码设计器
	重做	Ctrl+Y	
	剪切	Ctrl+X	
	复制	Ctrl+C	
	粘贴	Ctrl+V	
	删除	Del	
	全选	Ctrl+A	
	查找和替换	Ctrl+F	
	XML 格式化	Ctrl+K	
	从选定文本新建模块	Ctrl+P	
视图		Alt+V	
	工具栏		
	状态栏		
	资源管理器	F8	
	模块代码	F12	
	模块视图	F11	
	流程变量		
	代码错误列表	Ctrl+W	
	查询结果列表	Ctrl+R	
	编译输出	Ctrl+O	
	工具箱	F10	
	属性	F9	
	窗口		Alt+W
拆分代码窗口			对模块设计器窗口进行操作
浮动			
停靠			
以选项卡式停靠			
关闭			
关闭所有			
窗口列表集合			
窗口			
其他		Alt+H	
	密码修改		
	软件授权		
	关于		

说明：在某些目录菜单项靠右的位置，具有执行该菜单项命令的快捷键。





3.2 工具栏


工具栏如同目录菜单，提供了一些系统常用命令快速执行入口。在视图子菜单里的“工具栏”菜单项可对工具栏进行隐藏和显示。



如上图所示，工具栏除了包含了一系列的命令按钮外，下表展示了工具栏中的按钮对应的快捷键和说明：

工具栏控件	图标	快捷键	说明
打开 Admin			
打开 Monitor			
新建...			该菜单项还拥有子菜单项
新建控制容器			
新建流程		Ctrl+L	
新建定时器		Ctrl+T	
新模块		Ctrl+M	
AI 智能创建作业		Ctrl+J	
装载归档			若当前控制器存在多个发布归档，可选择加载
编译		F6	
保存		Ctrl+S	
保存所有		Ctrl+Shift+S	
剪切		Ctrl+X	当前设计器为代码设计器
复制		Ctrl+C	
粘贴		Ctrl+V	
撤销		Ctrl+Z	
重做		Ctrl+Y	
查找和替换		Ctrl+F	
工具栏			
状态栏			
资源管理器		F8	
属性		F9	
工具箱		F10	
模块代码		F12	
模块视图		F11	
变量编辑器			

代码错误列表		Ctrl+W	
查询结果列表		Ctrl+R	
编译输出		Ctrl+O	
关于			

说明： 鼠标悬浮到工具栏按钮(如：“”), 可以查看到命令的快捷键。

3.3 状态栏

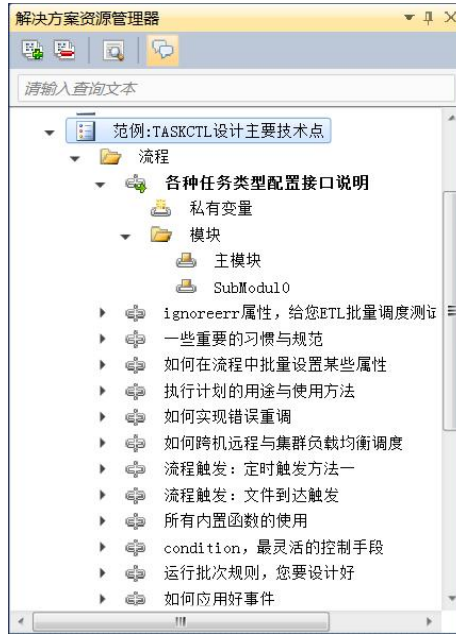
概括的讲, 状态栏动态的表述了系统命令的执行情况以及系统状态。通过状态栏, 可以实时掌握系统命令的执行动态。



如上图所示, 状态栏中显示了系统正在编译某一个流程以及命令的执行进度。另外视图子菜单可以对状态栏进行隐藏和显示。

3.4 资源管理窗口

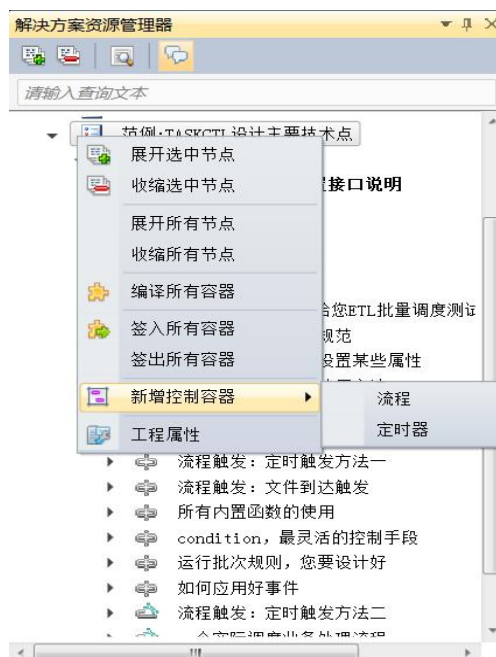
通过资源管理器, 可以快速的打开模块代码、树节点属性、流程变量等设计窗口。并直观的展示了工程到流程, 流程到模块的树形结构。如下图所示:



资源管理器拥有自身特有的工具栏，通过其工具栏按钮，可以方便地对资源树及其选中节点进行收缩和展开等操作。

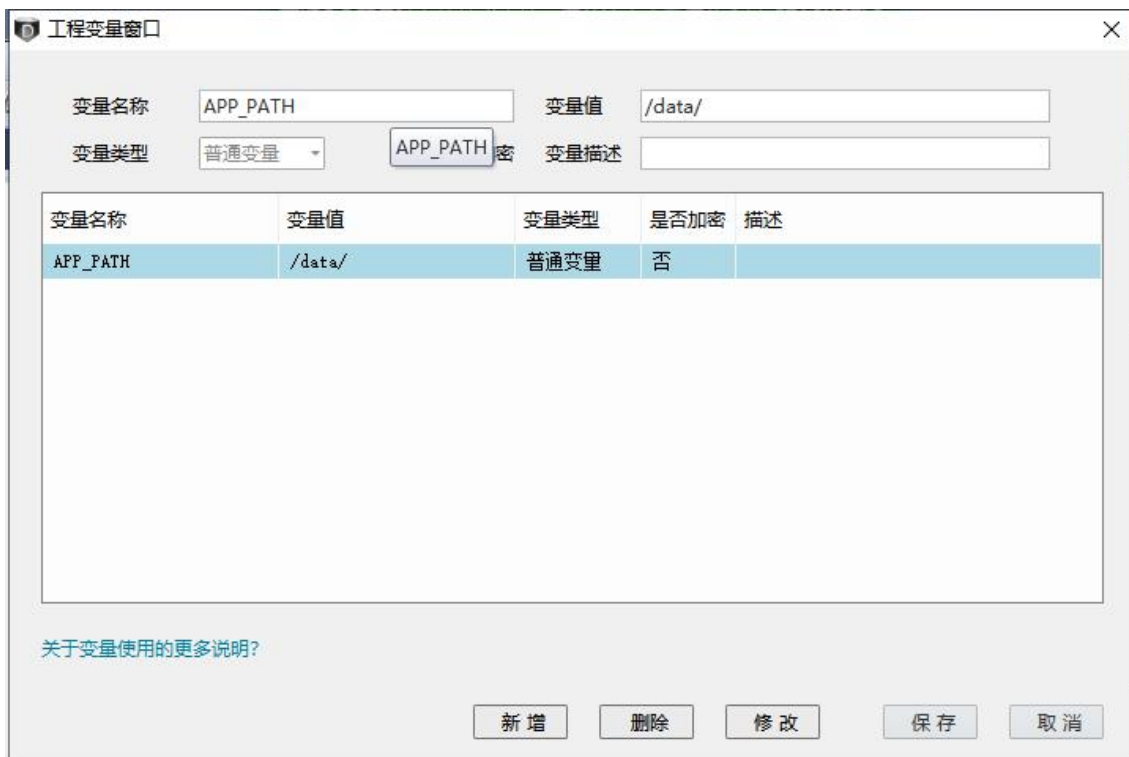
3.4.1 工程节点

为了从工程项目角度上对调度流程进行分类管理，在系统初始化部署的阶段ADMIN程序预设了(具体设置方式请参见《TASKCTL Admin 平台管理》)工程节点。通过该节点的右键菜单，可以快速建立属于该工程的控制容器。如下图所示：



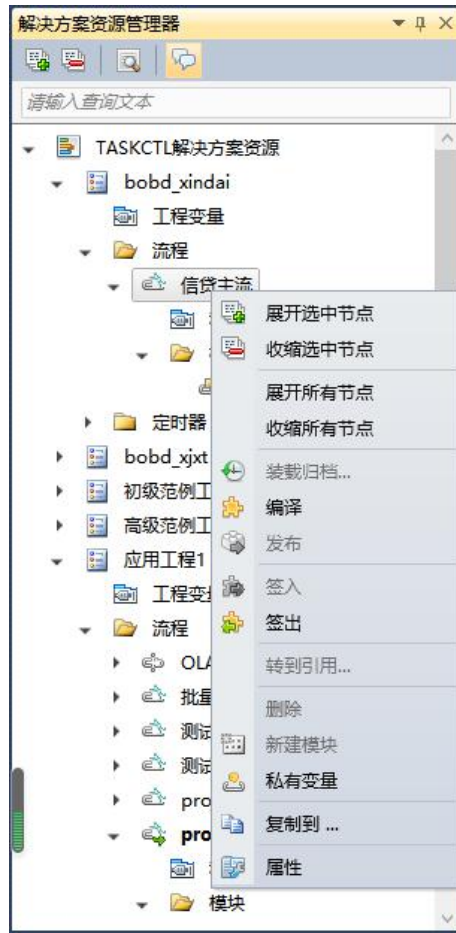
3.4.2 工程变量节点

从 v7.0+版本开始支持工程级别的变量，以帮助用户更好的管理各种范围（平台级、工程级、控制容器级）的变量。双击“工程变量”节点弹出所属工程的变量编辑窗口。如下图所示：




3.4.3 流程节点

流程信息配置是作业调度项目实施的主要内容，流程包含模块信息及流程变量信息。通过资源树右键菜单，可进行该流程节点的维护操作，以及快速新增模块单元。如下图所示：



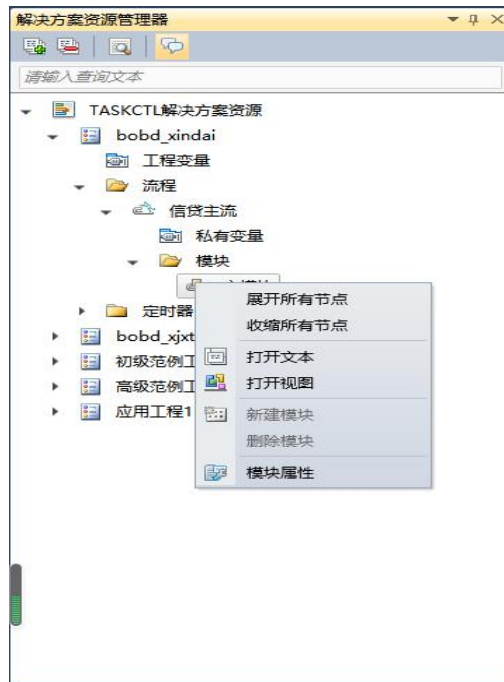
3.4.4 流程变量节点

双击流程变量节点，或选资源树上的流程节点后点击工具栏“流程变量”按钮，即打开该流程变量编辑窗口。如下图所示：



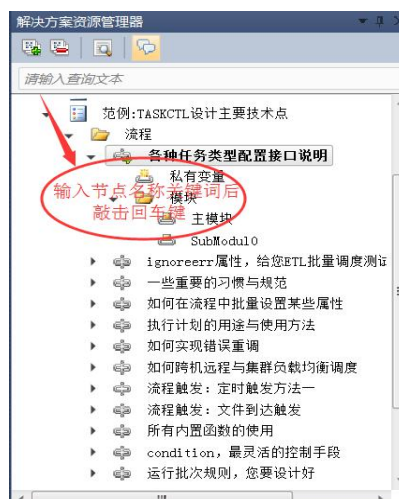
3.4.5 模块节点

模块是调度流程设计中最基本的单元，每个调度流程有且仅有一个主执行模块，但可存在多个子模块。通过模块与模块之间的有效引用来实现完整的调度流程。模块存放在流程节点的模块文件夹里面，双击模块节点将快速导航到该模块的“模块设计器”。该节点右键菜单也提供了对此模块的维护操作。如下图所示：



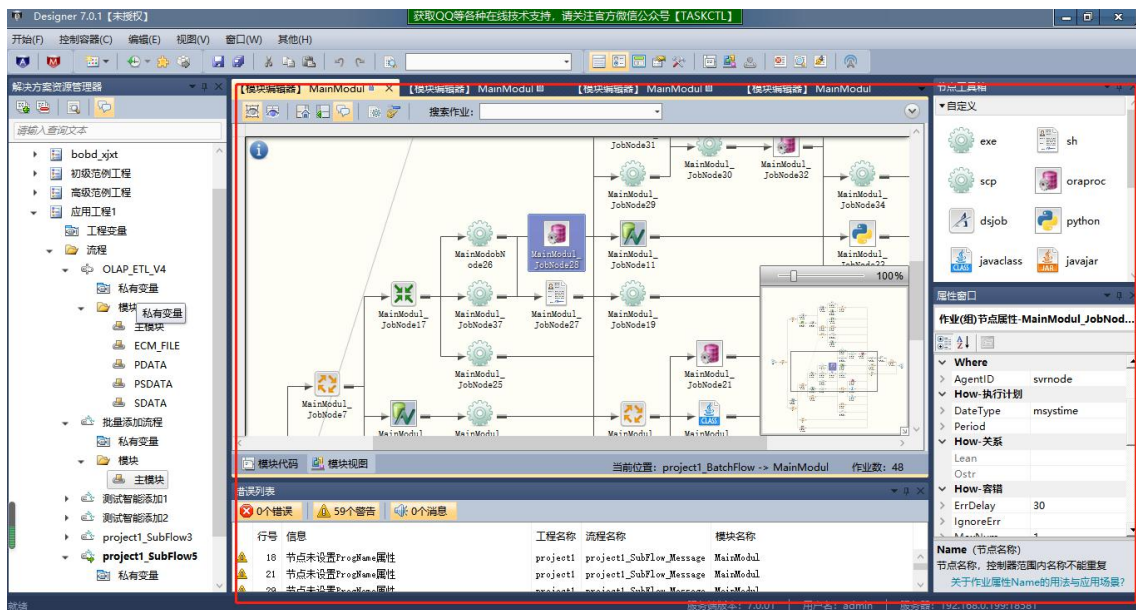
3.4.6 资源节点搜索框

在搜索框内输入节点名称关键词后敲击回车键，可在资源树上快速定位到下一个匹配的节点。支持大小写敏感的模糊匹配。如下图所示：



3.5 模块设计窗口

模块作为最基本的流程开发设计单元，平台为每个模块提供独立的流程设计窗口来进行设计，双击资源管理器的模块节点可打开对应的设计窗口。如下图所示，由“模块代码窗口”、“模块视图窗口”结合属性窗口、节点工具箱和开发信息输出区，构成了流程设计器。



通过拖拽节点工具箱作业类型组件到图形节点位置或节点关系数据，以及对节点属性框里属性的编辑，来实现对模块的可视化设计模块。也可以随时切换到模块代码设计窗口，直接通过编码的方式来完成模块的代码设计模块。

3.6 流程节点工具箱

流程配置的主要思路是对作业进行节点化抽象。因此 TASKCTL 通过系统缺省节点，如：root 节点，begin 节点，end 节点，以及串并组节点，把用户自定义的作业按照一定的序列排列，来描述作业之间的关系。



如上图所示：节点工具箱按照“平台所有节点”，“串并组”，“系统缺省”，“自定义”四种类别进行分类排列。

3.7 属性窗口


通过点击界面上“工程”，“流程”，“模块”，“作业”等资源对象，系统实时地把对象属性展示到属性框里面。在传统应用软件中，编辑对象的属性通常要利用弹出模态对话框来进行编辑，比较麻烦。Designer 利用浮动窗口可使流程设计更加扁平化。




如上图所示：可直接在属性框里编辑 TASKCTL 资源对象属性，系统将实时更新页面上的对象信息。这使得调度流程的开发更加便捷直观。

3.8 消息窗口

3.8.1 错误输出窗口

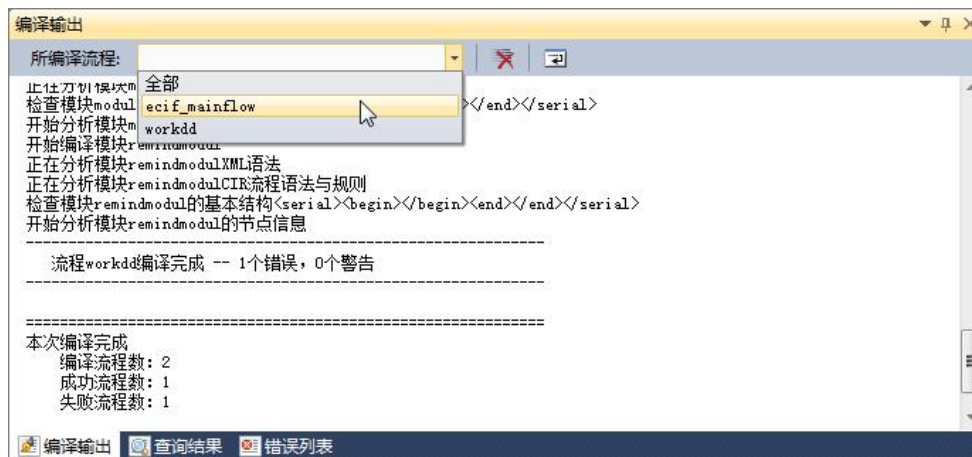
点击工具栏的“代码错误列表”按钮，或者按“Ctrl+W”呼出该窗口。如下图所示：该窗口分为“错误”，“警告”，“消息”。实时的展示了模块编码过程中出现的问题。双击表格的数据行，可快速定位到所指定的代码位置。如下图所示：



另外，点击如“ 0个警告”类似的按钮可对输出信息进行筛选显示。

3.8.2 流程编译输出窗口


在流程完成编译命令之后将自动弹出该窗口，展示了流程的编译输出信息，通过所编译流程下拉列表框，可对编译输出信息进行筛选。如下图所示：



3.8.3 查询结果输出窗口

在“查找与替换”窗口中找到的结果将显示到本窗口中，通过双击结果数据项，将自动导航到相应文档位置。

3.9 查找与替换

您可以快速搜索特定单词或短语在模块代码设计窗口出现的位置。点击工具栏  “从文件中查找”按钮，或者按“Ctrl+F”，随即将打开“查找与替换”窗口。如下图所示：



在查找内容框中，键入需要查找的文字，设置好查找范围。点击“查找下一个”按钮开始进行查找。查找成功后，系统将自动定位到代码位置。如果没有找到，系统将会提示“没有找到指定文本”。默认是不需要区分大小写的模糊查找，可通过勾选“大小写匹配”或“全字匹配”来限定查找模式。

可以在“查找”和“替换”选项卡片中进行快速切换，点击“**替换**”按钮进入文本替换模式。下面的步骤将介绍如何进行代码文本替换：


1. 在“查找内容”文本框中键入需要替换的内容。
2. 在“替换为”文本框中键入替换为目的内容。
3. 选择好查找范围，只能替换签出流程的模块代码内容。（有关“签出”将在以后的章节进行介绍）
4. 点击“替换”按钮将替换下一个查找到的内容，点击全部替换将批量替换所有查找到的内容。
5. 同样的，可通过勾选“大小写匹配”或“全字匹配”来限定需要替换内容的查找模式。

4 流程开发设计

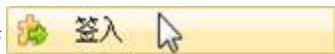
4.1 流程签入与签出

在多用户流程开发过程中，为了防止其它用户同时更改同一流程，引起流程开发混乱。TASKCTL 引入了“流程签入签出”的概念。当用户把流程签出后，即获得该流程最新的数据信息和开发权限，才允许对该流程及所属模块、流程变量进行编辑和删除等操作。如果试图签出已被其它用户签出的流程时，系统将提示“该流程已被占用，无法签出”。另外，如果需要编译流程，也应先签出流程。关闭系统的时候，将会自动签入所有已签出的流程。

若要签出流程，请执行以下操作：

1. 选中资源树上的流程节点。
2. 在流程节点的右键菜单中点击“”按钮。

若要签入流程，请执行以下操作：


1. 选中资源树上的流程节点。
2. 在流程节点的右键菜单中点击“”按钮。

说明： 以下章节对流程的所有开发设计都是以“流程签出状态”为前提。

4.2 流程创建与删除




系统有多种新增流程的方式。通常情况下，我们在资源树上进行快速新增。也可以通过“添加新项”窗口选择“流程”进行新增。

若要添加缺省流程，请执行以下操作：

1. 选择资源树上的工程节点。
2. 在工程节点的右键菜单点击“”按钮。系统

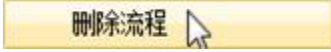
将为生成的流程自动分配一个新流程名称。

若要添加自定义流程，请执行以下操作：

1. 点击工具栏上的“”按钮打开“添加新项”窗口。
2. 选择“”，再选择所属工程后键入所需新增流程的名称。
3. 点击“”按钮完成新增流程操作。

如果流程名称重复或者没有键入流程名称，则不能完成新增操作。


若要删除流程，请执行以下操作：

1. 在资源树上选中流程节点。
2. 在流程节点的右键菜单点击“”按钮。
3. 在弹出的确认删除对话框里点击“是”按钮完成删除操作。

4.3 流程编辑

4.3.1 流程基本属性编辑

若要编辑流程属性，请执行以下操作：

1. 在资源树上选中流程节点。
2. 点击工具栏的“”或按快捷键“F9”呼出属性窗口。
3. 在属性项后面的文本框里输入新的属性值。如果属性文本框有下拉按钮，则表示该属性可进行选择。
4. 敲击回车按钮以确定输入。

特别需要注意的是“启动模块”和“是否子流程”这两个属性。选择“启动模块”以确定流程运行的第一个模块，选择“是否子流程”来确定是否运行流程被其他流程所引用，v5.0版本及以前新增默认不是子流程，v5.1+版本开始新增默




认为子流程。

4.3.2 变量管理

为增强流程的可移植性、灵活性，配置变量的引入是不可缺少的。TASKCTL在变量管理方面有一套完整的机制。它将变量分为三类，分别为：

- 平台全局常量：在 Admin 中统一定义管理，所有控制器都可以访问，但不可修改。
- 工程变量(v7.0+)：每一个工程都有自己的一套工程范围变量，仅在工程范围内有效。
- 私有变量：用户可以定义控制器级别的私有变量，以适应更灵活的变量引用需求。


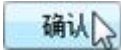
若要通过 Designer 进行私有变量管理，请执行以下操作：

1. 在“资源管理器”中双击“私有变量”节点。即打开“变量编辑器”文档窗口。
2. 选择“私有变量”选项卡片进入私有变量编辑模式。
3. 键入“变量名称”、“变量值”，选择“变量类型”及“是否加密”选项，点击“”新增按钮完成私有变量新增操作。
4. 选中表格中的流程变量，点击“”删除按钮完成私有变量删除操作。
5. 选中表格中的流程变量，编辑流程变量值，点击“”修改按钮完成私有变量更新操作。

4.3.3 模块新增、修改、删除


跟新增流程类似，可通过资源树及“添加新项”窗口进行新增模块操作。这里不再累述。特别地，可在模块代码中选择代码片段新增子模块。

若要通过代码片段新增子模块，请执行以下操作：


1. 确保模块所属流程为签出状态。
2. 在模块代码中，框选合法格式的代码片段。
3. 在选好的代码片段上右键菜单点击“”菜单项。即弹出“新建模块”对话框。
4. 在“新建模块”对话框中键入模块名称后点击“”按钮。

操作完成后将转到新生成的模块代码设计器窗口，同时系统把框选的代码片段替换为指向新生成模块引用的 `include` 节点。

若要编辑模块属性，请执行以下操作：

1. 确保模式所属流程为签出状态。
2. 在资源树上选中模块节点。
3. 点击工具栏的“”或按快捷键“F9”呼出属性窗口。
4. 在属性项后面的文本框里输入新的属性值。
5. 敲击回车按钮以确定输入。

若要删除模块，请执行以下操作：

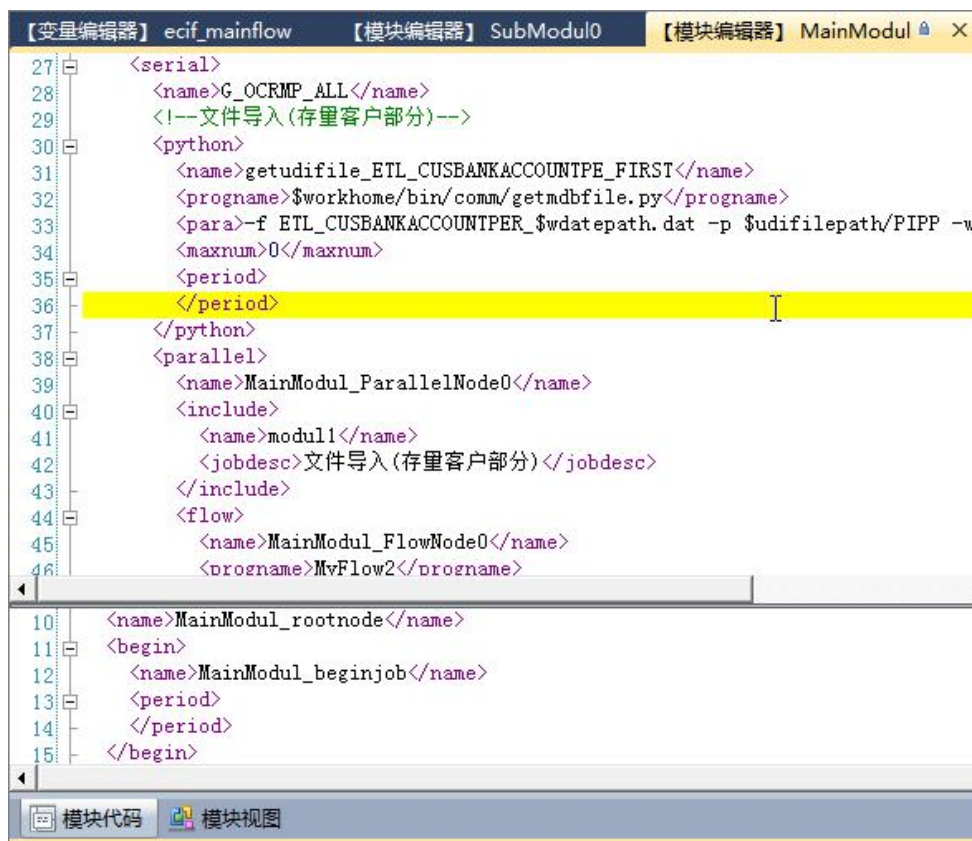
1. 在资源树上选中流程节点。
2. 在流程节点的右键菜单点击“”按钮。
3. 在弹出的确认删除对话框里点击“是”按钮完成删除操作。

需要注意的是：如果模块有被引用的情况，删除该模块后，也应删除引用模块的地方。

说明： 可通过编译流程来确定模块引用错误的位置。

4.3.4 模块代码编辑

模块代码设计是 TASKCTL 对调度流程设计提出的全新概念。TASKCTL 模块语法类似于 XML 结构化标签语言。掌握好代码开发对提高流程开发效率具有举足轻重的作用。



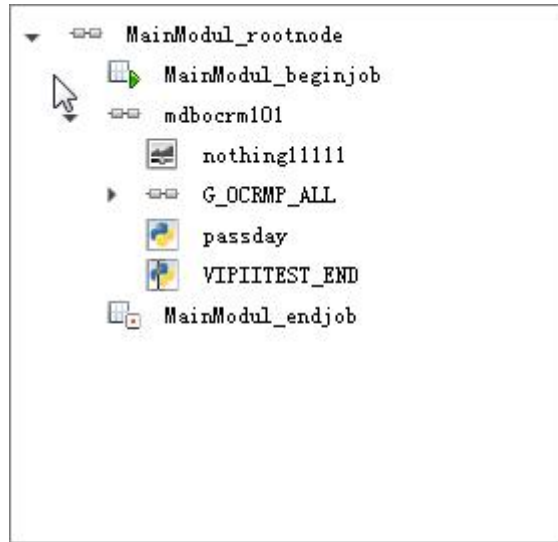
```
27 <serial>
28   <name>G_OCRMP_ALL</name>
29   <!--文件导入(存里客户部分)-->
30   <python>
31     <name>getudifile_ETL_CUSBANKACCOUNTPE_FIRST</name>
32     <progrname>$workhome/bin/comm/getmdbfile.py</progrname>
33     <para>-f ETL_CUSBANKACCOUNTPER_$wdatepath.dat -p $udifilepath/PIPP -w
34     <maxnum>0</maxnum>
35     <period>
36   </period>
37   </python>
38   <parallel>
39     <name>MainModul_ParallelNode0</name>
40     <include>
41       <name>modul1</name>
42       <jobdesc>文件导入(存里客户部分)</jobdesc>
43     </include>
44     <flow>
45       <name>MainModul_FlowNode0</name>
46       <progrname>MvFlow2</progrname>
47
10 <name>MainModul_rootnode</name>
11 <begin>
12   <name>MainModul_beginjob</name>
13   <period>
14 </period>
15 </begin>
```

上图示例了一个模块的部分代码，可以看出模块代码设计器是一个标准的文本编辑工具。具有“拷贝”、“剪切”、“粘贴”、“撤销”、“重做”等文本操作。还具有“关键字高亮”，“行号”，“缩进”，“格式化文本”，“代码注释”等传统高级语言编程工具的特征。可以对代码窗口进行拆分，方便随时参考。

4.3.5 模块图形编辑

和模块视图一起同步展示了调度流程的节点关系。模块图形编辑窗口由两部分组成：“作业关系图”以及“作业树”。通过“作业关系图”，可以很直观的看到

作业之间的串并关系。如果一个模块的作业关系图太大而不好展示的情况下，也可以对作业关系图进行缩放展示。通过拖动节点工具箱里的 TASKCTL 节点到节点树来完成模块的可视化设计。



如上图所示：由 root 节点作为根节点，begin 节点作为起始节点，end 节点作为结束节点，把用户自定义作业通过串并组等系统缺省节点，按照 TASKCTL 定义的一系列规则来构成节点树。以下是模块设计在节点树上的设计规则：

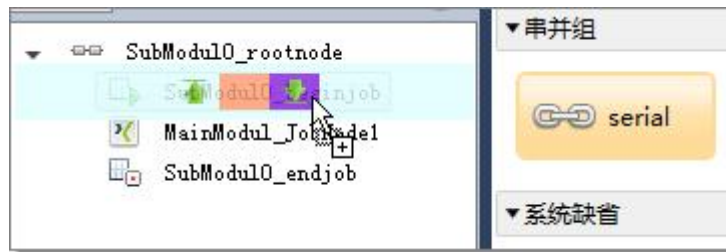
- ✓ 节点树有且仅有一个 root 根节点，其包含一个 begin 节点和一个 end 节点，其他所有节点都只能在 begin 节点之后，end 节点之前。
- ✓ 串并组节点和 include 节点是特殊的 TASKCTL 节点，分别用来表达作业之间的串行、并行、引用包含关系。可允许其他 TASKCTL 节点作为其子节点。
- ✓ flow 节点、nulljob 节点和自定义节点是最基础的 TASKCTL 节点，不允许再包含其他 TASKCTL 节点，是作业树的叶节点。

4.3.6 串并关系结构

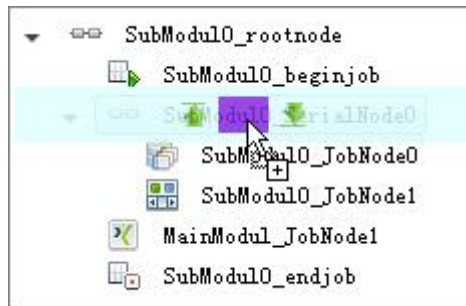
串并关系调度是作业之间的主要关系调度，通过串并关系可以实现所有作业无关的关系调度以及决大多数作业依赖调度。串并关系通过串并组节点及其组节点灵活嵌套来实现。

若要通过图形设计器实现节点之间的串行关系，请执行以下操作：

1. 在节点工具箱中拖拽“serial 节点”到作业树。如下图所示



2. 拖拽作业到“serial 节点”里面。如下图所示：

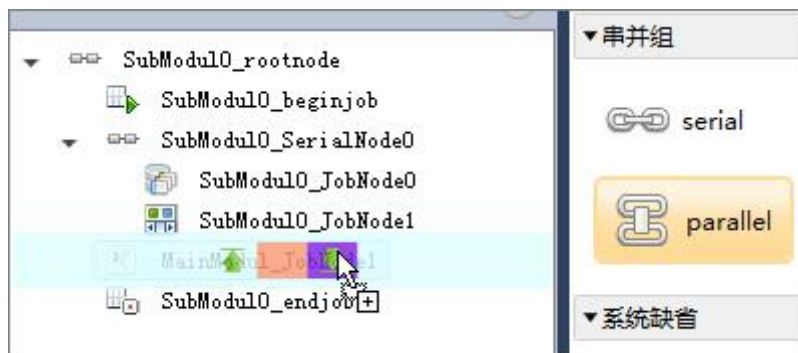


3. 完成作业串行关系设计。

完成上面增加串行节点的示例后，仔细的读者可能已经发现“模块视图”并没有体现“MainModul_SerialNode2”，是因为在“模块视图”里面，两个相邻节点之间已经隐含了串行的关系。

若要通过图形设计器实现节点之间的并行关系，请执行以下操作：

1. 在节点工具箱中拖拽“parelled 节点”到作业树。如下图所示

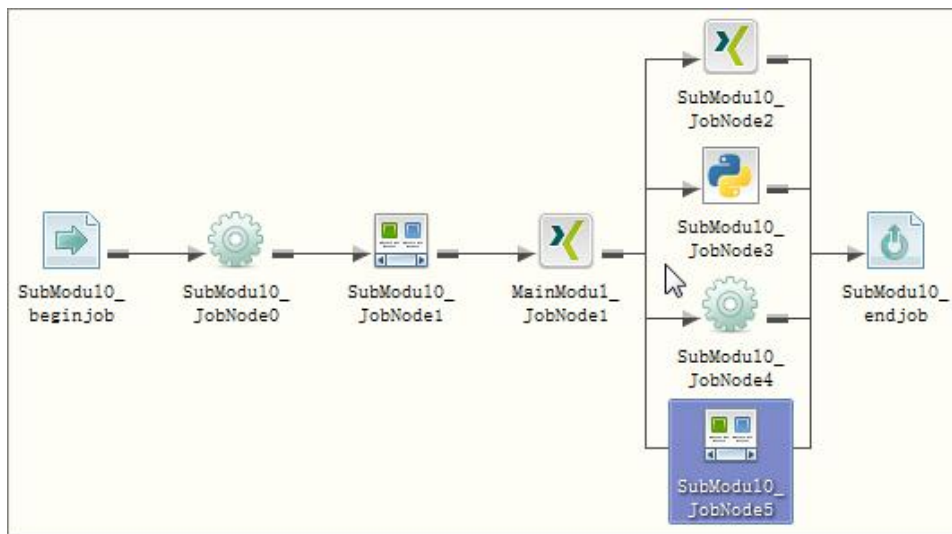


2. 拖拽作业到“parallel 节点”里面。如下图所示：



3. 完成作业并行关系设计。

完成上面增加串行节点的示例后，“模块视图”将把节点之间的并行关系显现出来。如下图所示：




4.3.7 节点属性编辑

由于作业的属性相对较多，为了方便我们熟悉作业属性的操作，TASKCTL 按照作业属性特征，按用途分为“**What** --基本属性”，“**How** -- 控制属性”，“**Where** -- 目标主机属性”和“**Misc** -- 杂项属性”等。另外某些属性还具有复合（继承、缺省、自定义）特征。

4.3.7.1 继承属性 (vInherit)

在流程控制文件配置中，根据作业（组）节点的父子关系，对于父节点的很多属性子节点都可以缺省继承。

若要设置属性为可被子节点继承，请执行以下操作：

1. 只有属性框前面有“”所示的三角形按钮的属性，才可以设置为可继承。
2. 选择该属性的 flag 为“vfInherit”，如下图所示：



若不需要继承父节点属性，请执行以下操作：

1. 选择该属性的 flag 为“vSelf”，如下图所示：




2. 在该属性的 Value 设置为自定义的值，如上图所示 Value 为“selfValue”。

4.3.7.2 缺省属性 (vfDefault)

属性节点的缺省值目前可以分为两类：系统缺省值、设置缺省值。

- ✓ 系统缺省值：指平台固定设置的缺省值。比如作业最大重复次数‘maxnum’属性，系统固定缺省为 1，即执行次数为一次，类似的还有 DateType 等，不过这些缺省都只是在 root 根节点才能看到，在子节点上默认是继承。
- ✓ 设置缺省值：这种缺省值不是自身决定的，而是设置的。比如 para 属性。我们在流程控制文件中的缺省值是来源于在 ctlconf.xml 文件中插件的设置。目前可支持设置缺省值的属性有 para、exppara、SuccessV、ErrorV、FailedV、WarningV 等属性，缺省值均来源于 ctlconf.xml 文件中插件的设置，这些值是在 Admin 中在作业类型中配置的。

若要设置属性为缺省值，请执行以下操作：

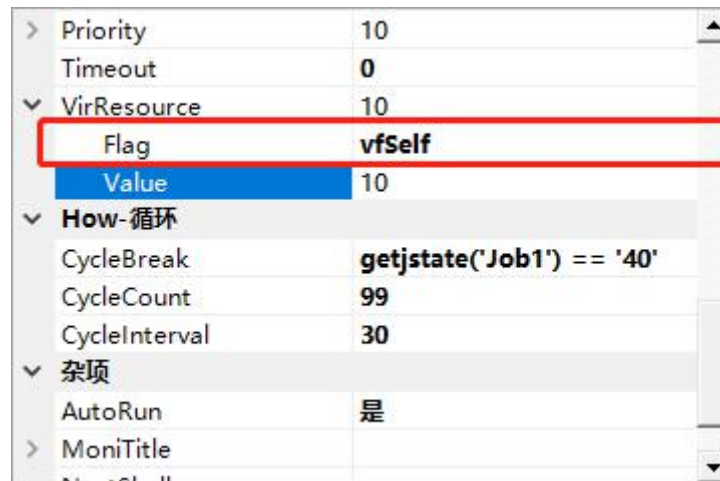
1. 只有属性框前面有“”所示的三角形按钮的属性，才可以设置为缺省值。
2. 选择该属性的 flag 为“vfDefault”，如下图所示：



4.3.7.3 自定义属性 (vfSelf)

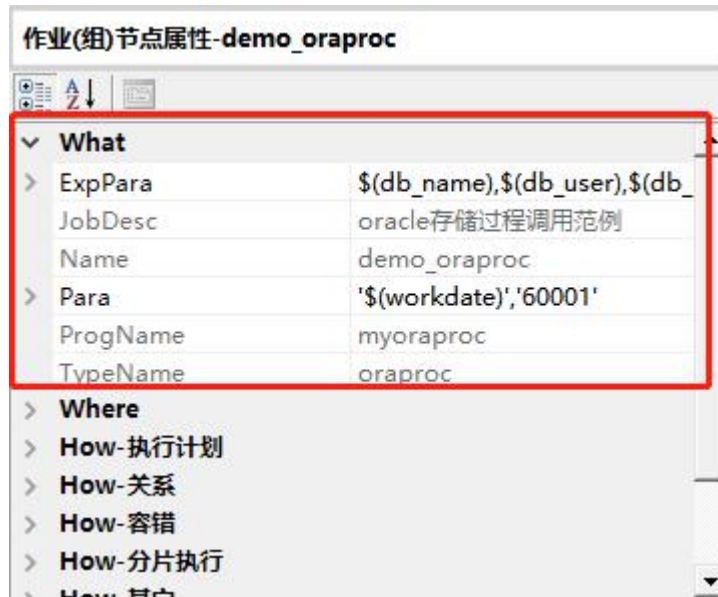
作业属性如果不想引用继承或缺省值，可以显式设定作业的自定义值，请执行如下操作：

- 1、设置复合属性的 **flag** 为 **vfSelf** 选项
- 2、在 **Value** 中设置作业的自定义属性值



4.3.7.4 What -- 基本属性 (name / progame / para / exppara / jobdesc)

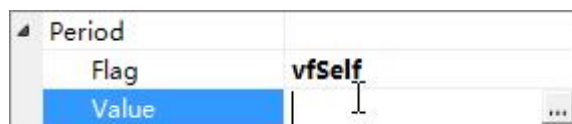
- ✓ Name -- 作业名称，在控制容器范围唯一
- ✓ Progame -- 作业程序路径
- ✓ Para -- 作业程序入口参数
- ✓ Exppara -- 作业程序环境参数
- ✓ Jobdesc -- 作业描述信息



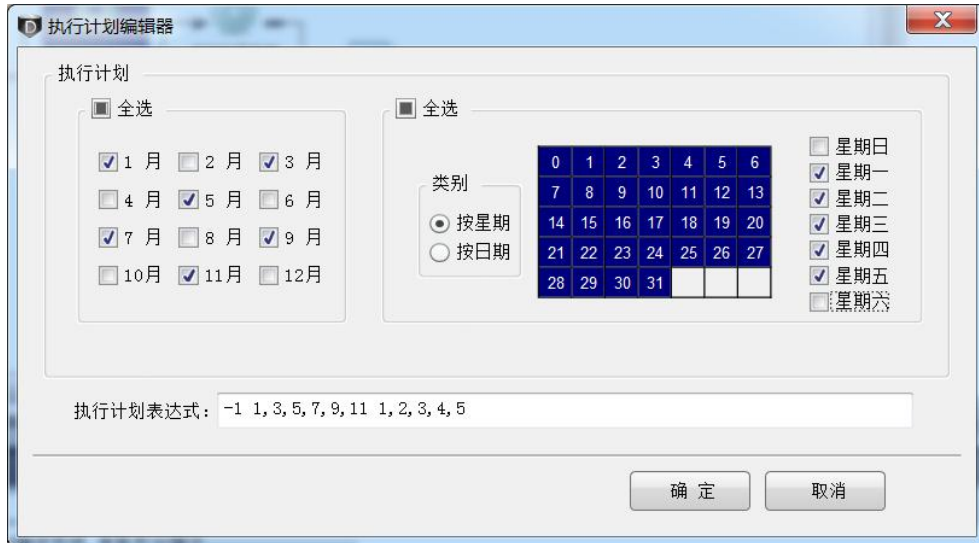
4.3.7.5 How -- 执行计划(period / datatype)

作业的执行计划决定作业的可调度执行周期或者特定可调度日期。对于执行计划，通过配置作业（组）的执行计划属性（period）来实现。若要通过属性框设置“执行计划”属性，请执行以下操作：

1. 选择 Period 属性的 Flag 选项为“vSelf”，如下图所示：



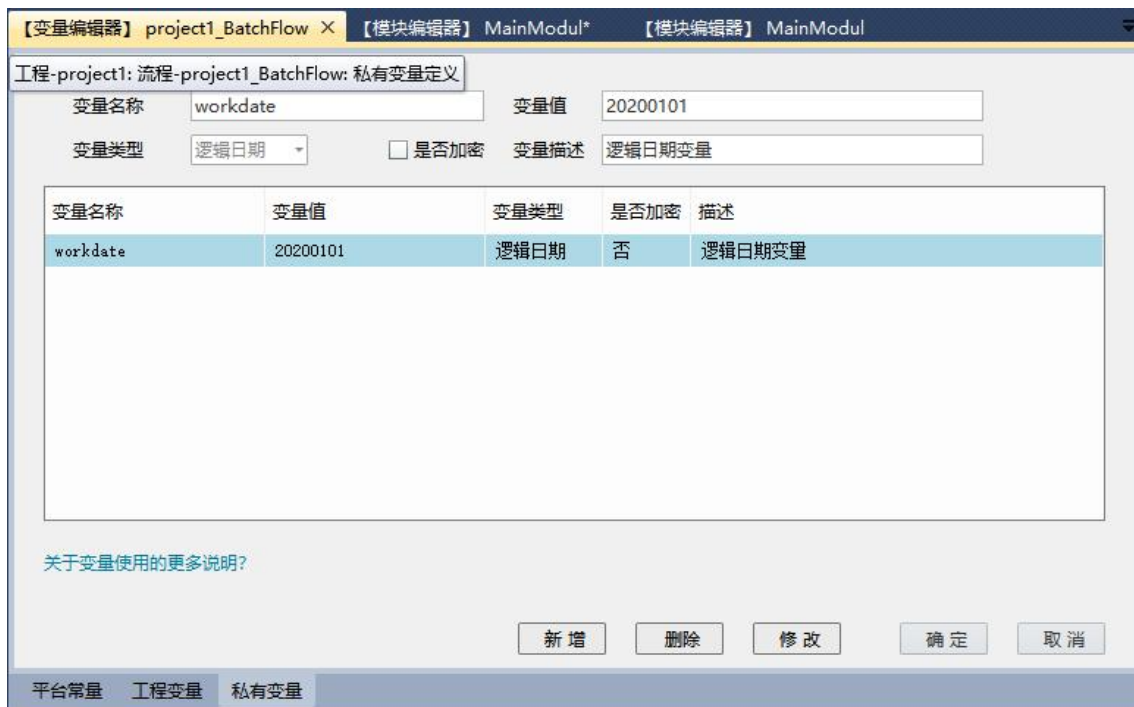
2. 点击 Value 属性框中的“...”，打开“执行计划属性编辑器”。
3. 在“执行计划属性编辑器”中设置好执行计划表达式，如下图所示：



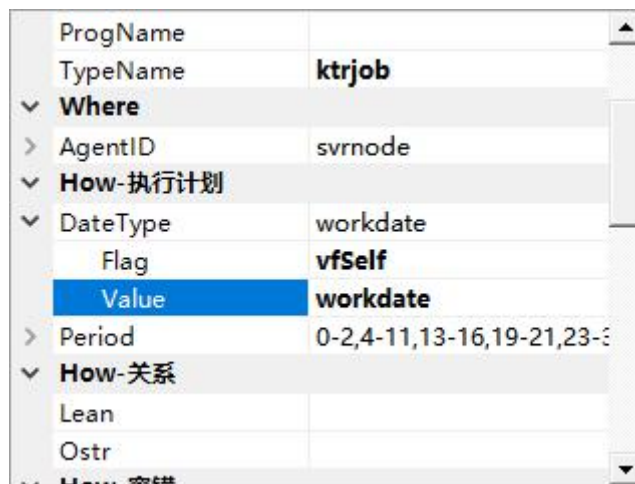
4. 点击确定按钮完成操作。

完成上述操作后，作业将在排程好的自然日期当天执行，不满足的日期将跳过不执行。如果需要在逻辑日期下也应用到 **period** 属性。那么请执行如下操作：

1、设置逻辑日期变量如下图所示：



2、通过作业的 **datatype** 属性引用定义好的逻辑日期变量，如“workdate”：



3、完成好逻辑日期排程计划设置。系统将以逻辑日期来判定当天是否执行。

说明: `period` 和 `datetype` 属性仅在流程下可用, 在定时器中不可用。

4.3.7.6 How -- 依赖 (lean)

串行节点配置实际上是依赖关系配置, 只是方便表示了一组作业的相互依赖关系。但对于一些特别的作业, 不方便直接用串行组表达。就可以通过作业依赖属性 (`lean`) 来补充。

若要通过属性窗口设置依赖属性, 请执行以下操作:

1. 在作业的 `lean` 属性值中输入依赖作业的 `name` (注意这里只能是在同一流程中作业, 如果要跨流程, 请用发送事件作业), 如果有多个, 中间用逗号隔开。

2. 敲击回车键以完成操作。

不过, 我们建议少使用 `lean` 属性, 尽量采用串行组来实现作业之间的依赖, 毕竟串行组配置的可读性要高。

4.3.7.7 How -- 互斥 (ostr)

互斥表示作业之间不能同时运行。将通过作业的 `ostr` 属性实现。若要通过属性窗口设置互斥属性, 请执行以下操作:

- 1.在作业的 `ostr` 属性值中输入互斥作业的 `name` (注意这里只能是在同一流程中作业), 如果有多个, 中间用逗号隔开。
- 2.敲击回车键以完成操作。

4.3.7.8 How -- 容错 (`ignoreerr / maxnum / errdelay`)

通过设置作业的容错机制, 从调度系统层面增强作业的健壮性。

✓ Ignoreerr -- 是否应用容错机制

对于有些作业, 出错后流程仍需要继续向后运行, 及不影响依赖于它的作业调度。对于这种需求, 将通过 `IgnoreErr` 属性来设置。

若要通过属性窗口设置“是否忽略错误”, 请执行以下操作:

1. 选择 `IgnoreErr` 属性的 `Flag` 选项为“`vSelf`”, 如下图所示:

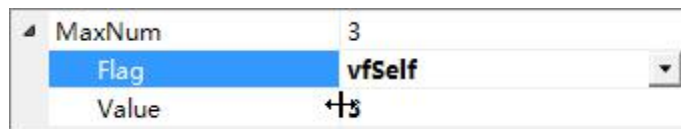


2. 在 `Value` 属性框中选择是否忽略。
3. 敲击回车键以完成操作。

✓ Maxnum -- 最大错误重试次数

作业运行错误后, 根据配置最大重复次数(`maxnum`)决定尝试执行次数, 直到成功为止。若要通过属性窗口设置“重复次数”, 请执行以下操作:

1. 选择 `Maxnum` 属性的 `Flag` 选项为“`vSelf`”, 如下图所示:



2. 在 `Value` 属性框中键入所需“重复执行次数”的值, 该值为整数。
3. 敲击回车键以完成操作。

说明: 如果 `maxnum` 设置为 3, 作业第二次重试却成功了, 那么执行完第二次, 作业就通过了, 不会再执行第三次; 如果执行完了第三次, 依然没有成功, 则根据下面的忽略错误决定走向。当 `maxnum` 设置为 0 时, 表示不限制重试次数, 知道成功为止, 每次重试其间的间隔由平台的配置决定。

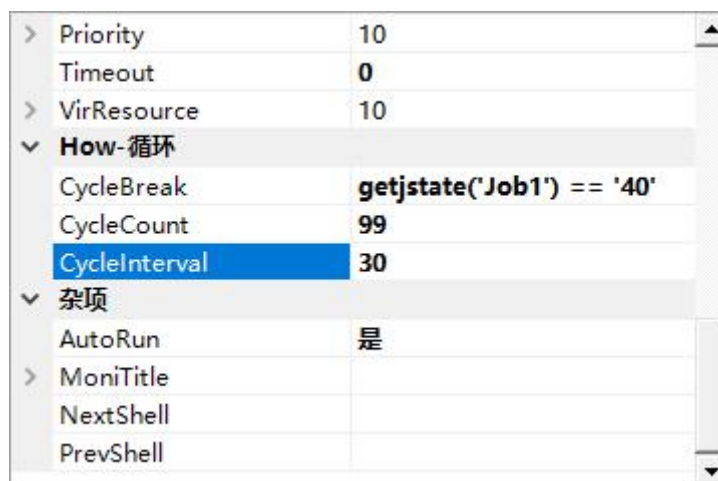
- ✓ Errdelay (v6.0+) -- 每次错误后，延迟重试的时间间隔（秒）

4.3.7.9 循环(cyclecount / cyclebreak / cycleinterval)

对于一些作业或模块希望循环执行，可以通过设置节点的循环属性来实现。

- ✓ cyclecount (v7.0+) -- 循环次数， v7.0-之前版本为 cycle
- ✓ Cyclebreak (v7.0+) -- 循环退出条件，满足条件则退回循环模块
- ✓ Cycleinterval (v7.0+) -- 每次循环结束后的等待间隔时间

参见如下示例：



> Priority	10
Timeout	0
> VirResource	10
▼ How-循环	
CycleBreak	getjstate('Job1') == '40'
CycleCount	99
CycleInterval	30
▼ 杂项	
AutoRun	是
> MoniTitle	
NextShell	
PrevShell	

4.3.7.10 How -- 返回信息(successv / errorv / failedv / warningv)

- ✓ successv -- 作业成功状态返回信息
- ✓ Errorv -- 作业错误状态返回信息
- ✓ Failedv -- 作业失败状态返回信息
- ✓ Warningv -- 作业警告状态返回信息

作业的返回信息用来判断该作业调用成功与否。用数字来匹配作业程序的退出码。可使用连串数据：成功返回值 0-10，警告返回值 11-30 等。注意：用户自

定义的返回值只能是在 0-100 之间。

v 7.0+ 新增支持返回信息匹配作业程序的输出信息。如: [ORA-]可匹配 oracle 程序的错误特征码。



4.3.7.11How -- 分片执行(issplit / splitcount)

在一些分布式计算的作业中，通常会用到分片技术，可以通过设置作业的分片执行属性来实现。

- ✓ Issplit (v7.0+) -- 是否应用分片执行策略
- ✓ Splitcount (v7.0+) -- 分片的个数或分片个数表达式

参见如下示例：





设置了分片属性的作业，将以“★”角标来特殊标识：

4.3.7.12 How -- 条件分支 (condition)

作业 **condition** 属性是功能非常强大的属性 (**condition** 属性应用可以参见自带的高级范例《强大的自定义控制-Condition》)，通过它可以实现作业各种不确定运行条件调度。并可以实现条件分支调度。

若要通过属性窗口设置“条件分支”，请执行以下操作：

1. 选择 **Condition** 属性的 **Flag** 选项为“vSelf”，如下图所示：

Condition	if getjresult(GetData1)==2 C1
Flag	vfSelf
Value	if getjresult(GetData1)==2

2. 在 **Value** 属性框中键入所需“条件分支”的值，该值为表达式。
3. 敲击回车键以完成操作。

4.3.7.13 How -- 优先级 (priority)

作业 **priority** 属性用于控制并发的优先执行顺序。值越小表示优先级越高。

若要通过属性窗口设置“优先级”，请执行以下操作：

- 1、选择 **priority** 属性的 **Flag** 选项为“vSelf”，如下图所示：

> How-容错	
> How-返回信息定义	
▼ How-分片执行	
IsSplit	是
SplitCount	getClusterCount("MAgent")
▼ How-其它	
> Condition	
▼ Priority	1
Flag	vfSelf
Value	1
Timeout	0
> VirResource	10
> How-进程	

- 2、在 Value 属性框中键入所需优先级值，该值为整数。
- 3、敲击回车键以完成操作。

4.3.7.14 How -- 超时失败 (timeout)

作业 timeout 属性用于控制作业最大运行时间，单位为秒。当作业超时后，状态为失败，流程将暂停执行。当超时失败值为 0 时，表示不应用该属性。



4.3.7.15 How -- 虚拟资源 (virresource)

作业 virresource 属性用于控制并发的优先执行权重。值越大表示权重越高。若要通过属性窗口设置“虚拟资源”，请执行以下操作：

- 2、选择 virresource 属性的 Flag 选项为“vSelf”，如下图所示：



- 2、在 Value 属性框中键入所需虚拟资源值，该值为整数。
- 3、敲击回车键以完成操作。

4.3.7.16 Misc -- 自动执行 (autorun)

默认情况下，作业只要满足调度条件后就会自动执行。如果我们需要对作业进行人为的确认后再执行，那么可以设置 autorun 为“否”。当流程运行到该作业时，流程会暂停。直到进行确认执行后，流程才会继续运行。



设置了非自动运行的作业，将以“审核员”角标来特殊标识：

4.3.7.17 Misc -- 预处理与后置处理 (prevshell / nextshell)


- ✓ prevshell -- 作业执行前的预操作脚本
- ✓ nextshell -- 作业执行后的操作脚本


可以实现一些与业务流无关的操作，比如创建目录，导入环境等。

4.3.8 保存模块

现在让我们保存在 Designer 中创建或编辑的模块。在 Designer 中，当中断工作或退出时必须“保存”模块，否则所做的工作将会丢失。“保存”后，该模块将会上传到服务器。如果是本地开发，将会保存到本地磁盘。以后可以打开该模块，再次进行修改操作。

若要进行保存模块的操作，请执行以下步骤：


1. 点击工具栏的“”按钮。
2. 完成保存操作。

也可以点击工具栏的“”全部保存按钮。对所有未保存的模块进行保存操作。

4.4 流程编译

流程编译是流程开发的一个关键性步骤，您需要通过流程编译来验证流程开发过程中的一些语法问题。


若要进行“流程编译”，请执行以下操作：

1. 点击工具栏的“”按钮或按快捷键 F6 开始编译流程。
2. 直到“编译输出”窗口提示“编译完成”后完成编译操作。

4.5 流程发布及归档

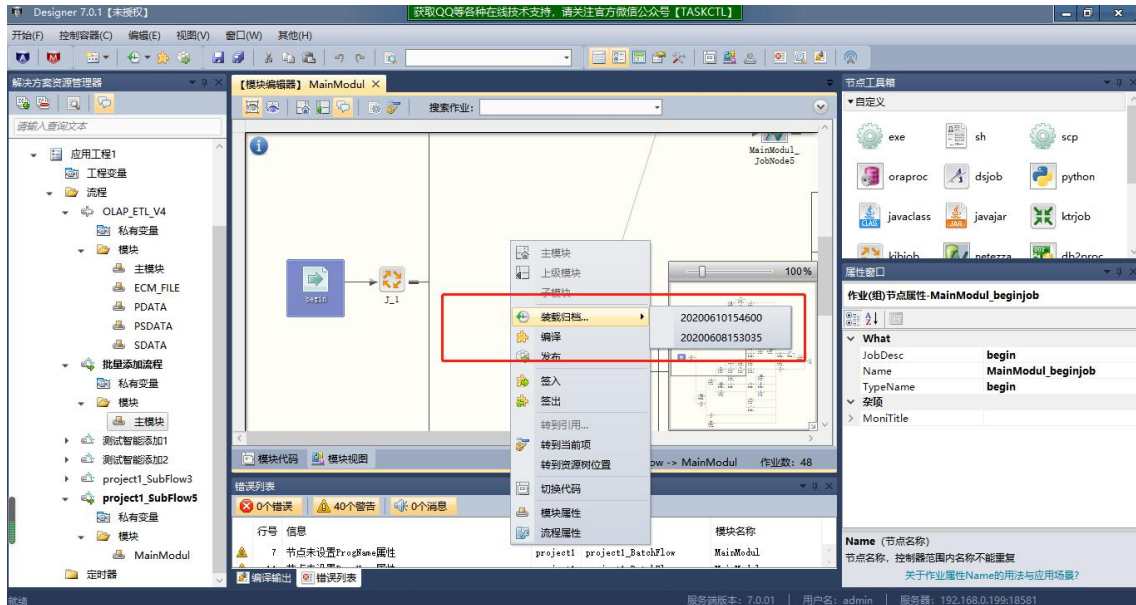
流程发布 (v7.0+)：流程编译通过后，就需要通过发布功能发布到 Monitor 环境进行运行监控了。(v7.0-之前版本编译后会自动发布到运行环境。)

若要进行“发布”，请执行以下操作：

3. 点击工具栏的“”按钮开始发布。
4. 直到弹出“发布成功”对话框，完成发布操作。



发布归档 (v7.0+)：每次发布成功后将产生一个归档文件，用户可以选择任意归档项装载之前发布的版本。




4.6 一些快捷操作技巧

4.6.1 如何快速在图形设计器中定位指定作业

Designer 提供了多种方式来定位作业，您可以在图形设计器工具栏里面的作业搜索框中输入节点名称关键词，系统自动弹出匹配的作业列表，选中列表项进行定位。如果您对该模块结构比较熟悉的情况下，也可以在作业树里面选中期望的作业进行定位。

4.6.2 如何快速在图形设计器中定位指定模块

在流程开发过程中，难免会遇到一个流程很多模块，多级引用的情况。Designer 提供了多种方式导航到流程下的指定模块的设计器。下面示例怎样利用“模块导航器”快速打开模块的设计器。

1. 点击模块设计器中工具栏的“”按钮，即打开模块导航器。
2. 点击模块导航器的“模块名称”按钮，可直接导航到所点击模块的设计器。
3. 也可以通过在“搜索模块”的下拉框中，输入模块名称的关键词，查

找到模块列表。通过点击列表项，同样达到模块导航的目的。

4.6.3 如何快速通过图形增删一个作业

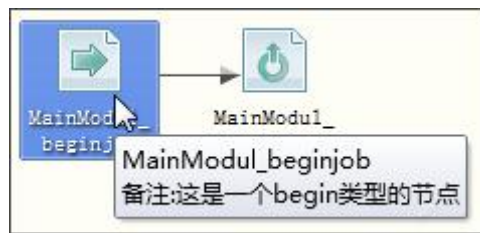
在遵循 TASKCTL 代码设计规则的原则上，通过拖拽作业工具箱的 CTL 节点到作业节点资源树或者作业节点位置，完成对作业的新增操作。若要删除作业，则仅需选中所要删除的作业。在其右键菜单中点击“删除”菜单项完成删除操作。

4.6.4 如何快速将大模块拆分更多的小模块

您可以在模块代码中框选所期望划分为子模块的代码片段(应遵循 TASKCTL 代码设计规则)，然后使用热键“Ctrl+P”弹出新建子模块窗口，在其中键入模块名称后确认以完成操作。

4.7 流程相关备注信息的重要性

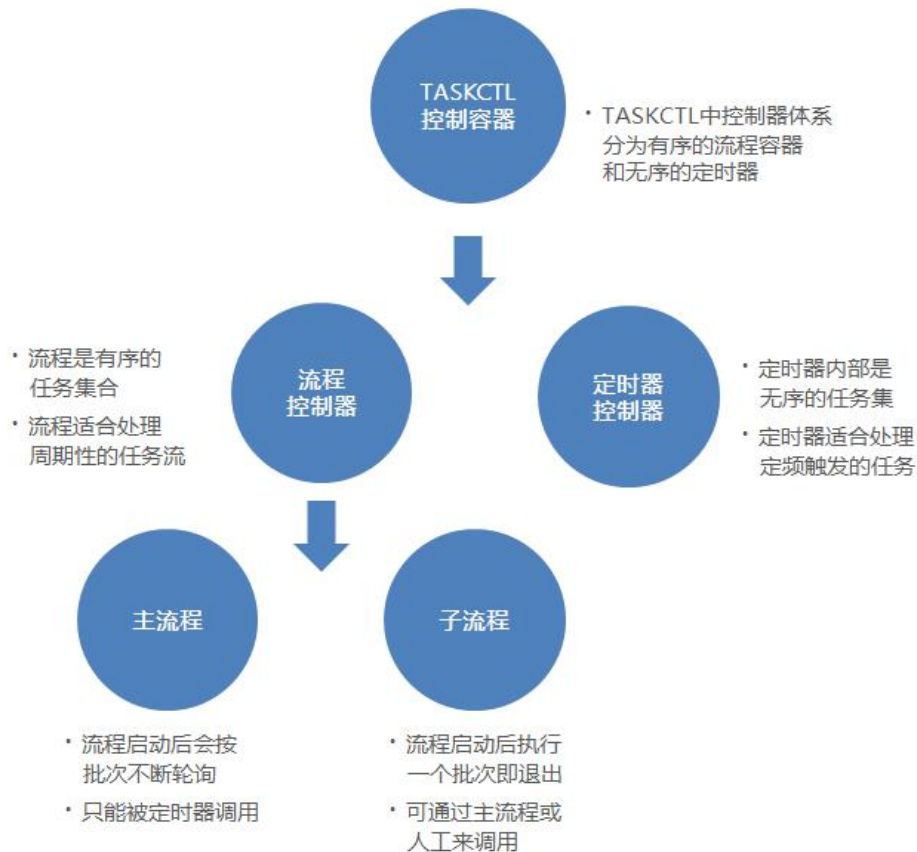
仔细的读者可能已经注意到，每个 TASKCTL 资源都有类似“备注”这样的属性。如流程对象有“流程描述”属性，模块对象有“模块描述”属性，作业对象有“JobDesc”属性。Designer 利用“备注”属性为对象在界面上展示提供了有效的方案。如下图所示：



合理利用好“备注”属性，有利于提高模块代码的可读性，提高调度流程开发设计的效率和正确性。

5 定时器开发设计

定时器在开发设计上,看上去几乎和流程一样,但是在理念上和流程又有这千差万别,都有串并组,都支持各种作业类型,那是因为平台本身 xml 组织语法都是一样,但不论是串并关系还是被删减的 Lean、Ostr 等设置依赖互斥的属性,在定时器中都是失效的,定时器中的所有作业是无序的,都是根据自己的 timingplan 属性决定什么时候运行,保留的并行组也仅仅是为了属性的继承,精简定时器的设计而已。流程和定时器主要用在不同的场景,定时器更加适合处理一些无序的定频的作业。下面这张图就是简单的说明:



5.1 定时计划 TimingPlan

TimingPlan 是定时器最核心最重要的参数, TimingPlan 表示作业自动执行的

频度周期，周期间隔最小为 1 秒，TimingPlan 不设置，表示该作业不自动执行。自 5.1 版本开始，TimingPlan 属性也新增弹窗设计，和 period 属性类似，如下图所示：



TimingPlan 属性由三段组成，第一段是时间基线，需要选择启用才会生效，默认为“*”，表示以启动定时器的时间作为时间基线；第二段是间隔单位，可选择 s（秒）、m（分）、h（小时）、d（天）；第三段是间隔频率。

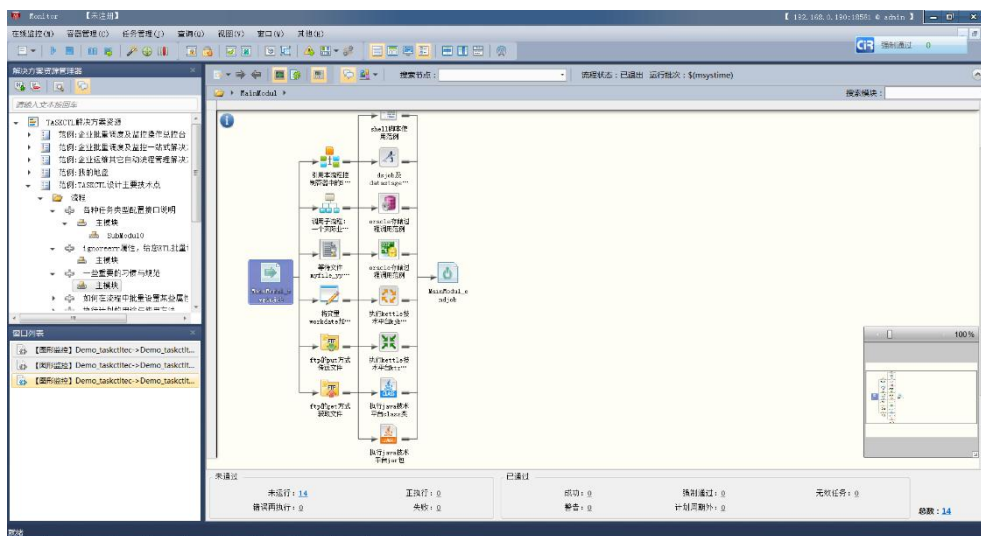
说明：这里要弄清楚第二个和第三个参数合起来表达了完整的频率，但是第一个参数不是起始时间，而是时间基线。调度核心根据当前时间和时间基线的差值是否为间隔频率的整数倍，来决定是否执行。举例 `<timingplan>100101 h 1</timingplan>`，因为第一个参数不是起始时间，只是时间基线，所以 090101 的时候，该作业同样会执行，而不是从每天 10 点 1 分 1 秒开始，每隔一小时运行一次；而 `<timingplan>100101 d 1</timingplan>`，是每天 10 点 1 分 1 秒启动是因为间隔是天，这个地方一定要认真的理解。

5.2 其他

定时器其他部分的设计和语法均于流程类似，此处就不在介绍，更多使用示例可以参照范例中定时器部分。

6 流程运行与监控

至此，您已经有能力开发一个完整的调度流程了。在完成流程的开发后，我们需要在“监控维护管理平台—Monitor”中去运行它。您会看到如下图所示的监控维护界面，我们将在《Monitor 监控管理》手册中作更详细的介绍。



7 快捷键速查表

快捷键	功能描述
Atl+F4	关闭系统
Ctrl+A	全选当前文本控件的文本
Ctrl+C	拷贝选定文本
Ctrl+F	打开“查找和替换”窗口
Ctrl+J	AI 智能创建作业窗口
Ctrl+K	整理当前“模块代码”设计窗口代码
Ctrl+L	打开“新建流程”窗口
Ctrl+M	打开“新建模块”窗口
Ctrl+O	打开“编译输出”窗口
Ctrl+P	在“模块代码”设计模式中从选定文本建新模块
Ctrl+R	打开“查询结果列表”窗口
CTRL+S	保存当前已更改的模块代码
Ctrl+Shift+S	保存所有已更改的模块代码
Ctrl+V	粘贴已拷贝文本
Ctrl+W	打开“代码错误列表”窗口
Ctrl+X	剪切选定文本
Ctrl+Y	重做上次文本操作
Ctrl+Z	撤销上次文本操作
Del	删除选定文本
F1	打开帮助文档
F6	编译流程
F8	打开资源管理器
F9	打开属性窗口
F10	打开节点工具箱
F11	切换到“模块视图”设计模块
F12	切换到“模块代码”设计模块

8 常见问题 QA

Q: 系统为什么不允许对流程进行任何操作?

A: 请先确认该流程是否被签出。只有签出后,才能对该流程及其模块、变量、作业等对象进行开发设计。

Q: 系统一直提示“该流程已被其它用户占用,无法签出”,怎么办?

A: 请确认其它成员是否签出了该流程,需要其它成员签入流程后,您才能签出该流程。还有种情况是系统非法退出后,导致流程未正常签入。您可以进入后 `ctladmin` 程序,用 `listco` 命令查找签出流程后, `clearco` 命令强制签入指定流程。