
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

v6

字符界面客户端系统

成都塔斯克信息技术有限公司

产品网站：www.taskctl.com

1 前言	4
1.1 文档目的.....	4
1.2 读者对象.....	4
2 认识后台客户端系统	5
2.1 软件的介绍.....	5
2.2 关于 HELP 帮助.....	5
3 管理平台应用系统详解	6
3.1 平台总控文件 XML 语法分析.....	6
3.1.1 一份简单的平台配置文件.....	6
3.1.2 XML 关键字.....	7
3.1.3 任务类型.....	9
3.1.4 CIR 节点基础信息.....	10
3.1.5 工程信息.....	11
3.1.6 全局变量信息.....	11
3.1.7 关系节点信息.....	12
3.2 登录系统.....	12
3.3 功能介绍.....	13
3.3.1 平台搭建.....	13
3.3.2 节点启动、停止.....	14
3.3.3 产品注册授权.....	16
3.3.4 流程导入、导出.....	18
3.3.5 流程签出状态管理.....	20
3.3.6 平台信息查询.....	21
3.4 应用实例.....	错误! 未定义书签。
3.4.1 如何快速的增加一个调度服务器并让它启动.....	错误! 未定义书签。
4 开发平台应用系统详解	27
4.1 登录系统.....	27
4.2 功能介绍.....	27

4.2.1 流程环境部署.....	28
4.2.2 流程开发设计.....	29
4.2.3 流程编译.....	41
4.2.4 信息查询.....	42
4.3 DEBUG 调试系统.....	错误! 未定义书签。
4.3.1 认识 debug 系统.....	错误! 未定义书签。
4.3.2 登录系统.....	错误! 未定义书签。
4.3.3 功能介绍.....	错误! 未定义书签。
4.4 应用实例.....	错误! 未定义书签。
4.4.1 如何快速发布一个新的流程并进行调试.....	错误! 未定义书签。
4.4.2 如何快速拆解流程中的多个任务到新增模块中去.....	错误! 未定义书签。
5 监控平台应用系统详解.....	50
5.1 登录系统.....	50
5.2 功能介绍.....	51
5.2.1 流程环境部署.....	51
5.2.2 流程监控维护.....	52
5.2.3 任务监控维护.....	64
5.2.4 信息查询.....	71
6 附录.....	79

1 前言

1.1 文档目的

后台客户端系统主要由 Admin(平台管理)、Flowc(流程配置与管理)、Monitor(流程监控与维护)三套软件构成，它们分别于桌面软件 Admin、Designer、Monitor 相对应。本文旨在介绍后台客户端系统三套软件功能与操作，以便帮助相关人员对该系列软件的使用。

1.2 读者对象

《TASKCTL 字符界面客户端系统》主要适合以下读者对象：

- ✓ 系统维护人员
- ✓ 项目管理人员
- ✓ 技术开发人员
- ✓ 运维监控人员

2 认识后台客户端系统

后台客户端系统又称做后台字符界面系统，该系统主要由 3 个软件构成：分别是 `ctladmin`、`ctlflowc` 以及 `ctlmoni`。它们共同协作完成调度功能。后台客户端系统是运行在 `linux` 环境下的交互程序，同后台核心系统一样是通过 `TCP/IP` 协议进行通信的。后台客户端系统主要为了方便后台运维人员以及相关技术人员使用。

2.1 软件的介绍

✓ `ctladmin`

`ctladmin` 即管理平台，它主要是完成核心系统信息配置、维护管理以及流程的导入导出等功能。

✓ `ctlflowc`

`ctlflowc` 即流程开发配置平台，它主要完成流程基础信息配置、流程开发设计以及任务调试等功能。

✓ `ctlmoni`

`ctlmoni` 即流程监控平台，它主要完成流程运行时的监控以及维护管理等功能。

2.2 关于 help 帮助

三个客户端软件都提供了“`help`”帮助命令，方便用户快速的了解各个命令的正确使用方法。进入系统后直接输入“`help`”即可获取该系统的所有功能菜单信息；使用“`help+命令`”即可获取该命令的具体使用方法。

3 管理平台应用系统详解

3.1 平台总控文件 XML 语法分析

平台总控信息文件对整个平台来说非常重要，它是通过 XML 语言进行描述的。第一次初始化该文件的时候，需要通过桌面客户端软件 Admin 来完成。之后如果需要维护，则可以通过后台字符界面 ctldadmin 客户端软件或者 Admin 桌面软件来协同完成。

3.1.1 一份简单的平台配置文件

```
<taskctl-cir>
<def-jobtype>
  <jobtype>
    <typename>exe</typename>
    <type>1</type>
    <pluginname>cprunexe</pluginname>
    <stopplugin></stopplugin>
    <defaultpara></defaultpara>
    <defaultexppara></defaultexppara>
    <successv>0</successv>
    <failedv></failedv>
    <errorv></errorv>
    <ignoreerv></ignoreerv>
    <relapicname>icon1</relapicname>
    <treepicname>icon1</treepicname>
  </jobtype>
</def-jobtype>
<ctinode>
  <id>2</id>
  <name>svr1</name>
  <desc></desc>
  <ip>172.22.1.210</ip>
  <port>8586</port>
  <runmaxnum>10</runmaxnum>
  <errdelay>2</errdelay>
  <msgcommlog></msgcommlog>
  <msgscancel></msgscancel>
  <logsavedays>1</logsavedays>
  <vjobflag></vjobflag>
  <vjobrttime></vjobrttime>
  <plugmaxnum></plugmaxnum>
  <hissavedays>0</hissavedays>
  <scandelav>4</scandelav>
  <def-app>
    <app>
      <appid>1</appid>
      <name>ecif</name>
      <bz></bz>
    </app>
  </def-app>
  <def-para>
    <para>
      <name>test</name>
      <value>111</value>
      <paratype>date</paratype>
      <encry>N</encry>
    </para>
  </def-para>
</ctinode>
```

任务类型信息

调度服务器基础信息

工程信息

全局变量信息

```

</def-ctlnode>
<def-noderela>
  <em-node>
    <name>emnode</name>
    <srv-node>
      <name>svr1</name>
    </srv-node>
  </em-node>
</def-noderela>
</taskctl-cir>

```



通过以上范例可以看出平台配置信息总共分为三部分：任务类型信息、节点信息以及节点关系信息。而节点信息又由节点基础配置信息、工程信息以及全局变量信息组成。

3.1.2 XML 关键字

分类	关键字	作用
根节点	taskctl-cir	整个平台配置信息文件 XML 根节点
任务类型信息	def-jobtype	整个任务类型描述主节点
	typename	任务类型名称，不可缺省
	description	备注
	type	任务类型编号，不可缺省
	rundrivertype	任务类型驱动定义执行方式
	stopdrivertype	任务类型驱动定义停止方式
	pluginname	执行插件名称，不可缺省
	stopplugin	停止插件名称
	defaultpara	缺省参数
	defaultexppara	缺省环境参数
	defaultvirres	任务消耗资源缺省值
	successv	任务返回值缺省定义中成功返回值
	failedv	任务返回值缺省定义中失败返回值
	errorv	任务返回值缺省定义中错误返回值
	ignoreerv	任务返回值缺省定义中警告返回值
relapicname	关系树图标	
treepicname	节点树图标	
节点基础信息	def-ctlnode	整个节点信息描述主节点
	ctlnode	单个节点信息描述主节点
	id	CIR 节点编号，不可缺省
	name	CIR 节点名称，不可缺省
	desc	CIR 节点描述
	ip	节点 IP 地址，不可缺省
	port	节点端口，不可缺省
	enable	CIR 节点启用状态

	runmaxnum	最大并行度, 缺省默认为 10
	maxvirresval	最大虚拟资源总值
	errdelay	平台错误重调间隔, 单位秒
	msgcommlog	组件通信日志: Y 输出; N 不输出
	msgscancel	消息中心自动取消客户端订阅时间, 单位分, 缺省默认 30
	logsavedays	平台日志保存天数, 单位天
	vjobflag	虚拟运行标志: Y 虚拟运行; N 实际运行
	vjobrttime	虚拟运行时间, 单位秒, 缺省默认 20
	plugmaxnum	插件最大运行数
	hissavedays	历史数据保留天数
	scandelay	进行扫描间隔, 单位秒
工程信息	def-app	在指定调度服务器下的整个工程信息描述主节点
	app	在指定调度服务器下的单个工程信息描述主节点
	appid	工程编号, 不可缺省
	name	工程名称, 不可缺省
	bz	工程描述
全局变量信息	def-para	在指定调度服务器下的整个全局变量信息描述主节点
	para	在指定调度服务器下的单个全局变量信息描述主节点
	name	变量名称, 不可缺省
	value	变量值
	paratype	变量类型: comm-普通变量; date-日期变量; 缺省默认 comm
	encry	是否加密: Y-加密; N-不加密; 缺省默认 N
	desc	描述
节点关系	def-noderela	整个节点关系描述主节点
	em-node	EM 节点, 不可缺省
	svr-node	调度服务器节点, 不可缺省
	magt-node	主代理节点
	sagt-node	从代理节点
	name	各个节点名称

3.1.3 任务类型

任务类型信息主要包括：任务类型名称、编号、插件信息、参数信息、返回值信息以及图形标签信息等。

(一) 任务类型名称

任务类型名称是流程开发任务时使用到的名称，使用时需要注意以下几点：

- ✓ 唯一性：任务类型名称在整个平台是唯一的，不能重复
- ✓ 长度：任务类型名称长度不能超过 10 个字符
- ✓ 输入限制：只能有英文字符与数字组合，不能输入全角字符与汉字

(二) 任务类型编号

任务类型编号是任务类型的关键索引，使用时需要注意以下几点：

- ✓ 唯一性：任务类型编号在整个平台是唯一的，不能重复
- ✓ 输入限制：只能是数字，且大于 0 小于 50

(三) 插件名称

执行插件名称是任务运行时需要靠执行插件来调用的插件程序名称。使用时需要注意几点：

- ✓ 长度：插件名称长度不能超过 20 个字符
- ✓ 输入限制：只能有英文字符与数字组合，不能输入全角字符与汉字

停止插件跟执行插件一致，在任务中断的时候需要靠停止插件来中断调用程序。

(四) 参数信息

在这里就不做介绍,请参考文档《TASKCTL 代码规则语法》相关章节。

(五) 任务返回值

每一个任务类型都有返回值，用于判断任务调用成功与否。可使用如“1-10”这样的连续数字表达式。

(六) 图形标签信息

主要用于桌面软件 Designer 和 Monitor 相关资源的展示性图标，提高其辨识度。具体参考文档《TASKCTL 管理平台 Admin》相关章节。

3.1.4 CIR 节点基础信息

CIR 节点基础信息主要包括：节点编号、节点名称、节点描述、IP 地址和端口号等。

(一) 节点编号

节点编号是该节点信息的关键索引之一，在后台客户端很多命令都需要设置该编号，然后才能进入到该节点中对其进行相关命令操作。使用时注意以下几点：

- ✓ 唯一性：节点编号在整个平台里是唯一的，不能重复
- ✓ 输入限制：只能是数字

(二) 节点名称

节点名称是该节点信息的关键索引之一，使用时需要注意以下几点：

- ✓ 唯一性：节点名称在整个平台是唯一的，不能重复
- ✓ 长度：节点名称长度不能超过 30 个字符
- ✓ 输入限制：只能有英文字符与数字组合，不能输入全角字符与汉字

(三) 节点描述

节点描述是对该节点进行概述性的说明，说明该节点的作用等相关信息。该信息不是必输项，且可以是任务字符组合。

(四) IP 地址和端口号

这两个字段组合在一起构成该节点关键索引之一，主要用于服务节点间的通信。

(五) 最大并行度

最大并行度指的是该节点上的任务最大同时运行的数量。

3.1.5 工程信息

工程信息主要内容包括：工程编号、工程名称、工程描述等相关信息。

(一) 工程编号

工程编号是工程信息的关键索引之一，在使用工程编号时需要注意以下几点：

- ✓ 唯一性：工程编号在整个平台是唯一的，不能重复
- ✓ 输入限制：只能是数字

(二) 工程名称

工程名称是工程信息的索引之一，在使用工程名称时需要注意以下几点：

- ✓ 唯一性：工程名称在整个平台是唯一的，不能重复
- ✓ 长度：工程名称长度不能超过 20 个字符
- ✓ 输入限制：只能有英文字符与数字组合，不能输入全角字符与汉字

(三) 工程描述

工程描述是对工程的说明信息，该信息不是必输项，且可以是任意字符组合。

3.1.6 全局变量信息

全局变量信息主要包括：变量名称、变量值、变量类型、是否加密等信息。

全局变量的应用范围为调度服务器内。

(一) 变量名称

变量名称在使用时需要注意以下几点：

- ✓ 唯一性：变量名称在调度服务器内是唯一的
- ✓ 长度：变量名称长度不能超过 50 个字符
- ✓ 输入限制：只能由英文字符与数字组合，不能输入全角字符或汉字

(二) 变量值

变量值即是该全局变量的值，它类似于开发语言中的“宏替换”。该值的长度不能超过 100 个字符，且可以任意组合。

(三) 变量类型

变量类型有两种：“comm-普通变量”和“date-日期变量”。

(四) 是否加密

在 TASKCTL 中提供了变量加密功能，因为有的变量是不能展示给用户的，诸如用户密码等信息。Y-加密和 N-不加密，缺省系统默认为不加密。

3.1.7 关系节点信息

在 CIR 节点信息里面，系统并没有定义节点类型。在关系节点信息里，我们定义了哪些节点是调度服务节点、哪些是主代理或者从代理节点以及上下级的关系。

一般规则：调度服务节点挂载主代理节点；主代理节点挂载从代理节点。

3.2 登录系统

在登录后台管理客户端 `ctladmin` 之前需要初始化调度服务节点。之后，登录过程中需要用户输入用户名，密码方可进入到该应用系统中，如下图所示：

```
[test_cs_50@localhost conf]$ ctladmin
CTL*Admin: TASKCTL平台综合管理软件 Ver5.0.1 2017年06月13日 17:53:10

版权所有：成都塔斯克信息技术有限公司
授权用户：

登录用户：admin
登录密码：

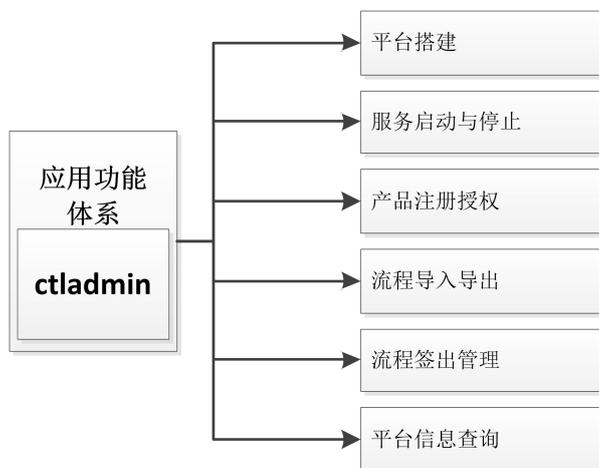
~~~~~
                软件未注册
            当前产品安装ID为：017E-09WW-05PB-00T8

                请登录www.taskctl.com获取授权
            ~~~~~
~~~~~ 官方网站：www.taskctl.com ~~~~~ 【未注册】 ~~~~~
** 温馨提示 **
    'help'命令可以获取所有命令的帮助信息
ADMIN>
```

通过该图，可以得到如下信息：版本号、系统日期时间、使用单位以及该版本注册情况等基本信息。

3.3 功能介绍

ctladmin 的功能如下图所示:



在命令行输入“help”帮助命令，查看“ctladmin”的全部命令介绍。

3.3.1 平台搭建

为了方便用户能够更好、更快的搭建一个平台，后台管理客户端 **ctladmin** 提供了平台配置信息上传和平台配置信息管理维护两个功能。

3.3.1.1 平台配置信息上传

- ◆ 命令: loadcfg
- ◆ 语法: loadcfg
- ◆ 功能概述: 该命令主要是上传平台配置信息。用户只需要把 **ctlconf.xml** 文件拷贝到后台客户端的 $$(TASKDIRCTL)/conf$ 目录下。然后使用此命令，系统会自动把文件上传到核心平台并加载到核心系统中。

具体使用如下:

```
ADMIN> loadcfg
是否确定上传文件'/home/taskctl/src/taskctl/conf/ctlconf.xml' (y/n) y
上传成功
ADMIN>
```

由上图所示，若上传成功，则可以使用相关查询命令查询平台节点信息、工程信息、全局变量信息、任务类型信息等。

3.3.1.2 平台配置信息管理维护

- ◆ 命令: `opencfg`
- ◆ 语法: `opencfg`
- ◆ 功能概述: 该命令功能为在线打开平台配置信息文件, 方便管理者对平台配置信息进行维护和管理。

具体使用如下:

```
ADMIN>  
ADMIN> opencfg
```

由上图所示: 该命令执行后, 系统进入到文本编辑窗口, 此时可以对平台配置信息进行编辑。管理者具体可以进行如下操作:

- ✚ 对任务类型进行增、删、改;
- ✚ 对全局变量信息进行增、删、改;
- ✚ 对工程信息进行增、删、改;
- ✚ 对 CIR 节点信息进行增、删、改;
- ✚ 对 CIR 节点关系进行增、删、改;

对以上内容进行修改后, 需要重新上传总控文件。在上传时确保调度平台处于未启动状态。之后, 可以使用 `listcirm`, `listjt`, `listapp`, `listpara` 等命令查看相关信息。

具体管理维护请参见第 3 章里的[平台总控文件 XML 语法分析](#)

3.3.2 节点启动、停止

若对调度流程进行开发和运行监控, 则必须要启动服务器。系统为管理员提供了对服务器的启动和停止功能。

3.3.2.1 CTL 节点启动

- ◆ 命令: `ctlstart`
- ◆ 语法: `ctlstart [CTL 节点号]`
- ◆ 功能概述: 该命令功能为实例启动一个 CTL 节点。TASKCTL 产品其它

相关命令的使用都是建立在 CTL 启动的前提条件下。其主要完成以下事情:

- ✚ 建立各种 IPC 资源。比如: 共享内存、消息队列、以及初始化系统数据。
- ✚ 启动 CTL 节点的相关服务组件。

具体使用如下:

```
ADMIN> ctlstart 2
开始启动调度服务节点'svr1' ... ..
调度服务节点'svr1'启动成功
ADMIN>
```

如上图所示, 如果成功, 可以使用命令 `cirinfo` 命令查看此 CTL 节点启动成功。作为管理者, 还必须在用户环境下查看以下信息:

- ✚ `ipcs` 命令: 查看系统是否分配了共享内存出来。
- ✚ 如果是调度平台, 需要查看 `ctlcpg,ctlsps,ctldrr,ctldar,ctlstr,ctldsy,ctlemr,ctlucd,ctljmm` 等进程是否成功启动。
- ✚ 如果是主代理或者从代理平台, 需要查看 `ctlcpg,ctlsps,ctldrr,ctldar,ctlstr` 等进程是否成功启动。

在实际应用中, 该平台一旦启动, 如果没有特殊情况, 平台不需要停止重启。具体使用方法请使用“`help+ctlstart`”命令查看。

3.3.2.2 CTL 节点停止

- ◆ 命令: `ctlstop`
- ◆ 语法: `ctlstop [CTL 节点编号]`
- ◆ 功能概述: 该命令功能为实例停止一个 CTL 服务平台。其主要完成以下事情:
 - ✚ 拒绝接收远程执行命令
 - ✚ 停止本地流程, 同时等待所有现有正在运行的任务运行完闭;
 - ✚ 共享内存信息同步到相应文件中
 - ✚ 停止 CIR 节点的相关服务组件; 释放 IPC 各种资源, 比如: 共享内存、消息队列等,

最后停止该 CTL 平台

具体使用如下:

```
ADMIN> ctlstop 2
开始停止调度服务节点'svr1' ... ..
调度服务节点'svr1'成功停止
ADMIN>
```

另外,可以使用命令 `listcirm` 查看该 CTL 平台的运行状态。作为管理者,还应在用户环境下查看以下信息:

- ✚ `ipcs` 命令: 查看系统是否释放了共享内存。
- ✚ 若在调度平台下,查看 `tlcpg,ctlsps,ctldrr,ctldar,ctlstr,ctldsy,ctlemr,ctlucd,ctljmm` 等进程是否成功退出。
- ✚ 若在主代理或者从代理平台下,查看 `ctlcpg,ctlsps,ctldrr,ctldar,ctlstr` 等进程是否成功退出。

在实际应用中,该平台一旦启动,如果没有特殊情况,平台不需要停止重启。具体使用方法请使用命令“`help+ctlstop`”查看。

3.3.3 产品注册授权

3.3.3.1 产品 key 文件生成器

- ◆ 命令: `genlickey`
- ◆ 语法: `genlickey`
- ◆ 功能概述: 该命令功能为产生一个 `key` 文件,到授权中心获取 `licence` 信息。<该命令从 5.0 开始失效,5.0 以后改用安装 ID>

具体使用如下:

```
ADMIN> genlickey
平台ctlkey文件产生成功,文件名为:/home/taskctl/ctlkey.txt
ADMIN>
```

如上图所示:生成的 `key` 文件系统默认放在 `HOME` 目录下。此时用户可以把该文件发送到授权中心生成 `licence` 文件。

3.3.3.2 产品注册

- ◆ 命 令: `regist`
- ◆ 语 法: `regist`
- ◆ 功能概述: 该命令主要是把通过授权中心获取 `taskctl.lic` 信息, 然后注册到本系统中, 用来获取更高的使用权限。

具体使用如下:

```
ADMIN> regist
输入licence文件名称, 缺省为当前目录taskctl.lic文件:taskctl.lic
注册成功
ADMIN>
```

用户可以使用 `licinfo` 查看该产品的授权信息。

3.3.3.3 产品注册信息查询

- ◆ 命 令: `licinfo`
- ◆ 语 法: `licinfo`
- ◆ 功能概述: 该命令主要是显示该调度产品注册后的版本类型, 使用期限等信息。

具体使用如下:

平台licence如下:

注册方式: LICKEY
安装ID: 01AT-09YV-05LN-010C
是否绑定: 是
EM平台: 无
主备机制: 无
注册状态: 已注册
到期日期: 20180601
授权作业数: 54321
授权代理数: 99
当前作业数: 15474
当前代理数: 33

3.3.4 流程导入、导出

为了快速的实现流程数据备份和移植功能, 系统特为管理员提供了流程的导出(备份)和流程导入(数据移植)功能。

3.3.4.1 流程导出

- ◆ **命令:** export
- ◆ **语法:** export
- ◆ **功能概述:** 为了方便流程数据的快速移植, 系统提供了流程导入导出功能。该导出功能可以导出单个, 或者多个以及整个平台的所有流程信息。
- ◆ **命令前置条件:** 导出调度服务器必须为启动状态。

具体使用如下:

- ✓ 导出前的准备步骤, 如下图所示:

```
ADMIN> export
```

```
输入导出文件名称:abc.txt
导出应用工程编号 :
导出流程或定时器对象编号列表 :
```

```
是否导出2号服务器下所有对象(y/n)y
```

由上图所示：使用 `export` 命令后，如上所示选择 `y` 表示导出 2 号服务器下的所有流程。

- ✓ 导出中，如下图所示：

```
正在导出2号服务器下所有对象，请稍后... ..
```

- ✓ 导出成功后，如下图所示：

```
是否继续(y/n)n
```

```
成功导出到文件'abc.txt'，文件大小555872字节
ADMIN>
```

由上图所示，选择 `n`，将不继续导出，系统返回导出文件名及文件大小。如果导出文件没有路径，系统则默认导出到当前目录下。

3.3.4.2 流程导入

- ◆ **命令：** `import`

- ◆ **语法：** `import`

- ◆ **功能概述：** 为了方便流程数据的快速移植，系统提供了导入导出流程功能。该导入功能可以导入文件流程信息，同时支持编译功能。在导入前，必须确保系统的调度服务器名和工程名必须和导入流程文件里的信息一致，否则无法导入。

- ◆ **命令前置条件：** 导入调度服务器必须为启动状态。

具体使用如下：

- ✓ 导入前：

```
ADMIN> import
输入导入文件名称:abc.txt
```

由上图所示，用户需要录入导入文件名称。

✓ 导入中：

系统开始导入前准备...

```
[1:45]开始导入Demo_OP_Time_MainOper流程...  
'Demo_OP_Time_MainOper'导入成功
```

```
[2:45]开始导入Demo_OP_Timer_ECIFView流程...  
'Demo_OP_Timer_ECIFView'导入成功
```

✓ 导入后：

是否编译成功导入对象(y/n)y

由上图所示，如果选择 y 则快速编译导入成功的流程信息，如下图所示：

```
[1:45]开始编译导入对象：Demo_OP_Time_MainOper  
'Demo_OP_Time_MainOper'编译成功，共23个节点
```

```
[2:45]开始编译导入对象：Demo_OP_Timer_ECIFView  
'Demo_OP_Timer_ECIFView'编译成功，共9个节点
```

说明： 流程在导入过程中，若某一流程导入失败，系统会继续导入下一流程。
另外，只编译成功导入的流程。

3.3.5 流程签出状态管理

为了方便开发者因为其它不确定原因导致流程没有正常签入，而其它用户要修改此流程的情况，同时方便管理者对流程签出状态的任务进行有效管理。

3.3.5.1 流程签出状态查询

◆ 命令：listco

- ◆ **语 法:** listco
- ◆ **功能概述:** 该命令主要是列举出指定调度服务器内签出状态的流程。它主要是方便管理者对流程开发进行统一管理。为清除流程签出状态提供支持。
- ◆ **命令前置条件:** 查询的调度服务器必须为启动状态。

具体使用如下:

```
ADMIN> listco
```

编号	控制容器名称	容器类型	签出时间
29	Demo_taskctltec_Flow1	流程	20170614101619
39	Demo_taskctltec_Flow11	流程	20170614101623

共 2 条记录

3.3.5.2 清除流程签出状态

- ◆ **命 令:** clearco
- ◆ **语 法:** clearco [流程号]
- ◆ **功能概述:** 强制清除指定调度服务器上指定流程编号的签出状态。
- ◆ **命令前置条件:** 流程所属调度服务器必须为启动状态。

具体使用如下:

```
ADMIN> clearco 29
```

流程29清除签出成功

如上图所示: 命令使用成功后再次使用 listco 查询该流程签出状态。另外, 系统每隔 30 分钟会对不活动的流程进行强制签入操作。

3.3.6 平台信息查询

为了方便管理者对平台信息有更详细的了解, 系统提供了多种平台信息查询命令。如对工程信息查询、对节点信息查询等。借此, 管理者对平台可进行更好

的管理和维护。

3.3.6.1 平台调度服务器查询

- ◆ 命令: listsvr
- ◆ 语法: listsvr
- ◆ 功能概述: 该命令功能为列举出整个平台的调度服务器信息列表。其展示了调度服务器编号、名称、IP 地址和端口号、节点状态以及下级节点列表等相关信息。通过此命令可以了解到整个平台调度服务器的分布情况。具体使用如下:

```
ADMIN> listsvr
```

编号	服务名称	IP地址	端口	节点状态	下级节点列表
2	svrnode	192.168.0.236	18501	5-启动状态	4,3,5,6,7,9,10,11,12,13,14,30,

共 1 条记录

3.3.6.2 平台工程信息查询

- ◆ 命令: listapp
- ◆ 语法: listapp
- ◆ 功能概述: 该命令功能为查询整个平台的工程信息。不指定参数, 则默认列举出平台所有工程信息。其展示了应用编号、应用名称以及相关描述信息。通过此命令也可以了解指定应用编号的工程信息情况。

具体使用如下:

```
ADMIN> listapp
```

应用编号	应用名称	描述
1	Demo_aaOperMP	范例:企业批量调度及监控操作总控台
2	Demo_aentbatdispat	范例:企业批量调度及监控一站式解决方案
3	Demo_entautoflowpr	范例:企业运维其它自动流程管理解决方案
4	Demo_myworkspace	范例:我的地盘
5	Demo_taskctltec	范例:TASKCTL设计主要技术点
6	project1	应用工程1

共 6 条记录

3.3.6.3 平台全局变量信息查询

- ◆ 命 令: listvar
- ◆ 语 法: listvar
- ◆ 功能概述: 该命令功能为查询整个平台的全局变量信息。其展示了变量所属调度服务器、变量类型、名称以及变量值等相关信息。

具体使用如下:

```
ADMIN> listvar
```

服务器名称	类型	变量名称	变量值	说明
svrnode	普通变量	test	value	描述
svrnode	普通变量	home	/home/taskctl	根路径

共 2 条记录

由上图所示, 可以查询出查询出整个平台的全局变量信息。

3.3.6.4 平台任务类型查询

- ◆ 命 令: listjt
- ◆ 语 法: listjt
- ◆ 功能概述: 该命令功能为查询整个平台的任务类型信息列表。其展示了任务类型的编号、名称、执行插件以及停止插件等相关信息。

具体使用如下:

```
ADMIN> listjt
```

编号	类型名称	类型描述
1	exe	可执行程序任务类型
2	sh	shell文件类型
3	script	shell脚本类型
4	oraproc	oracle存储过程任务类型
....
60	sendevent	自定义事件发送节点组件
61	recvevent	自定义事件接收节点组件
62	ftpget	ftp接收文件节点组件
63	ftpput	ftp发送文件节点组件
70	monitcc	控制容器监控组件

共 25 条记录

由上图所示, 可以查询出整个平台的任务类型信息, 如果需要查看单个任务类型的详细信息, 请使用 `jtinfo` 命令。

3.3.6.5 任务类型详细信息

- ◆ 命 令: `jtinfo`
- ◆ 语 法: `jtinfo [任务类型名称]`
- ◆ 功能概述: 该命令功能为展示指定任务类型的详细信息。主要有任务类型的插件信息、参数信息以及返回值信息等。

具体使用如下:

```
ADMIN> jtinfo sh
```

作业类型 'sh' 的插件配置信息如下:

类型编号	: 2
类型名称	: sh
类型描述	: shell文件类型
执行驱动方式	: command
执行驱动插件	: sh
停止驱动方式	: command
停止驱动程序	:
缺省参数	:
缺省扩展参数	:
成功返回值	: 0
错误重做返回值	: 1-98
失败返回值	: 100
错误警告返回值	: 99
插件描述	: shell文件类型
树图标	: icon5
作业图标	: icon5

3.3.6.6 平台节点信息查询

- ◆ 命令: listcirm
- ◆ 语法: listcirm
- ◆ 功能概述: 该命令功能为查询整个平台所有 CIR 节点相关信息列表, 其展示了节点的编号、类型、节点状态、运行数以及上级节点。通过该命令可以了解整个平台的 CIR 节点分布情况以及运行情况等相关信息。

具体使用如下:

```
ADMIN> listcirm
```

编号	节点名称	节点类型	节点状态	运行数	上级节点	握手状态
2	svrnode	2-控制服务节点	2-未启动	0	1	2-主机握手
3	abc	3-主代理节点	3-异常错误	0	2	0-未握手

共 2 条记录

由上图所示, 可以查询出整个平台的 CIR 节点信息。如果要查询每个 CIR 节点详细信息, 请使用 **cirninfo** 命令。

3.3.6.7 平台节点详细信息

- ◆ 命令: cirninfo
- ◆ 语法: cirninfo [CIR 节点号]
- ◆ 功能概述: 该命令功能为展示指定 CIR 节点的详细信息。通过此命令了解到节点的类型、IP 地址和端口号、运行状态、是否打印日志、最大并行度、是否虚拟任务以及其它相关属性信息。

具体使用如下:

```
ADMIN> cirninfo 2
```

```
控制节点'svrnode'信息:
```

```
-----  
id           : 2  
svrid        : 2  
type         : 2-控制服务节点  
name         : svrnode  
nodemem      : 调度服务节点  
ip           : 192.168.0.190  
port         : 18581  
runmaxnum    : 100  
currunnum    : 0  
maxvirresval : 1000  
curvirresval : 0  
errdelay     : 60  
msgcommlog   : N  
msgscancel   : 30  
logsavedays  : 10  
vjobflag     : N  
vjobrtime    : 20  
lastnode     : 1  
nextnode     : 3,  
cup_used     : 48.06  
disk_used    : 36.00  
state        : 2-未启动  
-----
```

4 开发平台应用系统详解

4.1 登录系统

服务启动后才能登录后台流程开发软件 `ctlflowc`，在登录过程中需要用户输入用户名，密码方可进入到该应用系统中，如下图所示：

```
[test_cs_50x@localhost ~]$ ctlflowc

CTL*Flowc: TASKCTL流程及定时器设计管理软件 Ver5.0.1 2017年06月14日 15:05:06

版权所有：成都塔斯克信息技术有限公司
授权用户：

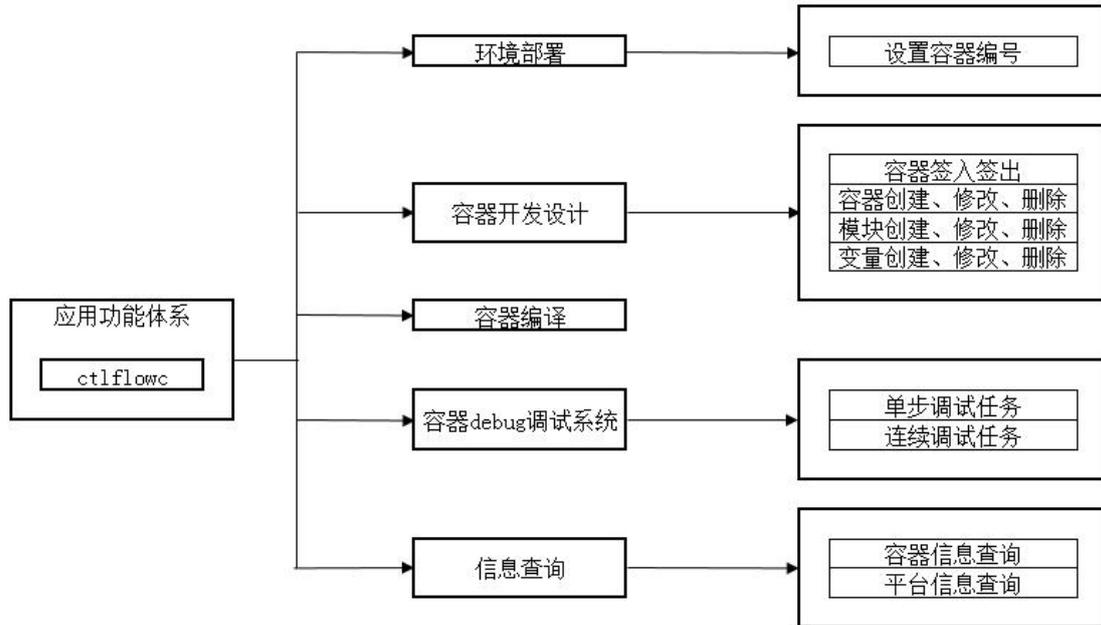
登录用户：admin
登录密码：

~~~~~ 官方网站：www.taskctl.com ~~~~~ 【已注册】 ~~~~~
** 温馨提示 **
    'help'命令可以获取所有命令的帮助信息
FLOWC_[0001]>
```

通过上图可知：版本号、系统日期时间、使用单位以及该版本注册情况等基本信息。

4.2 功能介绍

`ctlflowc` 的功能体系如下图所示：



用户也可以输入“help”帮助命令，查看后台流程开发软件 `ctlflowc` 的全部命令介绍。

4.2.1 环境部署

4.2.1.1 设置容器编号

- ◆ 命令：`tccid`
- ◆ 语法：`tccid [流程号]`
- ◆ 功能概述：该命令功能为设置容器为当前开发环境。之后才能在指定容器下对该容器进行如容器维护、模块维护、变量维护等相关操作。
- ◆ 命令前置条件：必须启动调度服务器。

具体使用如下：

```
FLOWC_[0002]> tccid 46
```

```
FLOWC_[0046]>
```

4.2.2 容器开发设计

4.2.2.1 容器创建、修改、删除

4.2.2.1.1 容器创建

- ◆ 命令: addtcc
- ◆ 语法: addtcc
- ◆ 功能概述: 整个技术平台容器分流程[flow]容器与定时器[timer]容器。下面主要的内容都是基于流程容器讲解。流程是将多个任务按一定关系组合,并具有相应的运行条件的集合。它是调度平台客户化的主要内容,调度核心通过流程信息完成各种调度工作。流程信息主要包含两方面的信息,即任务基本信息与调度控制策略信息。任务基本信息描述流程调度的任务是什么,在哪里调度等;控制策略信息主要描述怎么去控制任务的执行,任务在什么条件下才可以执行。所以流程的创建,主要是构建这两类信息。
- ◆ 命令前置条件: 必须启动调度服务器。

具体操作步骤如下:

- ✓ 使用命令 addtcc 新增容器, 然后输入相关要素,如下图所示:

```
FLOWC_[1-001]> addtcc
0. 流程; 1. 定时器:0
应用编号(1-200): 6
    流程名称:
    流程描述:
子流程(N-否;Y-是):
调度批次生成规则:
    日志保留天数:
    最大并行度:
```

```
是否保存(y/n)y
```

由上图可知:

- 🚦 容器类型: 容器的类型是流程或者定时器。如果不录入, 系统则默认为流程控制容器;

- ✚ 应用编号：即为该调度服务器下挂载的应用工程编号；容器也必须挂在某个应用工程下，必填项。
- ✚ 流程[定时器]名称：在调度服务器下必须唯一。如果不录入，系统则自动生成；
- ✚ 流程[定时器]描述：如果不录入，则默认为空。
- ✚ 子流程：如果不设置，系统则默认为主流程。只有子流程才能被调用，具体应用说明详见《TASKCTL 流程与模块代码》。
- ✚ 调度批次生成规则：系统默认为\$(msystime)。
- ✚ 日志保留天数：系统默认为 5。
- ✚ 流程最大并行度：系统默认为 5。

以上录入完成后提交系统，用户会得到如下信息：

增加成功。流程名称:project1_Flow1 流程编号:46

由上图可知，系统将会为新生成的流程自动分配流程编号。另外，用户可以使用命令 `tccinfo` 查询流程[容器]的详细信息，来确认之前新增的信息是否如预期。如下图所示：

```
FLOWC_[0046*]> tccinfo
流程基本信息如下:
-----
应用工程名称      : project1
流程编号          : 46
流程名称          : project1_Flow1
流程描述          :
是否子流程        : 否
调度批次规则      : $(msystime)
日志保留天数      : 5
启动模块          : MainModul
最大并行度        : 5
发布情况          : 1-已发布
EM同步情况        : 0-未同步
主备同步情况      : 0-未同步
创建者            : admin
所属用户组        : admin,
拥有用户          : admin,
创建时间          : 20170615111003
流程权限          : rwxrwxrwx
是否签出          : 1-签出
签出标识          : 2017061511461800087960
签出时间          : 20170615115301
更新时间          : 20170615115301
-----
```

此时，新生成的流程是默认是签出状态，可以直接对该流程进行维护管理。

4.2.2.1.2 容器基本属性修改

- ◆ 命令: moditcc
- ◆ 语法: moditcc
- ◆ 功能概述: 该命令功能为修改指定容器的基本信息。
- ◆ 命令前置条件:
 - ✚ 必须启动调度服务器。
 - ✚ 必须设置容器编号。
 - ✚ 容器必须为签出状态。

具体使用如下:

```
FLOWC_[0046*]> moditcc
    流程名称:
    流程描述: 测试修改流程描述
    子流程(N-否;Y-是):
    调度批次生成规则:
    日志保留天数:
    启动模块名称:
    最大并行度:

    是否保存(y/n)y

    基本属性修改成功
FLOWC_[0046*]>
```

由上图可知, 修改的要素包括: 流程名称、流程描述、是否子流程、调度批次生成规则、日志保留天数、启动模块、最大并行度等。录入需要修改的项后(若不录入则不修改该项), 保存提交即可。之后, 可以使用命令 `tccinfo` 查看相关项是否修改成功。

4.2.2.1.3 删除一个容器

- ◆ 命令: deltcc
- ◆ 语法: deltcc
- ◆ 功能概述: 对流程进行删除操作。将彻底删除开发区和工作区的流程信息(如模块、任务以及流程私有变量), 请谨慎执行流程删除命令。

◆ **命令前置条件:**

- ◆ 必须启动调度服务器。
- ◆ 必须设置容器编号。
- ◆ 确保容器为签出状态。
- ◆ 确保容器为停止状态。

具体使用如下:

```
FLOWC_[0046*]> deltcc  
  
是否删除编号为'46'的流程(y/n)y  
  
删除成功  
FLOWC_[-001]>
```

如上图所示: 删除成功后, 可以使用命令 `listtcc` 查看该流程是否存在, 同时也应在后台管理客户端 `ctlmoni` 中使用 `listtcc` 查看该流程是否存在。如果都不存在, 则删除成功, 否则删除失败。

4.2.2.2 容器签入签出

4.2.2.2.1 容器签入

- ◆ **命令:** `checkin`
- ◆ **语法:** `checkin`
- ◆ **功能概述:** 当用户不再对容器进行操作时, 请及时签入, 方便其它用户对该容器进行操作。
- ◆ **命令前置条件:**
 - ◆ 必须启动调度服务器。
 - ◆ 必须设置容器编号。

具体使用如下:

```
FLOWC_[0046*]> checkin  
46号流程 'project1_Flow1' 签入成功  
FLOWC_[0046]>
```

此时在容器号后的“*”号消失，表示流程为签入状态。可使用命令 `tccinfo` 查看容器签入情况。

4.2.2.2.2 容器签出

- ◆ 命令: `checkout`
- ◆ 语法: `checkout`
- ◆ 功能概述: 该命令功能为签出当前容器。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容器，引起流程开发混乱，系统特此提供了签入、签出功能。容器只有签出了，才能对该容器做一些相关操作。如果开发者在短时间内不再操作容器，请及时签入，这样能够方便其它用户签出容进行维护。另外，系统提供了对 30 分钟后非活动容器进行自动签入的机制。
- ◆ 命令前置条件:
 - ✚ 必须启动调度服务器。
 - ✚ 必须设置容器编号。

具体使用如下:

```
FLOWC_[0046]> checkout  
签出成功  
FLOWC_[0046*]>
```

由上图得知: 在没有签出前, 容器号后没有“*”符号, 在签出后多出一个“*”符号, 表示容器为签出状态。可使用命令 `tccinfo` 查看容器签出情况。

4.2.2.3 容器模块增加、修改、删除

4.2.2.3.1 容器模块增加

- ◆ 命令: addmodul
- ◆ 语法: addmodul [新增模块名称]
- ◆ 功能概述: 该命令功能为在当前容器下新增一个模块。若不指定模块名称, 系统将会自动默认。
- ◆ 命令前置条件:
 - ✚ 必须启动调度服务器。
 - ✚ 必须设置容器编号。
 - ✚ 确保容器为签出状态。

具体使用如下:

```
FLOWC_[0046*]> addmodul modul_test  
  
增加成功, 新模块名称为: modul_test  
FLOWC_[0046*]>
```

由上图可知: 系统将会以模块名称为关键索引, 所以请确保自定义的模块名称在其容器下不重复。另外, 可以使用命令 `listmodul` 查看模块是否增加成功, 如下图所示:

```
FLOWC_[0046*]> listmodul
```

模块名称	模块描述
MainModul	主模块
SubModul0	
modul_test	

共 3 条记录

4.2.2.3.2 容器模块属性修改

- ◆ 命令: modimodul
- ◆ 语法: modimodul [模块名称]

- ◆ **功能概述：**该命令功能为修改当前容器指定模块的基本属性。
- ◆ **命令前置条件：**
 - ✚ 必须启动调度服务器。
 - ✚ 必须设置容器编号。
 - ✚ 确保容器为签出状态。
 - ✚ 确保模块在容器中存在（模块名称大小写敏感）。

具体使用如下：

```
FLOWC_[0046*]> modimodul modul_test
  新名称:
  模块描述: test modimodul

修改成功
FLOWC_[0046*]>
```

由上图可以得知可以对模块名称和模块描述进行修改。完成修改后，可以使用 `listmodul` 命令查看是否修改成功，如下图所示：

```
FLOWC_[0046*]> listmodul
```

模块名称	模块描述
MainModul	主模块
SubModul0	
modul_test	test modimodul

共 3 条记录

```
FLOWC_[0046*]>
```

4.2.2.3.3 容器模块代码修改

- ◆ **命 令：** `openmc`
- ◆ **语 法：** `openmc [模块名称]`
- ◆ **功能概述：**容器的核心，所有任务的运行，逻辑控制等等都在此模块代码信息中。该命令功能为打开当前流程指定模块代码信息文件，以便进一步编辑。如果不指定模块名称，系统默认打开该流程的主模块文件。
- ◆ **命令前置条件：**
 - ✚ 必须启动调度服务器。

 必须设置容器编号。

 容器必须签出状态。

具体使用如下：

```
FLOWC_[0046*]> openmc modul_test
```

由上图使用命令后，系统展示该模块代码信息给用户。模块代码编辑应用具体参照文档《TASKCTL 流程与模块代码》。

4.2.2.3.4 容器模块删除

◆ 命 令：delmodul

◆ 语 法：delmodul [模块名称]

◆ 功能概述：该命令功能为删除当前容器指定模块代码信息文件。

◆ 命令前置条件：

 必须启动调度服务器。

 必须设置容器编号。

 确保容器为签出状态。

具体使用如下：

```
FLOWC_[0046*]> delmodul modul_test
```

```
是否确认将模块'modul_test'删除(y/n)?y
```

```
删除成功
```

```
FLOWC_[0046*]>
```

之后可使用命令 listmodul 查看容器下的模块列表，确认该模块是否删除成。

4.2.2.4 加载容器模块代码信息

◆ 命 令：loadmc

- ◆ **语 法:** loadmc [模块名称] [XML 文件]
- ◆ **功能概述:** 该命令功能为加载容器控制模块代码文件信息, 即把模块代码文件内容加载到模块中去。该功能实现了模块代码文件在任何场所开发, 无需在指定服务器上开发, 方便开发者对模块代码的快速移植。
- ◆ **命令前置条件:**
 - ✚ 必须启动调度服务器。
 - ✚ 必须设置容器编号。
 - ✚ 确保容器为签出状态。

具体使用如下:

把需要加载的文件放在\$HOME/ SubModul0.mxm 文件里, 操作如下图所示:

```
[test_cs_50@localhost ~]$ cat SubModul0.mxm
<!--*****
TASKCTL技术平台模块文件
  工程名称:
  流程名称:
  模块名称: SubModul0

*****-->
<serial>
  <name>SubModul0_rootnode</name>
  <begin>
    <name>SubModul0_beginjob</name>
  </begin>
  <!-- 用户模块代码自定义区开始 -->
  <sh>
    <name>SubModul0_JobNode0</name>
    <programe>/tmp/taskctl/abc.sh</programe>
  </sh>
  <!-- 用户模块代码自定义区结束 -->
  <end>
  <name>SubModul0_endjob</name>
</end>
</serial>
```

需要加载的模块名 SubModul0, 此时使用命令如下图所示:

```
FLOWC_[0046*]> loadmc SubModul0 /home/test_cs_50/SubModul0.mxm
是否确认将流程'project1_Flow1'的XML模块文件'/home/test_cs_50/SubModul0.mxm'上传(y/n)?y
上传模块文件成功
FLOWC_[0046*]>
```

然后使用 openmc 打开 SubModul0 模块代码, 如下图所示:

```

FLOWC_[0046]> openmc SubModul0
<!--*****
TASKCTL技术平台模块文件
  工程名称:
  流程名称:
  模块名称: SubModul0

*****-->
<serial>
  <name>SubModul0_rootnode</name>
  <begin>
    <name>SubModul0_beginjob</name>
  </begin>
  <!-- 用户模块代码自定义区开始 -->
  <sh>
    <name>SubModul0_JobNode0</name>
    <programe>/tmp/taskctl/abc.sh</programe>
  </sh>
  <!-- 用户模块代码自定义区结束 -->
  <end>
    <name>SubModul0_endjob</name>
  </end>
</serial>

```

通过以上所示，完成模块文件加载操作。

4.2.2.5 私有变量增加、修改、删除

4.2.2.5.1 私有变量增加

- ◆ 命 令: addvar
- ◆ 语 法: addvar
- ◆ 功能概述: 该命令功能为在开发区容器控制文件中，增加指定容器的私有变量信息。TASKCTL 平台支持容器控制文件高度参数化，开发配置时，针对某些信息可以用参数表示，在任务实例化时对这些参数进行宏替换。
- ◆ 命令前置条件:
 - ✚ 必须启动调度服务器。
 - ✚ 必须设置容器编号。
 - ✚ 确保容器为签出状态。

具体使用如下:

```
FLOWC_[0046*]> addvar
变量名称: dbuser
变量值: dbuser
变量类型(0-普通变量;1-逻辑日期; 2-常量): 0
是否加密(N-否;Y-是): N
说明: 数据库用户名

是否保存(y/n)y
```

增加成功

通过上图可知: 变量信息具有变量名称、变量值、变量类型、是否加密、说明这五个要素。变量名称必须录入, 其它可以选择录入, 变量类型缺省为“普通”, 是否加密缺省为“不加密”。

此时可以通过命令 `listvar` 查询是否有该变量信息存在, 如下图所示:

```
FLOWC_[0046*]> listvar
```

公私	变量名称	类型	变量值	说明
私有	dbuser	普通变量	dbuser	数据库用户名
公有	test	普通变量	value	描述
公有	home	普通变量	/home/taskctl	根路径

共 3 条记录

通过上图可以知道 `dbuser` 变量已经添加成功。

4.2.2.5.2 私有变量属性修改

- ◆ 命令: `modivar`
- ◆ 语法: `modivar [变量名称]`
- ◆ 功能概述: 该命令功能为修改当前容器指定私有变量的相关属性。
- ◆ 命令前置条件:
 - ✚ 必须启动调度服务器。
 - ✚ 必须设置容器编号。
 - ✚ 确保容器为签出状态。
 - ✚ 确保变量名称存在。

具体使用如下:

```
FLOWC_[0046*]> modivar dbuser
变量值: abc
变量类型(0-普通变量;1-逻辑日期;2-常量):
是否加密(N-否;Y-是):
说明:

是否保存(y/n)y

修改成功
```

通过上图可知, 可以修改变量的变量值、变量类型、是否加密、说明这四个要素, 如不录入则表示不修改。另外, 可以使用 `listvar` 查询变量属性是否修改成功。

说明: 只能修改私有变量属性, 不能修改公有变量和系统日期变量 `sysdate`。

4.2.2.5.3 私有变量删除

- ◆ **命令:** `delvar`
- ◆ **语法:** `delvar [变量名称]`
- ◆ **功能概述:** 该命令功能为删除当前容器的指定私有变量。
- ◆ **命令前置条件:**
 - ◆ 必须启动调度服务器。
 - ◆ 必须设置容器编号。
 - ◆ 确保容器为签出状态。
 - ◆ 确保变量名称存在。

具体使用如下:

```
FLOWC_[0046*]> delvar dbuser

是否删除46号流程的变量'dbuser'(y/n)y

删除成功
FLOWC_[0046*]>
```

之后, 可以通过 `listvar` 查看是否还存在该变量信息。如果没有, 则说明变量已经成功删除。

说明: 只能删除私有变量属性, 不能删除公有变量和系统日期变量。

4.2.3 容器编译

- ◆ 命令: compile
- ◆ 语法: compile
- ◆ 功能概述: 编译当前容器。容器只有通过编译, 把信息发布到发布区域, 用户才能在监控平台中去实现流程的监控维护操作。编译过程中会对当前流程做一些语法检查。具体请参见《TASKCTL 流程开发与设计》相关文档。
- ◆ 命令前置条件:
 - ✚ 必须启动调度服务器。
 - ✚ 必须设置容器编号。

具体使用如下:

```
FLOWC_[0046*]> compile  
  
编译成功  
FLOWC_[0046*]>
```

如上图所示, 编译成功之后, 可以使用 `listtask` 命令查询该流程所有任务节点列表, 如下图所示:

```
FLOWC_[0046*]> listtask
```

ID号	模块名称	作业名称	作业状态
2	MainModul	MainModul_beginjob	1-未执行
3	MainModul	MainModul_JobNode0	1-未执行
4	MainModul	MainModul_endjob	1-未执行

共 3 个作业

同样可以使用 `listnode` 命令查询该流程所有节点信息, 如下图所示:

```
FLOWC_[0046*]> listnode
```

ID号	类型	模块名称	节点名称	节点描述
1	串行	MainModul	MainModul_rootnode	
2	作业	MainModul	MainModul_beginjob	
3	作业	MainModul	MainModul_JobNode0	
4	作业	MainModul	MainModul_endjob	

共 4 个节点

由上面两个示图可知：生成的任务数和节点数一致（root 节点是虚拟节点），所以编译成功。之后可使用 **taskinfo** 对任务的详细信息进行查看，确认与编辑内容是否一致。

4.2.4 信息查询

4.2.4.1 信息查询

4.2.4.1.1 查询调度服务器下所有容器信息

- ◆ **命令：** listtcc
- ◆ **语法：** listtcc [应用工程号][容器号]
- ◆ **功能概述：** 该命令功能为查询出调度服务器下的容器信息。若不指定参数，系统将不执行应用工程号或流程号的过滤条件。
- ◆ **命令前置条件：** 必须启动调度服务器。

具体使用如下：

```
FLOWC_[0046*]> listtcc
```

编号	容器名称	类型	并行度	应用号	应用名称
1	Demo_OP_Time_MainOper	定时器	5	1	Demo_aaOperMP
2	Demo_OP_Timer_ECIFView	定时器	5	1	Demo_aaOperMP
3	Demo_OP_Timer_CRMView	定时器	5	1	Demo_aaOperMP
4	Demo_OP_Timer_EDWView	定时器	5	1	Demo_aaOperMP
5	Demo_OP_Timer_PMView	定时器	5	1	Demo_aaOperMP
6	Demo_EDT_CRM	定时器	5	2	Demo_aentbatdispat
7	Demo_ED_PM_Main	流程	10	2	Demo_aentbatdispat

4.2.4.1.2 查询指定容器的详细信息

- ◆ 命令: tccinfo
- ◆ 语法: tccinfo [容器号]
- ◆ 功能概述: 该命令功能为查询指定流程的详细信息。若不录入参数, 则查询当前流程信息。另外, 若没有设置容器环境, 则必须指定容器号。
- ◆ 命令前置条件: 必须启动调度服务器。

具体使用如下:

```
FLOWC_[0046*]> tccinfo 1
```

定时器基本信息如下:

```
-----  
应用工程名称      : Demo_aaOperMP  
定时器编号       : 1  
定时器名称       : Demo_OP_Time_MainOper  
定时器描述       : 企业调度及监控运维操作总视图  
启动模块         : MainModul  
最大并行度       : 5  
发布情况         : 1-已发布  
EM同步情况       : 0-未同步  
主备同步情况     : 0-未同步  
创建者           : demo  
所属用户组       : demo_enp  
拥有用户         : demo_enp  
创建时间         : 20090907002902  
流程权限         : rwxr-x---  
是否签出         : 0-签入  
-----
```

```
FLOWC_[0046*]>
```

4.2.4.1.3 查询当前容器下所有模块信息

- ◆ 命令: listmodul
- ◆ 语法: listmodul
- ◆ 功能概述: 该命令功能为查询当前容器下所有模块信息。
- ◆ 命令前置条件:

 必须启动调度服务器。

 必须设置容器编号。

具体使用如下:

```
FLOWC_[0046*]> listmodul
```

模块名称	模块描述
MainModul	主模块
SubModul0	

共 2 条记录

```
FLOWC_[0046*]>
```

4.2.4.1.4 查询当前容器下所有变量信息

- ◆ 命 令: listvar
- ◆ 语 法: listvar
- ◆ 功能概述: 该命令功能为查询当前容器下所有变量信息。
- ◆ 命令前置条件:
 -  必须启动调度服务器。
 -  必须设置容器编号。

具体使用如下:

```
FLOWC_[0046*]> listvar
```

公私	变量名称	类型	变量值	说明
私有	dbuser	普通变量	dbuser	数据库用户名
公有	test	普通变量	value	描述
公有	home	普通变量	/home/taskctl	根路径

共 3 条记录

4.2.4.1.5 查询当前容器所有任务节点信息

- ◆ 命 令: listtask
- ◆ 语 法: listtask [任务名称]
- ◆ 功能概述: 该命令功能为查询当前容器所有任务信息。若录入了参数, 系统将以参数为关键词模糊查找匹配的任务信息。
- ◆ 命令前置条件:
 -  必须启动调度服务器。

 必须设置容器编号。

具体使用如下：

```
FLOWC_[0046*]> listtask
```

ID号	模块名称	作业名称	作业状态
2	MainModul	MainModul_beginjob	1-未执行
3	MainModul	MainModul_JobNode0	1-未执行
4	MainModul	MainModul_endjob	1-未执行

共 3 个作业

4.2.4.1.6 查询当前容器所有任务(组)节点信息

- ◆ 命 令：listnode
- ◆ 语 法：listnode [任务名称]
- ◆ 功能概述：该命令功能为查询当前容器所有任务（组）节点信息。若录入了参数，系统将以参数为关键词模糊查找匹配的任务（组）节点信息。
- ◆ 命令前置条件：
 -  必须启动调度服务器。
 -  必须设置容器编号。

具体使用如下：

```
FLOWC_[0046*]> listnode
```

ID号	类型	模块名称	节点名称	节点描述
1	串行	MainModul	MainModul_rootnode	
2	作业	MainModul	MainModul_beginjob	
3	作业	MainModul	MainModul_JobNode0	
4	作业	MainModul	MainModul_endjob	

共 4 个节点

4.2.4.1.7 查询当前容器指定任务(组)节点详细信息

- ◆ 命 令：taskinfo
- ◆ 语 法：taskinfo [任务号]
- ◆ 功能概述：该命令功能为查询当前容器指定任务(组)节点详细信息。
- ◆ 命令前置条件：

- ✚ 必须启动调度服务器。
- ✚ 必须设置容器编号。
- ✚ 必须指定任务号。

具体使用如下:

```
FLOWC_[0046*]> taskinfo 2
```

```
流程作业(组)节点'MainModul_beginjob'信息:
```

```
-----[ No.046.0002 information ]-----
```

```
基本属性->
```

```
tccid      : 46
nodeno     : 2
nodename   : MainModul_beginjob
typename   : begin
progrname  :
para       :
exppara    :
agentid    :
monititle  :
modulname  : MainModul
jobdesc    :
```

```
控制属性->
```

```
runattr    : 0-作业
lastnode   : MainModul_rootnode
datetype   : msystime
period     :
maxnum     : 1
priority   : 0
virresouse : 0
condition  :
ostr       :
lean       :
cycle      : 1
egnoreerr  : N-不忽略错误
successv   : 0-10
errorv     :
failedv    :
warnningv  :
```

```
-----
FLOWC_[0046*]>
```

4.2.4.2 平台信息查询

4.2.4.2.1 查询调度服务器列表

- ◆ 命 令: listsvr
- ◆ 语 法: listsvr
- ◆ 功能概述: 该命令功能为查询整个平台的调度服务器信息列表。

具体使用如下:

```
FLOWC_[0046*]> listsvr
```

编号	服务名称	IP地址	端口	节点状态	下级节点列表
2	svrnode	192.168.0.190	18581	5-启动状态	

共 1 条记录

```
FLOWC_[0046*]>
```

4.2.4.2.2 查询服务器下所有工程信息

- ◆ 命令: listapp
- ◆ 语法: listapp
- ◆ 功能概述: 该命令功能为查询调度服务器下所有工程信息列表。

具体使用如下:

```
FLOWC_[0046]> listapp
```

应用编号	应用名称	描述
1	Demo_aaOperMP	范例:企业批量调度及监控操作总控制台
2	Demo_aentbatdispat	范例:企业批量调度及监控一站式解决方案
3	Demo_entautoflowpr	范例:企业运维其它自动流程管理解决方案
4	Demo_myworkspace	范例:我的地盘
5	Demo_taskctltec	范例:TASKCTL设计主要技术点
6	project1	应用工程1

共 6 条记录

```
FLOWC_[0046]>
```

4.2.4.2.3 查询整个平台 CIR 节点信息

- ◆ 命令: listcirm
- ◆ 语法: listcirm
- ◆ 功能概述: 该命令功能为查询整个平台的 CIR 节点信息列表。

具体使用如下:

```
FLOWC_[0046]> listcirn
```

编号	节点名称	节点类型	节点状态	运行数	上级节点	握手状态
2	svrnode	2-控制服务节点	5-启动状态	5	1	2-主机握手

共 1 条记录

4.2.4.2.4 查询指定 CIR 节点详细信息

- ◆ 命 令: cirninfo
- ◆ 语 法: cirninfo [CIR 节点号]
- ◆ 功能概述: 该命令功能为查询平台指定 CIR 节点详细信息。

具体使用如下:

```
FLOWC_[0046]> cirninfo 2
```

控制节点'svrnode'信息:

```
-----
id          : 2
svrid       : 2
type        : 2-控制服务节点
name        : svrnode
nodemem     : 调度服务节点
ip          : 192.168.0.190
port        : 18581
runmaxnum   : 100
currrunnum  : 5
maxvirresval : 1000
currvirresval : 0
errdelay    : 60
msgcommlog  : N
msgscancel  : 30
logsavedays : 10
vjobflag    : N
vjobrttime  : 20
lastnode    : 1
nextnode    :
cup_used    : 57.03
disk_used   : 36.00
state       : 5-启动状态
-----
```

```
FLOWC_[0046]>
```


5 监控平台应用系统详解

5.1 登录系统

ctlmoni 交互应用程序，在登录前需要启动调度服务，在登录过程中需要用户输入用户名，密码方可进入到该应用系统中，如下：

```
[test_cs_50@localhost flowcfg]$ ctlmoni
CTL*Moni: TASKCTL平台运维监控软件 Ver5.0.1 2017年06月16日 09:45:53
版权所有：成都塔斯克信息技术有限公司
授权用户：

登录用户：admin
登录密码：

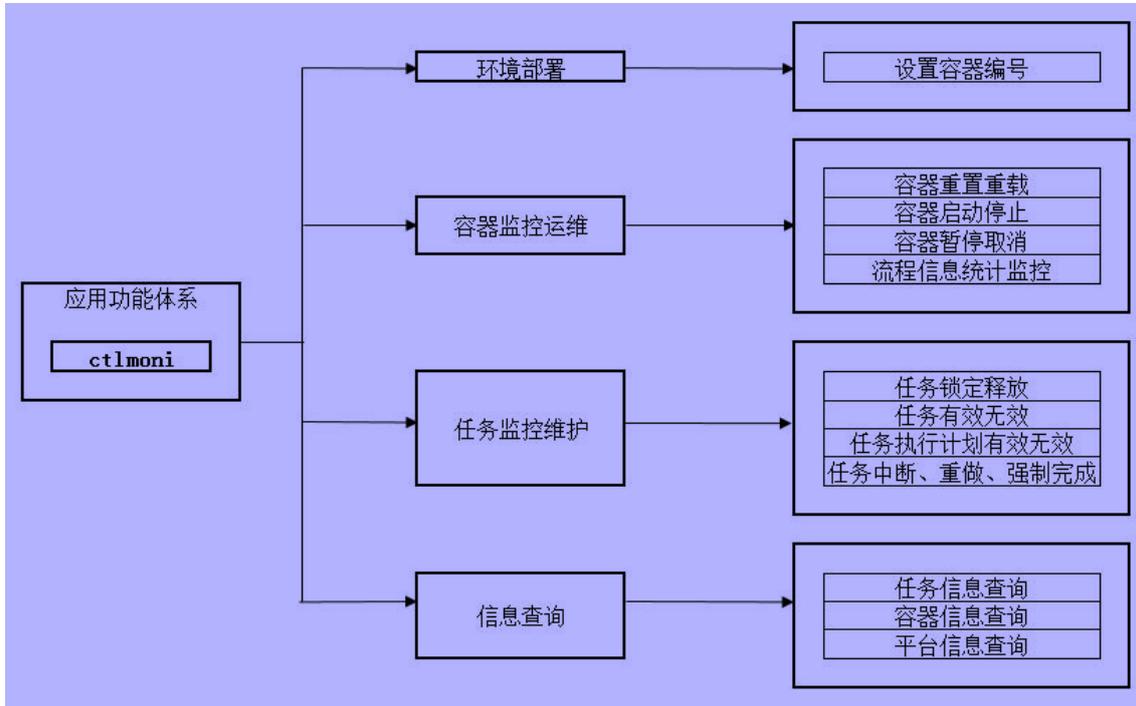
~~~~~
                软件未注册
            当前产品安装ID为：017E-09WW-05PB-00T8

            请登录www.taskctl.com获取授权
            ~~~~~
~~~~~ 官方网站：www.taskctl.com ~~~~~ 【未注册】 ~~~~~
** 温馨提示 **
    'help'命令可以获取所有命令的帮助信息
MONI_[0046]>
```

通过上图，可以得到：版本号、系统日期时间、使用单位以及该版本注册情况等基本信息。

5.2 功能介绍

Ctlmoni 的功能如下图所示：



用户也可以在命令行输入 `help` 帮助命令，查看当前 `ctlmoni` 监控平台的全部命令功能介绍。

5.2.1 环境部署

5.2.1.1 设置容器号

◆ 命令：`tccid`

◆ 语法：`tccid [容器号]`

参数说明：容器号 → 需要进入操作的容器号，不可缺省

◆ 功能概述：该命令功能为设置容器号，如果要对容器进行操作，那么系统必须进入到该容器下才可进行如容器维护、任务维护等相关操作。

◆ 命令前置条件：必须先启动调度服务器

具体使用如下：

```
MONI_[0002]> tccid 46
```

```
MONI_[0046]>
```

5.2.2 容器监控维护

5.2.2.1 容器重载和重置

5.2.2.1.1 容器重载

- ◆ 命令: load
- ◆ 语法: load
- ◆ 功能概述: 该命令功能为在监控平台应用系统下重新加载当前容器最新配置信息, 把容器从发布区加载到工作区。一般在 **ctlflowc** 系统里编译过后的容器放在发布区, 并没有加载到工作区, 所以用户要运行最新流程信息, 必须重新加载该容器信息。
- ◆ 命令前置条件:
 - ◆ 必须启动调度服务器。
 - ◆ 必须设置容器编号。
 - ◆ 容器必须处于停止状态。

具体使用如下:

```
MONI_[0046]> load
加载会覆盖目前活动配置信息, 是否确定(y/n)y
最新配置加载成功
MONI_[0046]>
```

5.2.2.1.2 容器重置

- ◆ 命令: reset
- ◆ 语法: reset
- ◆ 功能概述: 该命令功能为在监控平台应用系统下重置容器, 即让当前容器以及容器下面的任务归于初始化状态。
- ◆ 命令前置条件:
 - ◆ 必须启动调度服务器。

- ◆ 必须设置容器编号。
- ◆ 容器必须处于停止状态。

具体使用如下:

```
MONI_[0046]> reset  
  
重置会使流程重头运行, 是否确定(y/n)y  
  
重置成功  
MONI_[0046]>
```

5.2.2.2 容器启动和停止

5.2.2.2.1 容器启动

- ◆ 命令: starttcc
- ◆ 语法: starttcc
- ◆ 功能概述: 该命令功能为在运行应用系统平台下启动当前服务器下的当前容器。只有容器启动了, 一个调度才能正在运行起来。每个任务才能按照一定的逻辑规则运行起来。
- ◆ 命令前置条件:
 - ◆ 必须启动调度服务器。
 - ◆ 必须设置容器编号。
 - ◆ 容器必须处于停止状态。

具体使用如下:

```
MONI_[0038]> starttcc  
请确认启动模式[0-正常模式, 1-自由模式, 2 - 虚拟模式, 缺省为正常模式]: 0  
变量名-workdate[当前值:20110101]:  
变量名-db_name[当前值:abc]:  
变量名-db_user[当前值:abc]:  
变量名-db_passwd[当前值:abc]:  
  
确认输入(y/n)y  
  
是否启动(y/n)y  
  
流程启动成功!  
MONI_[0038]>
```

如上图所示，启动时，会让用户输入启动模式和相关私有变量值。如果启动成功，用户可以使用 `tccinfo` 命令查看当前容器状态是否是正在运行状态；管理员同样应该检查核心平台 `ctlfdc` 调度控制中心平台进程是否启动。如果失败，系统返回相应错误信息。

5.2.2.2.2 停止容器

- ◆ 命令： `stoptcc`
- ◆ 语法： `stoptcc`
- ◆ 功能概述：该命令功能为在运行应用系统下停止当前服务器下的当前已经启动的容器，如果当前容器有任务正在运行，那么等待当前任务运行完成后再停止容器。
- ◆ 命令前置条件：
 - ◆ 必须启动调度服务器。
 - ◆ 必须设置容器编号。
 - ◆ 容器必须处于启动状态。

具体使用如下：

```
MONI_[0038]> stoptcc  
  
流程成功退出  
MONI_[0038]>
```

如上图所示，如果停止成功，用户可以使用 `tccinfo` 命令查看当前容器状态是否是退出状态；管理员同样应该检查核心平台 `ctlfdc` 调度控制中心平台进程是否成功退出。如果失败，系统返回相应错误信息。

5.2.2.3 容器暂停和取消

5.2.2.3.1 容器暂停

- ◆ 命令: pause
- ◆ 语法: pause
- ◆ 功能概述: 该命令功能为在监控平台应用系统下暂停当前启动的容器。
- ◆ 命令前置条件:
 - ◆ 必须启动调度服务器。
 - ◆ 必须设置容器编号。
 - ◆ 容器必须处于启动状态。

具体使用如下:

```
MONI_[0038]> pause  
命令发送成功  
MONI_[0038]>
```

如上图所示,如果成功,用户可以使用 `tccinfo` 命令查看当前容器状态是否是暂停状态。如果失败,系统返回相应错误信息。

5.2.2.3.2 容器暂停取消

- ◆ 命令: unpause
- ◆ 语法: unpause
- ◆ 功能概述: 该命令功能为在监控平台应用系统下取消当前启动容器的暂停。
- ◆ 命令前置条件:
 - ◆ 必须启动调度服务器。
 - ◆ 必须设置容器编号。
 - ◆ 容器必须处于启动状态。

具体使用如下:

```
MONI_[0038]> unpause
```

```
命令发送成功
MONI_[0038]>
```

如上图所示, 如果成功, 用户可以使用 `tccinfo` 命令查看当前容器状态是否是正在运行状态。如果失败, 系统返回相应错误信息。

5.2.2.4 流程信息统计

5.2.2.4.1 分组监控当前流程运行状况

- ◆ 命 令: `total`
- ◆ 语 法: `total`
- ◆ 功能概述: 该命令功能为按任务类分组动态监控当前流程运行状况。它主要完成的事情是在核心平台按任务类进行统计计算当前流程的运行状况。在这里需要用户对 `<monititle>` 标签的使用, 详情请参照前面的《TASKCTL 流程开发设计手册》里的 `xml` 代码相关章节。
- ◆ 命令前置条件:
 - ✚ 必须启动调度服务器。
 - ✚ 必须设置流程编号。

具体使用如下图所示:

流程监控汇总信息

调度批次: 20170616112403
流程名称: 1_project1_Flow1

系统日期: 2017-06-16
流程状态: 正在运行

任 务 类	总数	未运	正执	错误	正确	其它
串行组一	2	1	0	0	1	0
并行组一	3	0	0	0	3	0
并行组二	6	0	1	0	5	0
串行组二	3	3	0	0	0	0
合计	14	4	1	0	9	0

'CTRL+C' 退出

以上信息是动态监控每个任务类的执行总体情况。具体统计方式请参照 `help`

命令详细说明。

5.2.2.4.2 监控指定任务状态的任务列表

◆ 命 令: moni

◆ 语 法: moni [任务状态列表]

参数说明: 任务状态列表 → 需要监控的任务状态列表, 不可缺省, 如 监控 1,2,3 状态的任务用“moni 1,2,3”命令

◆ 功能概述: 该命令功能为在运行应用系统平台下动态监控指定任务状态的任务运行情况, 任务状态列表之间用“,”号分隔符。任务状态详见附件任务状态清单。

◆ 命令前置条件:

 必须启动调度服务器。

 必须设置流程编号。

具体使用如下:

流程作业指定状态监控

正在监控状态:1,2,3,4,10,11,

ID号	作业名称	状态	开始时间	结束时间
2	MainModul_beginjob	11-成功	16 11:38:42	16 11:38:42
3	then	11-成功	16 11:38:42	16 11:38:42
5	cntMAgent1_sh	11-成功	16 11:38:42	16 11:38:45
6	cntMAgent11_sh	11-成功	16 11:38:45	16 11:38:48
7	cntMAgent12_sh	11-成功	16 11:38:49	16 11:38:51
9	cntMAgent13_sh	2-正在执行	16 11:38:52	
10	cntMAgent14_sh	2-正在执行	16 11:38:52	
11	cntMAgent15_sh	2-正在执行	16 11:38:52	
13	cntMAgent16_sh	2-正在执行	16 11:38:52	
14	cntMAgent17_sh	1-未执行		
15	cntMAgent18_sh	1-未执行		
17	cntMAgent19_sh	1-未执行		
18	cntMAgent20_sh	1-未执行		
19	cntMAgent101_sh	1-未执行		
20	MainModul_JobNode0	1-未执行		
21	MainModul_endjob	1-未执行		

共 16 个作业

'CTRL+C' 退出

以上信息是时时监控指定任务状态的详细情况。具体统计方式请参照 help 命令详细说明。

5.2.2.4.3 监控所有流程的综合统计信息

◆ 命令: mtall

◆ 语法: mtall [汇总方式] [应用工程号]

参数说明: 汇总方式 → 按不同方式进行汇总: 1-按服务器汇总、2-按应用汇总、3-按流程汇总。如果缺省默认按流程汇总

应用工程号 → 当汇总方式为“按流程汇总”时,此时可以统计某个工程下的流程汇总信息。如果缺省默认汇总当前服务器下所有流程

◆ 功能概述: 该命令功能为监控所有流程的综合统计信息,只统计用户自定义类型任务。

◆ 命令前置条件: 必须启动调度服务器。

具体使用如下:

- ✓ **按服务器汇总:** 录入命令 `mtall 1` 后, 响应结果如下图所示:

~~~~~  
流程综合信息统计  
~~~~~

服务器名称	总数	未运	正执	错误	正确	其它
svrnode	3433	3433	0	0	0	0
合计:	3433	3433	0	0	0	0

'CTRL+C' 退出

以上信息是按当前调度服务器进行统计任务信息概况

- ✓ **按应用汇总:** 录入命令 `mtall 2` 后, 响应结果如下图所示:

~~~~~  
流程综合信息统计  
~~~~~

服务器名称	应用名称	总数	未运	正执	错误	正确	其它
2	1_project1	16	16	0	0	0	0
2	2_30DaiLiC	3417	3417	0	0	0	0
合计:		3433	3433	0	0	0	0

'CTRL+C' 退出

以上信息是对当前调度服务器下按工程进行统计任务情况概况

- ✓ **按流程统计:** 录入命令 `mtall 3` 或者 `mtall` 后, 响应结果如下图所示:

~~~~~  
流程综合信息统计  
~~~~~

ID	流程名称	流程状态	调度批次	总数	未运	正执	错误	正确	其它
1	1_project1_Flow1	正在开始	201706161	14	14	0	0	0	0
3	2_30DaiLiCeShi_Flow1	正在开始	201706160	34	34	0	0	0	0
4	2_30DaiLiCeShi_Flow3	未启动	201706051	0	0	0	0	0	0
5	2_30DaiLiCeShi_Flow2	正在开始	201706151	1127	1127	0	0	0	0
6	2_30DaiLiCeShi_Flow4	未启动	201706051	1127	1127	0	0	0	0
7	2_30DaiLiCeShi_Flow5	未启动	201706051	0	0	0	0	0	0
8	2_30DaiLiCeShi_Flow6	未启动	201706051	1127	1127	0	0	0	0
9	2_30DaiLiCeShi_Flow7	未启动	201706051	1	1	0	0	0	0
10	2_30DaiLiCeShi_Flow8	未启动	201706051	1	1	0	0	0	0
23	1_project1_Flow2	未启动	201706101	2	2	0	0	0	0
合计:				3433	3433	0	0	0	0

'CTRL+C' 退出

以上信息是对当前调度服务器按流程进行统计任务信息概况

如果录入命令 `mtall 3 1` 后, 响应结果如下图所示:

~~~~~  
流程综合信息统计  
~~~~~

ID	流程名称	流程状态	调度批次	总数	未运	正执	错误	正确	其它
1	1_project1_Flow1	正在开始	201706161	14	14	0	0	0	0
23	1_project1_Flow2	未启动	201706101	2	2	0	0	0	0
合计:				16	16	0	0	0	0

'CTRL+C' 退出

以上信息是对当前调度服务器下，指定工程下的按流程进行统计任务信息概况

5.2.2.5 容器信息查询

5.2.2.5.1 在线打开容器模块代码信息

◆ 命令: openmc

◆ 语法: openmc [模块名称]

参数说明: 模块名称 → 指定在线打开的模块代码名称, 如果缺省, 默认打开系统主模块

◆ 功能概述: 该命令功能为打开工作区的模块代码信息文件, 此文件只可作为查阅, 不可修改。该功能主要是方便用户开发任务出错的情况下, 能够及时查询到该任务所指定的模块代码文件信息, 以供用户参阅。如果要修改, 必须到开发平台才可修改。

◆ 命令前置条件:

 必须启动调度服务器编号。

 必须设置容器编号。

具体使用如下:

```
MONI_[0046]> openmc MainModul
```

如上图所示, 如果成功, 系统展示该模块代码信息给用户。如果失败, 系统返回相应错误信息。

5.2.2.5.2 查询当前或指定容器的基本信息

◆ 命令: tccinfo

◆ 语法: tccinfo [容器编号]

参数说明: 容器编号 → 需要查看的容器信息编号, 缺省默认显示当前容器基本信息

◆ 功能概述: 该命令功能为查询当前或者指定容器基本信息, 如果不录入容器编号, 系统默认为当前容器编号。

◆ 命令前置条件: 必须启动调度服务器。

具体使用如下:

```
MONI_[0001]> tccinfo
```

定时器基本信息如下:

```
-----  
应用工程名称      : Demo_aaOperMP  
定时器编号       : 1  
定时器名称       : Demo_OP_Time_MainOper  
定时器描述       : 企业调度及监控运维操作总视图  
启动模块         : MainModul  
最大并行度       : 5  
发布情况         : 1-已发布  
EM同步情况       : 0-未同步  
主备同步情况     : 0-未同步  
创建者           : demo  
所属用户组       : demo_enp  
拥有用户         : demo_enp  
创建时间         : 20090907002902  
流程权限         : rwxr-x---  
状态             : 6-退出  
-----
```

```
MONI_[0001]>
```

如上图所示, 如果成功, 系统展示容器基本信息给用户。如果失败, 系统返回相应错误信息。

5.2.2.6 容器在线修改

5.2.2.6.1 修改当前容器指定私有变量信息

◆ 命令: modivarv

◆ 语法: modivarv [变量名称] [变量值]

参数说明: 变量名称 → 需要修改的变量名称, 不可缺省

变量值 → 需要修改的变量值, 不可缺省

◆ 功能概述: 该命令功能为修改当前容器指定私有变量信息, 只能修改工作区的内容, 且是一次性, 如果流程任务重新运行, 该改变无效。

◆ 命令前置条件:

✚ 必须启动调度服务器。

✚ 必须设置容器编号。

具体使用如下:

```
MONI_[0046]> modivarv cfdate 20170620
```

修改成功

```
MONI_[0046]> listvar
```

公私	变量名称	类型	变量值	说明
私有	cfdate	普通变量	20170620	moni私有变量
公有	test	普通变量	value	描述
公有	home	普通变量	/home/taskctl	根路径

共 3 条记录

```
MONI_[0046]>
```

如上图所示, 如果成功, 本次任务未运行的任务都按修改过后的变量值运行。如果失败, 系统返回相应错误信息。

5.2.2.6.2 在线修改流程最大并行度

◆ 命令: paltune

◆ 语法: paltune [流程最大并行度数]

参数说明: 流程最大并行度数 → 需要修改的流程最大并行度值, 不可缺省, 且必须为数字

◆ **功能概述:** 该命令功能为在线修改流程最大并行度。

◆ **命令前置条件:**

✚ 必须启动调度服务器。

✚ 必须设置流程编号。

具体使用如下:

```
MONI_[0046]> paltune 8
确认(y/n)y
修改成功
MONI_[0046]>
```

如上图所示,如果成功,可以使用 `tccinfo` 查看流程最大并行度是否改变,同时使用 `moni 2` 命令查看正在执行的任务数量是否超过此次修改的最大并行度数。如果失败,系统返回相应错误信息。

5.2.2.7 日志管理维护

5.2.2.7.1 容器日志

◆ **命令:** `getflog`

◆ **语法:** `getflog`

◆ **功能概述:** 该命令功能为获取当前容器当前批次的调度日志。

◆ **命令前置条件:**

✚ 必须启动调度服务器。

✚ 必须设置容器编号。

具体使用如下:

```
MONI_[0046]> getflog
```

如上图所示,如果成功,系统展示调度日志信息给用户。如果失败,系统返回相应错误信息。

5.2.2.7.2 任务日志

◆ 命 令: getjlog

◆ 语 法: getjlog [任务号]

参数说明: 任务号 → 需要查看的任务运行日志的任务节点号, 不可缺省, 只能是用户自定义任务类型的任务号

◆ 功能概述: 该命令功能为获取当前容器当前批次指定任务的运行日志。

◆ 命令前置条件:

 必须启动调度服务器。

 必须设置容器编号。

具体使用如下:

```
MONI_[0046]> getjlog 3
```

如上图所示, 如果成功, 系统展示任务日志信息给用户。如果失败, 系统返回相应错误信息。

5.2.3 任务监控维护

5.2.3.1 任务锁与释放

5.2.3.1.1 任务锁

◆ 命 令: lock

◆ 语 法: lock [任务号]

参数说明: 任务号 → 需要锁定的任务节点号, 不可缺省

◆ 功能概述: 该命令功能为在监控平台应用系统下锁定当前容器指定任务, 该命令主要是让任务运行到此处时, 暂停下来, 不在往后运行。

◆ 命令前置条件:

 必须启动调度服务器。

 必须设置容器编号。

具体使用如下:

```
MONI_[0046]> lock 3  
  
命令发送成功  
MONI_[0046]>
```

如上图所示，如果成功，用户可以通过 `showlock` 查看该任务是否是锁标志；同时用户也可以通过 `taskinfo` 命令查看该任务的 `lock` 属性栏是否是锁标志。如果失败，系统返回相应错误信息。

5.2.3.1.2 锁释放

◆ 命令: `free`

◆ 语法: `free [任务号]`

参数说明: 任务号 → 需要释放的任务节点号，不可缺省，只能是用户自定义任务类型的任务号

◆ 功能概述: 该命令功能为在监控平台应用系统下锁释放当前容器指定任务。恢复任务继续向下运行。

◆ 命令前置条件:

✚ 必须启动调度服务器。

✚ 必须设置容器编号。

具体使用如下:

```
MONI_[0046]> free 3  
  
命令发送成功
```

如上图所示，如果成功，用户可以通过 `showlock` 查看该任务是否相关信息存在；同时用户也可以通过 `taskinfo` 命令查看该任务的 `lock` 属性栏是否是锁释放标志。如果失败，系统返回相应错误信息。

5.2.3.2 任务有效与无效

5.2.3.2.1 使当前容器指定任务无效

◆ 命令: dsbjob

◆ 语法: dsbjob [任务号]

参数说明: 任务号 → 让任务无效的任务节点号, 不可缺省, 只能是用户自定义任务类型的任务号

◆ 功能概述: 该命令功能为在监控平台应用系统下, 让当前容器下的指定任务为无效状态。如果任务无效, 则不会运行该任务。只能修改工作区的内容, 且是一次性, 如果流程任务重新运行, 该改变无效。如果要想修改永久有效, 请到开发平台修改。

◆ 命令前置条件:

✚ 必须启动调度服务器。

✚ 必须设置容器编号。

具体使用如下:

```
MONI_[0046]> dsbjob 5  
命令发送成功
```

如上图所示, 如果成功, 用户可以通过 `taskinfo` 命令查看该任务的 `enable` 属性栏是否是无效标志。如果失败, 系统返回相应错误信息。

5.2.3.2.2 使当前容器指定任务有效

◆ 命令: enbjob

◆ 语法: enbjob [任务号]

参数说明: 任务号 → 让任务有效的任务节点号, 不可缺省, 只能是用户自定义任务类型的任务号

◆ 功能概述: 该命令功能为在监控平台应用系统下让当前流程下的指定任务为有效状态。如果任务有效, 当任务运行到此处时则会运行该任务, 只能修改工作区的内容, 且是一次性, 如果流程任务重新运行, 该改变无效。如果要想修改永久有效, 请到开发平台修改。

◆ **命令前置条件:**

- ◆ 必须启动调度服务器。
- ◆ 必须设置容器编号。

具体使用如下:

```
MONI_[0046]> enbjob 5  
命令发送成功
```

如上图所示, 如果成功, 用户可以通过 `taskinfo` 命令查看该任务的 `enable` 属性栏是否是有效标志。 如果失败, 系统返回相应错误信息。

5.2.3.3 任务执行计划有效与无效

5.2.3.3.1 使当前流程下指定任务执行计划控制无效

◆ **命 令:** dsbjp

◆ **语 法:** dsbjp [任务号]

参数说明: 任务号 → 让任务执行计划无效的任务节点号, 不可缺省, 只能是用户自定义任务类型的任务号

- ◆ **功能概述:** 该命令功能为在监控平台应用系统下使当前流程下指定任务执行计划控制无效。如果任务执行计划无效, 当任务运行到此处时则不会检查该任务的执行计划。只能修改工作区的内容, 且是一次性, 如果流程任务新运行, 该改变无效。如果要让修改永久有效, 请到开发平台修改。

◆ **命令前置条件:**

- ◆ 必须启动调度服务器。
- ◆ 必须设置流程编号。

具体使用如下:

```
MONI_[0046]> dsbjp 5  
命令发送成功
```

如上图所示, 如果成功, 用户可以通过用户可以通过 `taskinfo` 命令查看该任务的 `periodenb` 属性栏是否是无效标志。 如果失败, 系统返回相应错误信息。

5.2.3.3.2 使当前流程下指定任务执行计划控制有效

◆ 命令: enbjp

◆ 语法: enbjp [任务号]

参数说明: 任务号 → 让任务执行计划有效的任务节点号, 不可缺省, 只能是用户自定义任务类型的任务号

◆ 功能概述: 该命令功能为在监控平台应用系统下使当前流程下指定任务执行计划控制有效。如果任务执行计划有效, 当任务运行到此处时则会检查该任务的执行计划。只能修改工作区的内容, 且是一次性, 如果流程任务重新运行, 该改变无效。如果要让修改永久有效, 请到开发平台修改。

◆ 命令前置条件:

 必须启动调度服务器。

 必须设置流程编号。

具体使用如下:

```
MONI_[0046]> enbjp 5
```

```
命令发送成功
```

如上图所示, 如果成功, 用户可以通过用户可以通过 `taskinfo` 命令查看该任务的 `periodenb` 属性栏是否是有效标志。如果失败, 系统返回相应错误信息。

5.2.3.4 任务中断

5.2.3.4.1 中断当前流程指定任务的运行

◆ 命令: stopjob

◆ 语法: stopjob [任务号]

参数说明: 任务号 → 让正在运行的任务中断的任务节点号, 不可缺省, 只能是用户自定义任务类型的任务号

◆ 功能概述: 该命令功能为中断当前流程指定任务的运行, 且只能中断用户自定义任务。任务中断后, 系统默认为该任务运行成功, 流程任务继续向下运行。手工中断指定任务的运行, 取决是否配置相应类型任务的停止

插件。如果具备相应的插件并配置正确，则我们就可以通过该程序发起停止命令。此命令主要是解决一些任务过长时间处于运行状态而进行人工干预。

◆ **命令前置条件：**

- ◆ 必须启动调度服务器。
- ◆ 必须设置流程编号。
- ◆ 流程必须为启动状态。

具体使用如下：

```
MONI_[0046]> stopjob 5
```

如上图所示，如果成功，用户可以通过用户可以通过 `listtask` 命令查看该任务的运行状态是否为成功完成标志，后续任务是否继续向下执行。如果失败，系统返回相应错误信息。

5.2.3.5 任务结束后操作

5.2.3.5.1 重做错误不在执行的任务

◆ **命令：** `runjob`

◆ **语法：** `runjob [任务号][标志]`

参数说明：任务号 → 需要重做的任务节点号，不可缺省，只能是用户自定义任务类型的任务号

[标志] → 0：只运行指定节点；1：运行指定节点及其所有依赖节点

◆ **功能概述：** 该命令功能为在当前流程下重做错误不在执行的指定任务，该命令弥补了用户在 ETL 调度时程序出现异常，而导致数据还原，重新开始调度的缺陷，用户可以在错误的地方开始执行。大大提高了效率。任务状态详见附录任务状态清单。

◆ **命令前置条件：**

-  必须启动调度服务器。
-  必须设置流程编号。
-  流程必须为启动状态。

具体使用如下:

```
MONI_[0046]> runjob 5 0  
命令发送成功
```

如上图所示,如果成功,用户可以通过 `listtask` 命令查看该任务的运行状态是否为正在运行标志。如果失败,系统返回相应错误信息。

5.2.3.5.2 强制将错误不在执行的任务设为成功状态

◆ 命令: `forceok`

◆ 语法: `forceok [任务号]`

参数说明: 任务号 → 需要强制完成的任务节点号,不可缺省,只能是用户自定义任务类型的任务号

◆ 功能概述: 该命令功能为强制将错误不在执行的任务设为成功状态,该命令解决了某些任务对后续任务无影响,用户可以暂时将其设置成成功状态,继续运行后续任务,以后集中解决错误任务,这样大大提高了效率。任务状态详见附录任务状态清单。

◆ 命令前置条件:

-  必须启动调度服务器。
-  必须设置流程编号。
-  流程必须为启动状态。

具体使用如下:

```
MONI_[0046]> forceok 6  
命令发送成功
```

如上图所示,如果成功,用户可以通过用户可以通过 `listtask` 命令查看该任务

的运行状态是否为成功标志, 后续任务是否继续向下运行。如果失败, 系统返回相应错误信息。

5.2.4 信息查询

5.2.4.1 容器信息查询

5.2.4.1.1 查询当前调度服务器下的容器列表

◆ 命令: listtcc

◆ 语法: listtcc [应用编号] [容器编号]

参数说明: 应用编号 → 查询出指定应用编号下所有容器信息, 缺省默认查询出所有容器信息

容器编号 → 查询出指定应用工程下的指定容器信息, 缺省默认查询出指定应用工程下所有容器信息

◆ 功能概述: 该命令功能为查询当前调度服务器下的容器列表。该信息来源于工作区。

◆ 命令前置条件:

必须启动调度服务器。

具体使用如下:

```
MONI_[0046]> listtcc
```

编号	容器名称	类型	并行度	应用号	应用名称
1	Demo_OP_Time_MainOper	定时器	5	1	Demo_aaOperMP
2	Demo_OP_Timer_ECIFView	定时器	5	1	Demo_aaOperMP
3	Demo_OP_Timer_CRMView	定时器	5	1	Demo_aaOperMP
4	Demo_OP_Timer_EDWView	定时器	5	1	Demo_aaOperMP
5	Demo_OP_Timer_PMView	定时器	5	1	Demo_aaOperMP
6	Demo_EDT_CRM	定时器	5	2	Demo_aentbatdispat

以上信息列举出当前调度服务器下的所有流程信息

```
MONI_[0046]> listtcc 1
```

编号	容器名称	类型	并行度	应用号	应用名称
1	Demo_OP_Time_MainOper	定时器	5	1	Demo_aaOperMP
2	Demo_OP_Timer_ECIFView	定时器	5	1	Demo_aaOperMP
3	Demo_OP_Timer_CRMView	定时器	5	1	Demo_aaOperMP
4	Demo_OP_Timer_EDWView	定时器	5	1	Demo_aaOperMP
5	Demo_OP_Timer_PMView	定时器	5	1	Demo_aaOperMP

共 5 条记录

以上信息列举出当前调度服务器下指定应用编号下所有流程信息

```
MONI_[0046]> listtcc 1 1
```

编号	容器名称	类型	并行度	应用号	应用名称
1	Demo_OP_Time_MainOper	定时器	5	1	Demo_aaOperMP

共 1 条记录

以上信息列举出了当前调度服务器下指定应用编号下指定流程列表

5.2.4.1.2 显示当前容器的参数变量信息列表

- ◆ 命令: listvar
- ◆ 语法: listvar
- ◆ 功能概述: 该命令功能为显示当前容器的参数变量信息列表。该信息来源于工作区。
- ◆ 命令前置条件:
 - ◆ 必须启动调度服务器。
 - ◆ 必须设置容器编号。

具体使用如下:

```
MONI_[0046]> listvar
```

公私	变量名称	类型	变量值	说明
私有	cfdate	普通变量	20170620	moni私有变量
公有	test	普通变量	value	描述
公有	home	普通变量	/home/taskctl	根路径

共 3 条记录

5.2.4.1.3 列举当前容器模块信息列表

- ◆ 命令: listmodul
- ◆ 语法: listmodul
- ◆ 功能概述: 该命令功能为列举当前容器模块信息列表。该信息来源于工作区。
- ◆ 命令前置条件:
 - ◆ 必须启动调度服务器。
 - ◆ 必须设置容器编号。

具体使用如下:

```
MONI_[0046]> listmodul
```

模块名称	模块描述
MainModul	主模块

共 1 条记录

5.2.4.1.4 列出当前容器任务节点列表

- ◆ 命令: listtask
- ◆ 语法: listtask
- ◆ 功能概述: 该命令功能为列出当前容器任务节点列表。该信息来源于工作区。
- ◆ 命令前置条件:
 - ◆ 必须启动调度服务器。
 - ◆ 必须设置容器编号。

具体使用如下:

```
MONI_[0046]> listtask
```

ID号	模块名称	作业名称	作业状态
2	MainModul	MainModul_beginjob	11-成功
3	MainModul	MainModul_JobNode0	11-成功
4	MainModul	MainModul_JobNode1	11-成功
5	MainModul	MainModul_JobNode2	10-失败
6	MainModul	MainModul_JobNode4	1-未执行
7	MainModul	MainModul_JobNode3	1-未执行
8	MainModul	MainModul_endjob	1-未执行

共 7 个作业

```
MONI_[0046]>
```

5.2.4.1.5 列出当前容器任务以及任务组节点列表

◆ 命令: listnode

◆ 语法: listnode

◆ 功能概述: 该命令功能为列出当前容器任务以及任务组节点列表。该信息来源于工作区。

◆ 命令前置条件:

 必须启动调度服务器。

 必须设置容器编号。

具体使用如下:

```
MONI_[0046]> listnode
```

ID号	类型	模块名称	节点名称	节点描述
1	串行	MainModul	MainModul_rootnode	
2	作业	MainModul	MainModul_beginjob	
3	作业	MainModul	MainModul_JobNode0	
4	作业	MainModul	MainModul_JobNode1	
5	作业	MainModul	MainModul_JobNode2	
6	作业	MainModul	MainModul_JobNode4	
7	作业	MainModul	MainModul_JobNode3	
8	作业	MainModul	MainModul_endjob	

共 8 个节点

5.2.4.2 任务属性信息查询

5.2.4.2.1 列举显示当前容器锁定任务

- ◆ 命 令: showlock
- ◆ 语 法: showlock
- ◆ 功能概述: 该命令功能为列举显示当前容器锁定任务。该信息来源于工作区。
- ◆ 命令前置条件:
 - ◆ 必须启动调度服务器。
 - ◆ 必须设置容器编号。

具体使用如下:

```
MONI_[0046]> showlock
```

作业号	作业名称
5	MainModul_JobNode2

共计 1 条记录

```
MONI_[0046]>
```

5.2.4.2.2 显示当前流程任务节点的依赖关系

- ◆ 命 令: showlean
- ◆ 语 法: showlean
- ◆ 功能概述: 该命令功能为显示当前流程任务节点的依赖关系。该信息来源于工作区。
- ◆ 命令前置条件:
 - ◆ 必须启动调度服务器。
 - ◆ 必须设置流程编号。

具体使用如下:

```
MONI_[0046]> showlean

ID号 作业名称                依赖作业号
-----
  2 MainModul_beginjob
  3 MainModul_JobNode0      2,
  4 MainModul_JobNode1      3,
  5 MainModul_JobNode2      4,
  6 MainModul_JobNode4      5,
  7 MainModul_JobNode3      6,
  8 MainModul_endjob        7,
-----

                                共 7 条记录

MONI_[0046]>
```

5.2.4.2.3 查询当前调度服务器下所有互斥资源的占有情况

- ◆ 命 令: sostr
- ◆ 语 法: sostr
- ◆ 功能概述: 该命令功能为查询当前调度服务器下所有互斥资源的占有情况。该信息来源于工作区。
- ◆ 命令前置条件:

必须启动调度服务器。

具体使用如下:

```
MONI_[0046]> sostr

占有流程号    占有作业(组)                资源名称
-----
-----

                                共计 0 条件记录

MONI_[0046]>
```

5.2.4.3 平台信息查询

5.2.4.3.1 列举调度服务器信息列表

- ◆ 命 令: listsvr
- ◆ 语 法: listsvr
- ◆ 功能概述: 该命令功能为查询出整个平台的调度服务器的一些基本信息。
该信息来源于工作区。

具体使用如下:

```
MONI_[0046]> listsvr
```

编号	服务名称	IP地址	端口	节点状态	下级节点列表
2	svrnode	192.168.0.190	18581	5-启动状态	

共 1 条记录

```
MONI_[0046]>
```

5.2.4.3.2 列举平台所有 CIR 节点信息

- ◆ 命 令: listcirm
- ◆ 语 法: listcirm
- ◆ 功能概述: 该命令功能为列举平台所有 CIR 节点信息。该信息来源于工作区。

具体使用如下:

```
MONI_[0046]> listcirm
```

编号	节点名称	节点类型	节点状态	运行数	上级节点	握手状态
2	svrnode	2-控制服务节点	5-启动状态	0	1	2-主机握手

共 1 条记录

```
MONI_[0046]>
```

5.2.4.3.3 显示指定平台 CIR 节点的信息

- ◆ 命 令: listcirn
- ◆ 语 法: listcirn [CIR 节点号]
- ◆ 功能概述: 该命令功能为显示指定平台 CIR 节点的信息。该信息来源于工作区。

具体使用如下:

```
MONI_[0046]> cirninfo 2
```

控制节点'svrnode'信息:

```
id          : 2
svrid       : 2
type        : 2-控制服务节点
name        : svrnode
nodemem     : 调度服务节点
ip          : 192.168.0.190
port        : 18581
runmaxnum   : 100
currrunnum  : 0
maxvirresval : 1000
currvirresval : 0
errdelay    : 60
msgcommlog  : N
msgscancel  : 30
logsavedays : 10
vjobflag    : N
vjobrtime   : 20
lastnode    : 1
nextnode    :
cup_used    : 68.29
disk_used   : 36.00
state       : 5-启动状态
```

```
MONI_[0046]>
```

6 附录

✓ 附录 1 流程状态清单

状态值	状态说明
0	初始化
1	暂停状态
2	正在运行状态
3	正在开始状态
4	正在结束状态
5	空闲
6	退出
7	结束状态
8	正在退出

✓ 附录 2 任务状态清单

状态值	状态名称	状态说明
-1	M_JOB_STATE_NETERR	网络错误
-2	M_JOB_STATE_NOCOMMFILE	通信文件不存在
-3	M_JOB_STATE_NOPLUGIN	无执行插件
-4	M_JOB_STATE_CTLNODEERR	控制接点错
-5	M_JOB_STATE_MSGERR	消息队列错
-7	M_JOB_STATE_SYSOERR	系统其它错
-8	M_JOB_STATE_NOISSUE	程序未发布
-9	M_JOB_STATE_ROUTEERR	错误路由
-10	M_JOB_STATE_NONETINFO	无网络配置信息
0	M_JOB_STATE_SYSBUSS	系统忙
1	M_JOB_STATE_INIT	未执行
2	M_JOB_STATE_RUNNING	正在执行
3	M_JOB_STATE_CYCLE	正在循环
4	M_JOB_STATE_ERR	错误需再次执行
10	M_JOB_STATE_FAILED	错误不再执行
11	M_JOB_STATE_SUCCESS	成功执行
12	M_JOB_STATE_OKPERIOD	因周期本期不执行通过
13	M_JOB_STATE_IGNOREERR	忽略错误通过
14	M_JOB_STATE_DISABLE	该任务因为无效通过