

OAMTool 工具使用指导书



作者：李斌

日期：2020-10-20

目录

1. [简介](#)
2. [安装和卸载](#)
3. [界面说明](#)
4. [控件说明](#)
5. [关键文件介绍](#)
6. [功能扩展指导](#)
7. [技巧及其他](#)

1. 简介

OAMTool 旨在打造一款简便，快捷，稳定且极具扩展性的服务器操作维护客户端；能够同时满足维护，测试和开发的需求，提高工作效率。目前此工具已经实现稳定的框架，具体各个界面的布局和功能支持自定义和动态扩展，而无需修改工具源代码。工具代码采用 MVVM 软件架构，界面逻辑和数据分离。

2. 安装和卸载

此工具为绿色免安装软件，开箱即用，且无任何其他第三方依赖；直接删除程序文件即可完全卸载。

运行环境：Win7/Win10 64 位

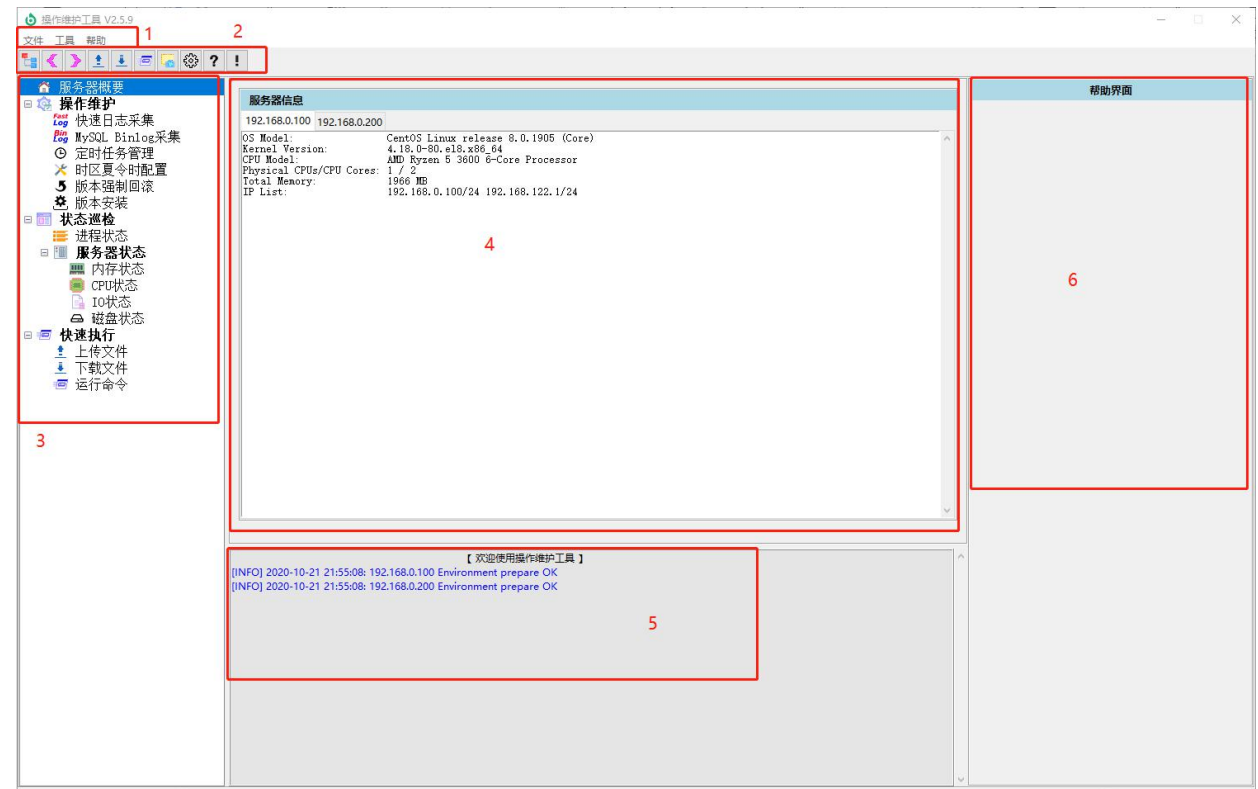
3. 界面说明

3.1. 双击 OAMTool.exe 运行软件后，首先弹出服务器登录界面：



序号	名称	说明
1	服务器信息输入栏	该栏中的输入框都是必填项，默认用户名为 ubp；用户密码为 eLTE@com123；root 密码为 eLTE@com；默认值目前可通过修改. \conf\setting. json 配置文件修改；后续提供界面修改功能。
2	“小眼睛”按钮	点住鼠标可查看隐藏的用户密码和 root 密码，释放鼠标变回*号。
3	添加登录服务器按钮	点击后可在服务器信息输入栏中增加一行，目前一次最多同时登录 4 台服务器。
4	一键登录按钮	点击后，会依次对服务器进行登录校验；登录成功后将切换至另一个操作界面。
5	菜单栏	用于对工具进行配置和提供帮助信息。
6	登录状态提示球	每台服务器 IP 输入框前均有对应的状态球，登录服务器过程的不同阶段将会显示不同的颜色，可使用户直观感受到当前服务器的登录状态。

3.2. 登录成功后，将出现另一个操作界面：

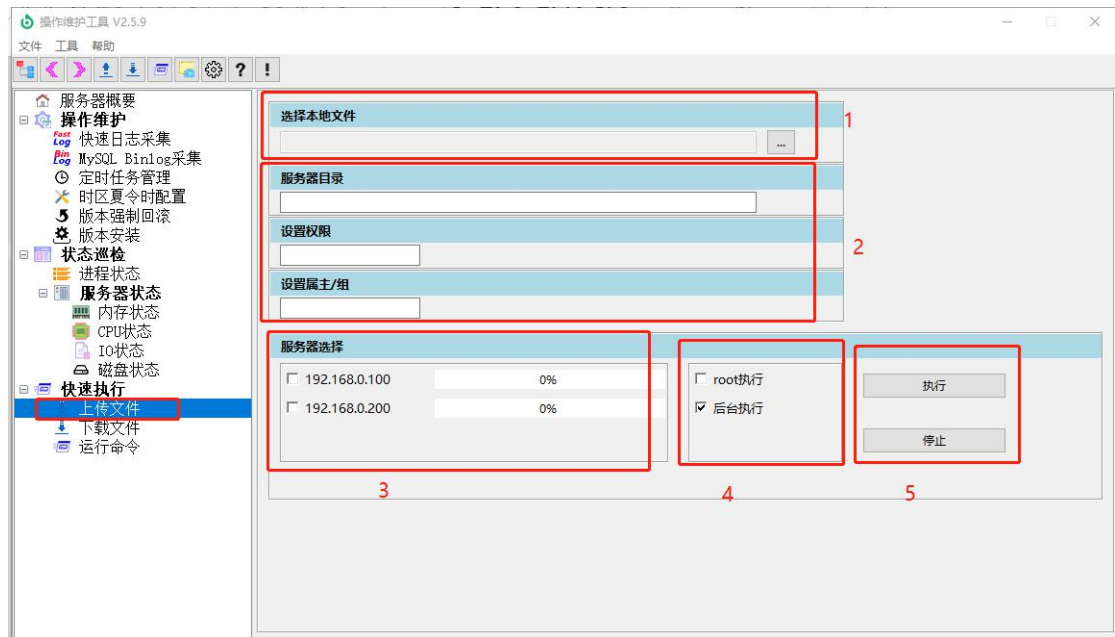


序号	名称	说明
1	菜单栏	提供部分额外功能。
2	快捷工具栏	用于快速进入指定的功能界面，可自定义。
3	目录导航栏	工具的主要功能栏，根据主配置文件生成。
4	主操作窗栏	各个功能的主操作界面，其控件根据主配置文件生成。

5	状态提示栏	执行操作的状态和结果提示栏，主要用于简短信息的提示。
6	帮助信息栏	点击帮助按钮后将显示该界面，用于显示操作界面的详细帮助信息。

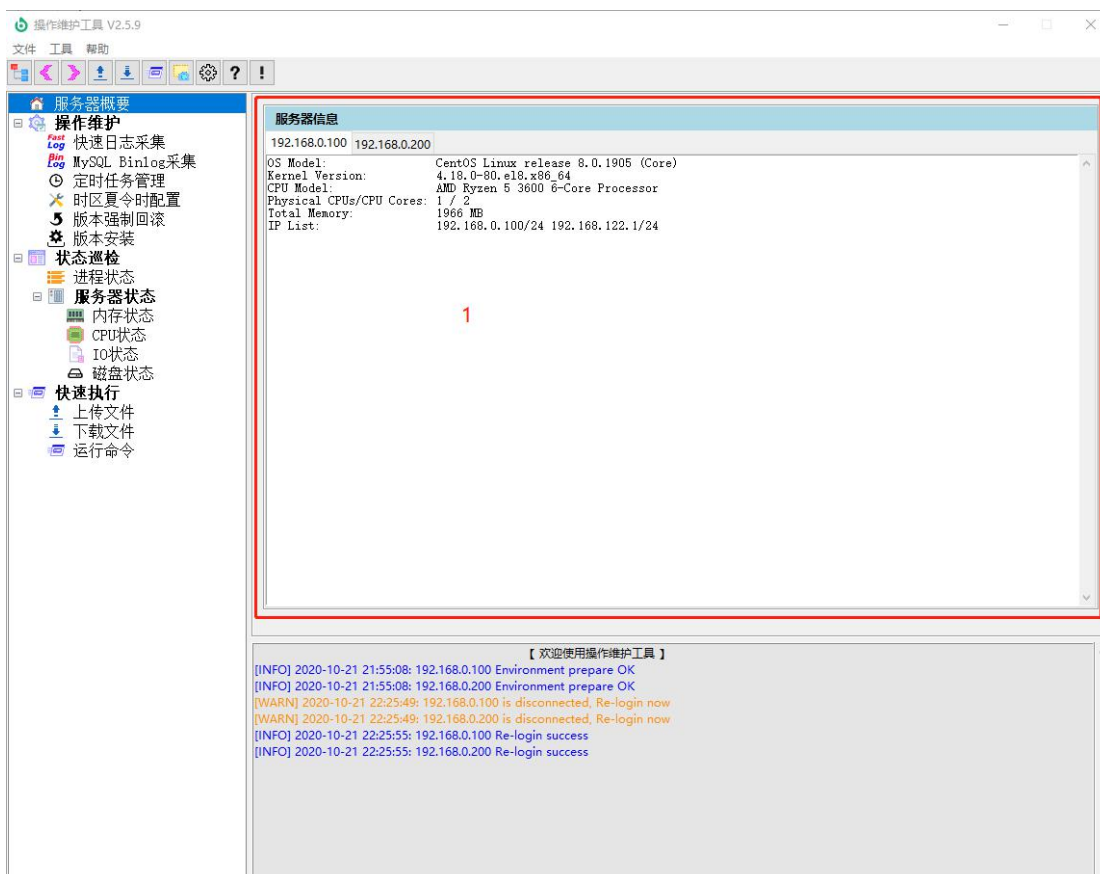
4. 控件说明

4.1. 例：上传文件



序号	控件名	说明
1	Button	文件选择按钮控件，点击右侧按钮可弹出 windows 选择本地文件窗体。
2	Entry	输入框控件，接受单行信息输入。
3	Checkbox	复选框控件，可同时勾选多台服务器，一同执行。
4	Checkbox	复选框控件，执行选项，默认以非 root 用户‘后台执行’。
5	Button	按钮控件，可对任务开始‘执行’或强行‘停止’。

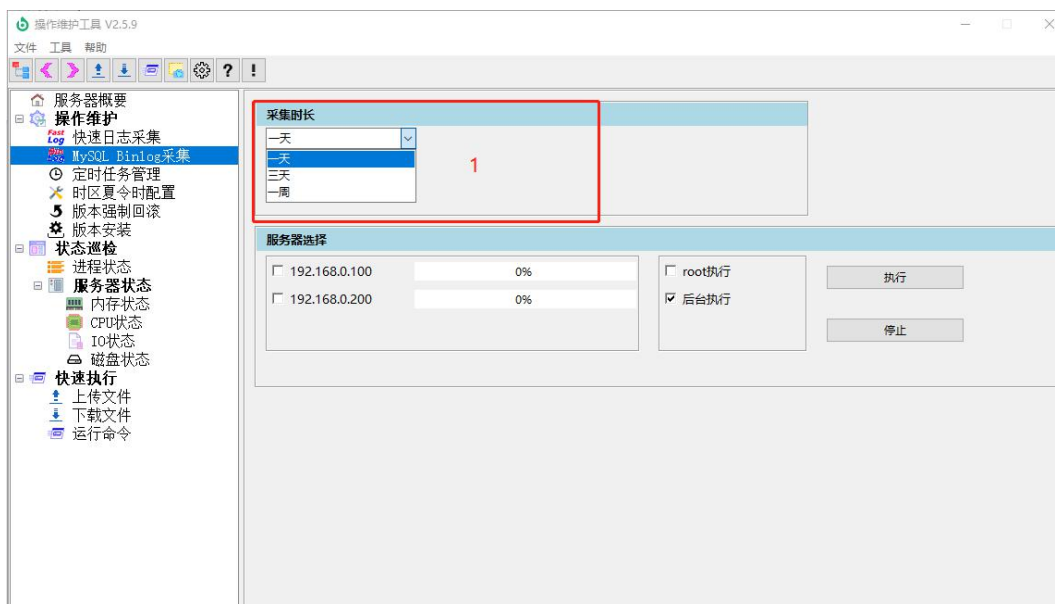
4.2. 例：服务器概要



点击进入界面自动刷新并显示状态，该类型界面无服务器 IP 选择执行栏。

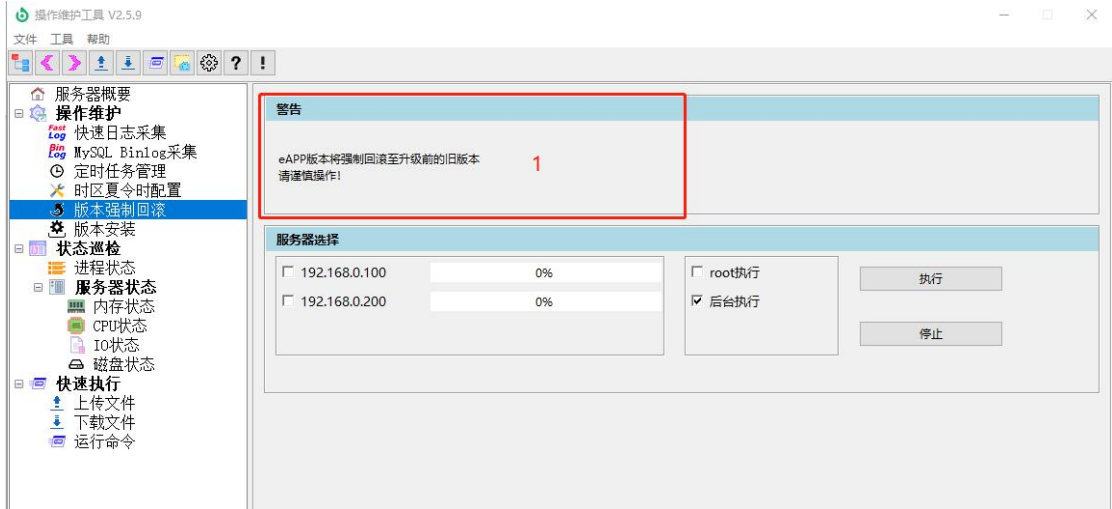
序号	控件名	说明
1	Notebook	分页显示控件，用于显示刚进入界面后自动获取的结果信息。

4.3. 例：MySQL Binlog 采集



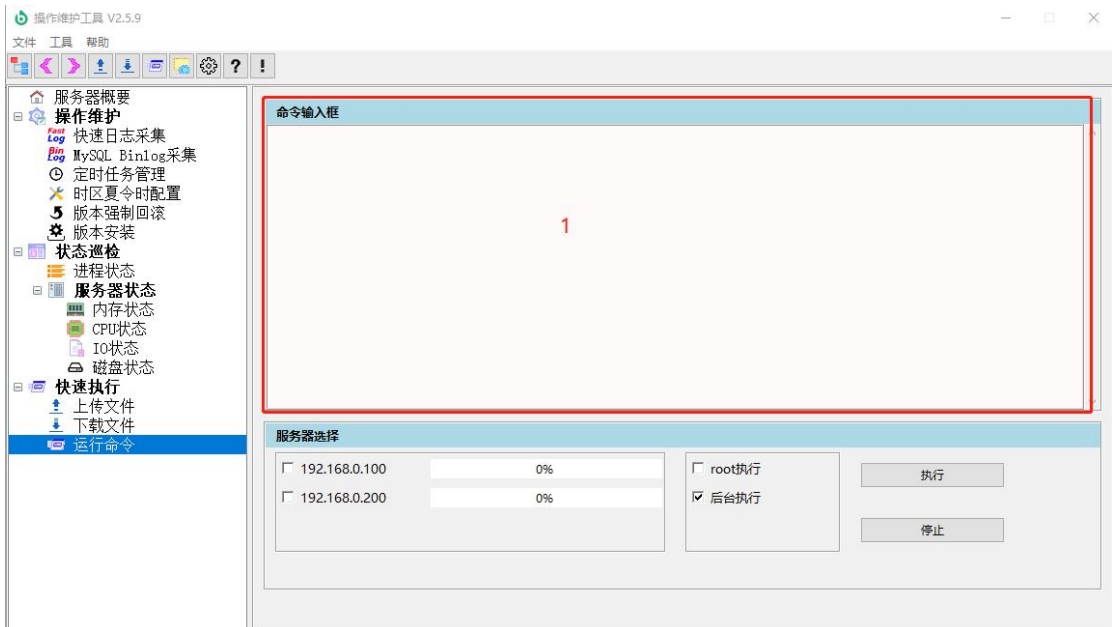
序号	控件名	说明
1	Combobox	单选框控件，从给定的一组值中选择一个。

4.4. 例：版本强制回滚



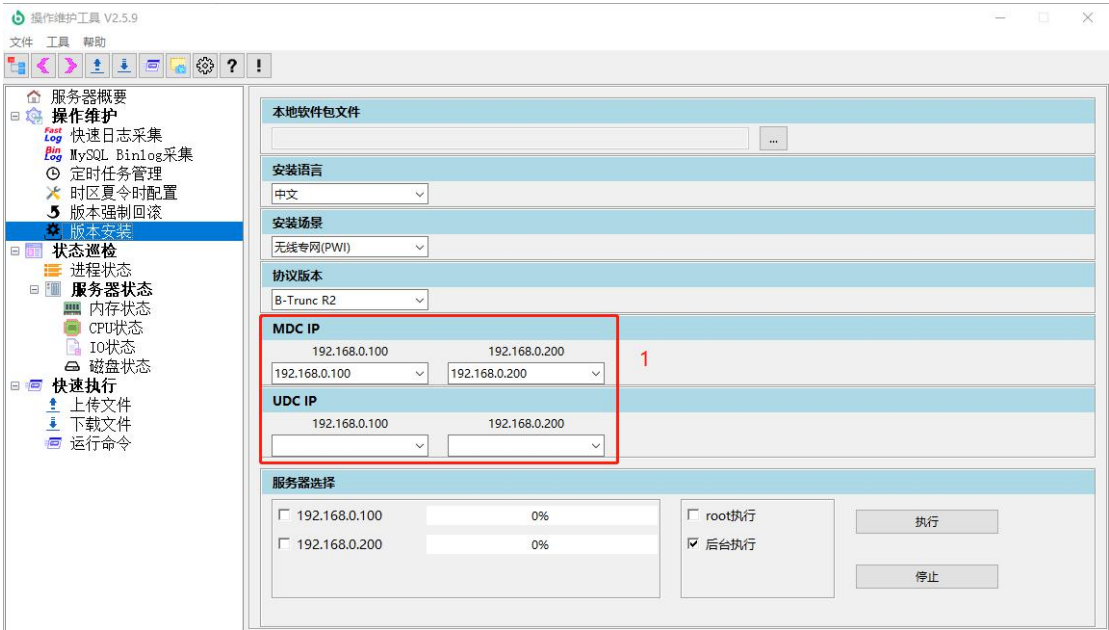
序号	控件名	说明
1	Label	信息显示控件，主要用于显示固定提示信息，也可用于显示刚进入界面后自动获取的结果信息。

4.5. 例：运行命令



序号	控件名	说明
1	Text	文本输入框控件，用于输入多行文本信息。

4. 6. 例：版本安装



序号	控件名	说明
1	MultiCombobox	多 IP 单选框控件，用于对不同服务器差异化选择。

5. 关键文件介绍

5. 1. 主目录介绍

名称	修改日期	类型	大小
cmds	2020/10/21 23:48	文件夹	
conf	2020/10/21 23:48	文件夹	
download	2020/10/22 0:05	文件夹	
image	2020/10/21 23:48	文件夹	
log	2020/10/22 0:05	文件夹	
OAMTool.exe	2020/10/21 23:47	应用程序	26,334 KB

名称	说明
cmds/	运行目录，其中有服务器运行脚本所在目录 shell/和运行过程产生的临时目录 temp/。
conf/	配置文件目录，其中有工具基本设置的配置文件 settings.json 和各个界面定义配置文件 dependance.json。
download/	文件下载目录，工具下载服务器文件目录，以 IP 作为子目录区分。
image/	图标和图片所在目录。
log/	工具日志文件输出目录。
OAMTool.exe	工具的主执行程序。

5.2. settings.json 文件介绍

```
{
  "Setting": {
    "tool_named": "操作维护工具",
    "tool_version": "3.0.9",
    "log_level": "debug",
    "log_size": 10485760,
    "server_home": "/home/Bingo",
    "retry_times": 2,
    "keepalive_interval": 5,
    "passwords": {
      "username": "ubp",
      "userpassword": "eLTE@com123",
      "rootpassword": "eLTE@com"
    }
  },
  "Timer": {
    "RefreshCache": {
      "Interval": 5,
      "Interface": [
        "get_cache_key_values.sh __IP__NETCARD__",
        "get_cache_key_values.sh __UBP_PROC__"
      ]
    },
    "RefreshFile": {
      "Interval": 30,
      "FileList": [
        "/home/ubp/logs/tracecollection/*.csv"
      ]
    }
  }
}
```

区域	索引项	说明
Setting	tool_named	工具 Title 名称。
	tool_version	工具版本号。
	log_level	工具日志打印级别。
	log_size	日志回滚大小，单位 Bite。
	server_home	服务器脚本部署的根目录。
	retry_times	失败尝试次数；（登录，获取结果等动作）
	keepalive_interval	与服务器连接保活的周期，单位秒。
	passwords	指定登录界面默认自动填入的用户名和密码以及 root 密码。

Timer	RefreshCache	<p>登录服务器后 自动周期刷新内部变量（WidgetValues 中的内部变量）的定时器。</p> <p>Interval: 定时器周期，单位秒；</p> <p>Interface: 接口，当定义内部变量名时，需通过此脚本接口获取该变量的具体值；具体接口实现参照 <code>get_cache_key_values.sh</code> 脚本内容。</p>
	RefreshFile	<p>登录服务器后自动周期从服务器下载并更新本地文件的定时器。</p> <p>Interval: 定时器周期，单位秒；</p> <p>FileList: 服务器文件列表绝对路径；工具将周期自动下载列表中的文件到本地的 <code>.\download\<IP>\DATA</code> 目录；支持正则表达式。</p>

5.3. dependance.json 文件介绍

该文件为工具界面定义配置文件，为工具最核心文件；

文件较大，分区块介绍；

A) “Images” 图片区：

```
"Images": {
  "ICO": ".\\image\\item.ico",
  "GUIDE": ".\\image\\guide.png",
  "ABOUT": ".\\image\\about.png",
  "LGN_HEAD": ".\\image\\login_head.jpg",
  "LGN_EYE": ".\\image\\login_eye.png",
  "TB_EXPAND": ".\\image\\toolbar_expand.png",
  "TB_LAST_ONE": ".\\image\\toolbar_last_one.png",
  "TB_NEXT_ONE": ".\\image\\toolbar_next_one.png",
  "TB_SCREEN_CUT": ".\\image\\toolbar_screen_cut.png",
  "TB_SETTING": ".\\image\\toolbar_setting.png",
  "TB_HELP": ".\\image\\toolbar_help.png",
  "TB_INFO": ".\\image\\toolbar_info.png",
  "TV_HOME": ".\\image\\treeview_home.png",
  "TV_OAM": ".\\image\\treeview_oam.png",
  "TV_OAM_SET": ".\\image\\treeview_oam_set.png",
  "TV_STATE": ".\\image\\treeview_state.png"
},
```

用于指定图标索引名和图标路径的对应关系；

B) “Tree” 目录导航树区：

```
"Tree": [
  {
    "Text": "服务器概要",
    "Image": "TV_HOME",
    "Page": {
      "Widgets": "HomePageWidgets",
      "Shell": "home_page.sh",
      "Attrs": []
    },
    "SubTree": []
  },
  {
    "Text": "操作维护",
    "Image": "TV_OAM",
    "Page": "NA",
    "SubTree": [
      {
        "Text": "快速日志采集",
        "Image": "TV_FAST_LOG",
        "Page": {
          "Widgets": "FastLogPageWidgets",
          "Shell": "collect_fast_logs.sh",
          "Attrs": ["OperateButtons"]
        },
        "SubTree": []
      },
      {
        "Text": "MySQL Binlog 采集",
        "Image": "TV_BINLOG",
        "Page": {
          "Widgets": "MysqlBinlogPageWidgets",
          "Shell": "collect_binlog.sh",
          "Attrs": ["OperateButtons"]
        },
        "SubTree": []
      }
    ]
  }
],
```

用于定义目录导航栏树型结构关系，界面小图标，以及界面控件/脚本和属性等信息。

C) “__ThisIsPageWidgets__” 各界面控件定义区：

```
"__ThisIsPageWidgets__": "NA",

"HomePageWidgets": [
  {
    "WidgetType": "Notebook",
    "WidgetTips": ["服务器信息"],
    "WidgetValues": [],
    "WidgetParams": {"Size": [900, 500, 900, 500]},
    "WidgetAttrs": ["ShowEnterResult"],
    "WidgetActions": []
  }
],

"FastLogPageWidgets": [
  {
    "WidgetType": "Checkbox",
    "WidgetTips": ["日志类型"],
    "WidgetValues": ["日志类型 A", "日志类型 B", "日志类型 C", "日志类型 D"],
    "WidgetParams": {"Size": [600, 200, 40, 1]},
    "WidgetAttrs": [],
    "WidgetActions": []
  }
],

"MysqlBinlogPageWidgets": [
  {
    "WidgetType": "Combobox",
    "WidgetTips": ["采集时长"],
    "WidgetValues": ["一天", "三天", "一周"],
    "WidgetParams": {"Size": [600, 100, 20, 1]},
    "WidgetAttrs": [],
    "WidgetActions": []
  }
],
```

用于定义各界面包含的控件类型，及控件本身的属性等信息。

5.4. .\cmds\shell 脚本介绍

M2_Data (D:) > 14.Git > Release > cmds > shell			
名称	修改日期	类型	大小
collect_binlog.sh	2020/10/17 22:08	SH 文件	2 KB
collect_fast_logs.sh	2020/10/17 21:59	SH 文件	1 KB
common_function.sh	2020/10/17 21:14	SH 文件	1 KB
crontab_management.sh	2020/10/19 19:29	SH 文件	1 KB
fast_commands.sh	2020/10/17 21:58	SH 文件	1 KB
fast_download.sh	2020/10/17 21:34	SH 文件	1 KB
fast_upload.sh	2020/10/17 22:00	SH 文件	1 KB
function_util.py	2020/8/5 21:45	Python File	2 KB
get_process_state.sh	2020/10/19 19:29	SH 文件	1 KB
home_page.sh	2020/10/18 15:21	SH 文件	2 KB
inner_function.sh	2020/10/21 20:00	SH 文件	3 KB
install_version.sh	2020/10/21 20:00	SH 文件	1 KB
rollback_version.sh	2020/10/17 22:01	SH 文件	2 KB
set_timezone.py	2020/8/5 21:45	Python File	16 KB
timezone_operate.sh	2020/10/17 22:01	SH 文件	1 KB

脚本名	说明
common_function.sh	shell 脚本公共函数脚本，不可篡改，且自定义功能脚本必须 source 此脚本。
inner_function.sh	工具内部调用接口脚本，不可篡改。
.sh/.py	自定义的服务器端功能实现脚本。

6. 功能扩展指导

6.1. 新增一个界面，你需要准备：

- 1) 一个 18x18 像素的 png 图标；
- 2) 编写至少一个功能脚本；
- 3) 在 dependance.json 配置文件的三个区域中分别增加相应的配置；

6.2. 界面功能流程

工具目前支持以下 4 类任务事件：

- 1) 界面进入后自动执行脚本预先加载状态数据的功能配置；
- 2) 点击执行按钮后触发服务器脚本任务执行；
- 3) 点击执行按钮后触发可视化图标的生成和显示；
- 4) 自定义界面和自定义事件；

1. 进入界面自动触发执行：功能脚本需要实现‘ENTER’参数接口，并输出结果（标准输出）。
2. 点击‘执行’按钮后，工具会对界面中的控件进行一一校验和获取输入值，并做相应的转换后，传递给服务器脚本并执行，获取输出信息和回显。
3. 点击‘执行’按钮后，工具获取必要输入后，将从本地.\download\<IP>\DATA 目录获取有效数据文件，并将数据可视化展示。

6.3. 定义新图标（Images 区）：

```
"Images": {  
  "ICO": ".\\image\\item.ico",  
  "GUIDE": ".\\image\\guide.png",  
  "ABOUT": ".\\image\\about.png",  
  "LGN_HEAD": ".\\image\\login_head.jpg",  
  "LGN_EYE": ".\\image\\login_eye.png",  
  "TV_NEWPAGE": ".\\image\\treeview_newpage.png"  
}
```

6.4. 定义界面入口（Tree 区）：

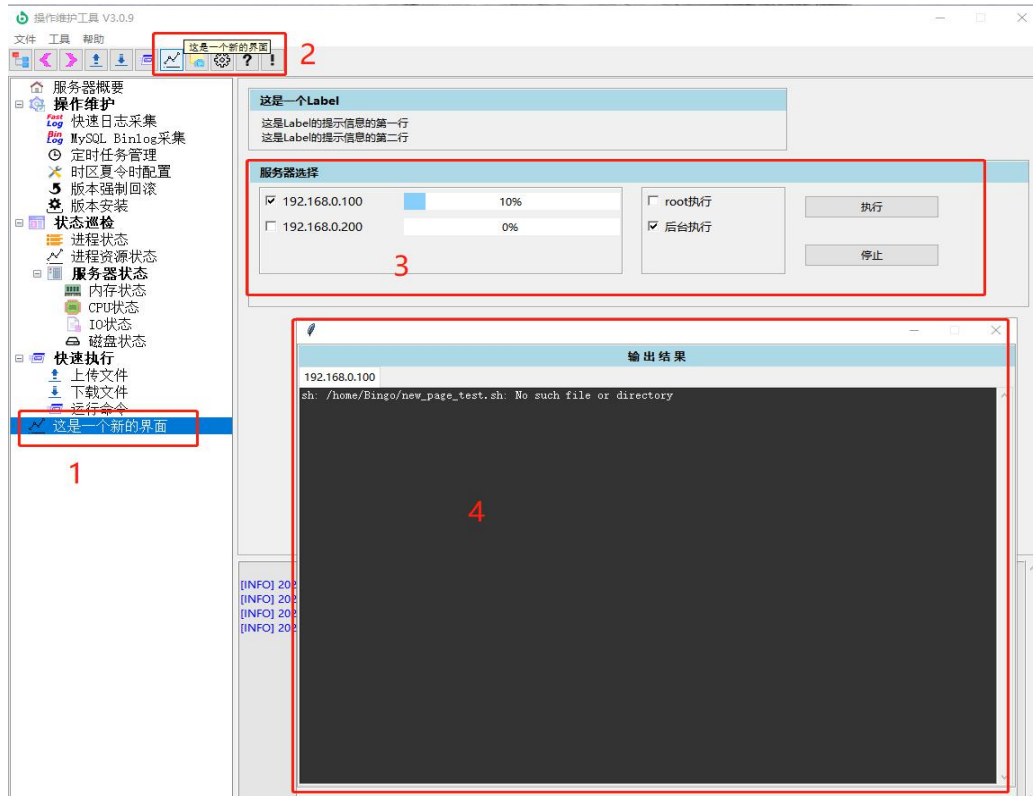
在对应同一类型操作的 SubTree 子树节点中添加一个以下界面实例：

```
{  
  "Text": "这是一个新的界面",  
  "Image": "TV_NEWPAGE",  
  "Page": {  
    "Widgets": "ANewTextPageWidgets",  
    "Shell": "new_page_test.sh",  
    "Attrs": ["ToolBarMember", "OperateButtons", "ResultWindow", "PlotWindow"  
  ]  
},  
}
```

```
"SubTree": []
}
```

字段	字段类型	说明
Text	字符串	目录导航栏显示的界面文字，如果本界面添加到了快捷工具栏中，在对应的图标上也会提示该文字信息。
Image	字符串	对应 Images 区中的图标索引，该值需全局唯一，且图标不能复用；因为在快捷工具栏中是以图标来区分界面，且以图标索引来获取界面信息。
Page	对象； 其中： Widget: 字符串 Shell: 字符串 Attrs: 数组	定义该新增界面的内部属性： Widgets: 对应__ThisIsPageWidgets__区中的 key，用于匹配定义的详细界面。 Shell: 对应.\\cmds\\shell 中的服务器功能脚本名。 Attrs: 界面属性；目前包含： ToolBarMember: 将本界面添加到快捷工具栏中，可快速点击跳转到界面。 OperateButtons: 界面类型为选择执行型；需要显示 IP、进度条以及执行按钮等。 ResultWindow: 界面为独立窗体显示执行结果型；需要一个独立窗体回显任务执行的结果打印信息。 PlotWindow: 界面为可视化视图界面型；工具将从服务器下载的数据文件中提取数据后进行可视化展示。
SubTree	数组	子界面设置区域，如果有子界面时，Page 字段需要设置为“NA”，因为父界面节点默认无界面显示。

效果图 1：添加新子界面



序号	说明
1	新加的目录导航栏；Tree 区新加实例效果。
2	新加的快捷工具栏。Page-Attrs 中 ToolBarMember 效果。
3	新加界面中 Page-Attrs 中的 OperateButtons 效果。
4	新加界面中 Page-Attrs 中的 ResultWindow 效果。

效果图 2：添加新父节点和子界面

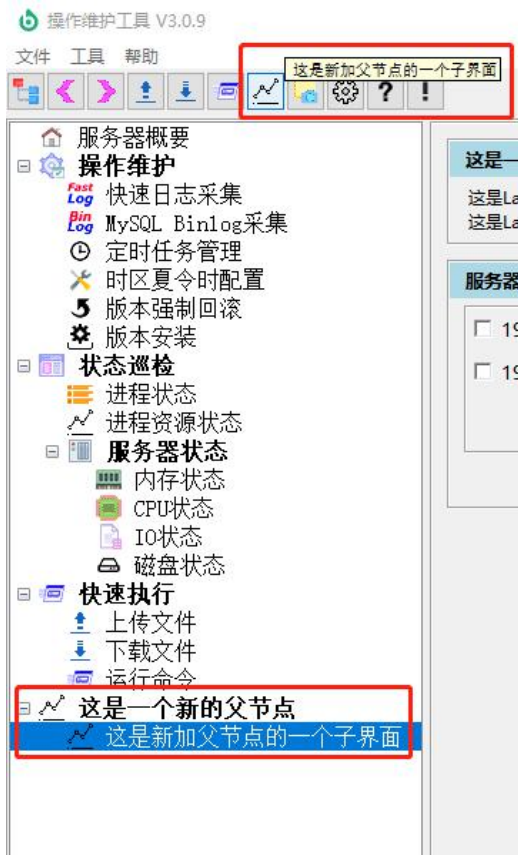
```
{
  "Text": "这是一个新的父节点",
  "Image": "TV_NEWPAGE",
  "Page": "NA",
  "SubTree": [
    {
      "Text": "这是新加父节点的一个子界面",
      "Image": "TV_NEWPAGE",
      "Page": {
        "Widgets": "ANewTextPageWidgets",
        "Shell": "new_page_test.sh",
        "Attrs": ["ToolBarMember", "OperateButtons", "ResultWindow"]
      },
      "SubTree": []
    }
  ]
}
```



```

]
},

```



6.5. 定义界面控件（__ThisIsPageWidgets__区）

```

"ANewTextPageWidgets": [
    {
        "WidgetType": "Label",
        "WidgetTips": ["这是一个 Label"],
        "WidgetValues": ["这是 Label 的提示信息的第一行", "这是 Label 的提示信息的第二行"],
        "WidgetParams": {"Size": [600, 50, 600, 50]},
        "WidgetAttrs": [],
        "WidgetActions": []
    },
    {
        "WidgetType": "Entry",
        "WidgetTips": ["这是一个 Entry"],
        "WidgetValues": ["这是 Entry 中的默认值"],
        "WidgetParams": {"Size": [600, 30, 20, 1]},
        "WidgetAttrs": [],
        "WidgetActions": []
    },
]

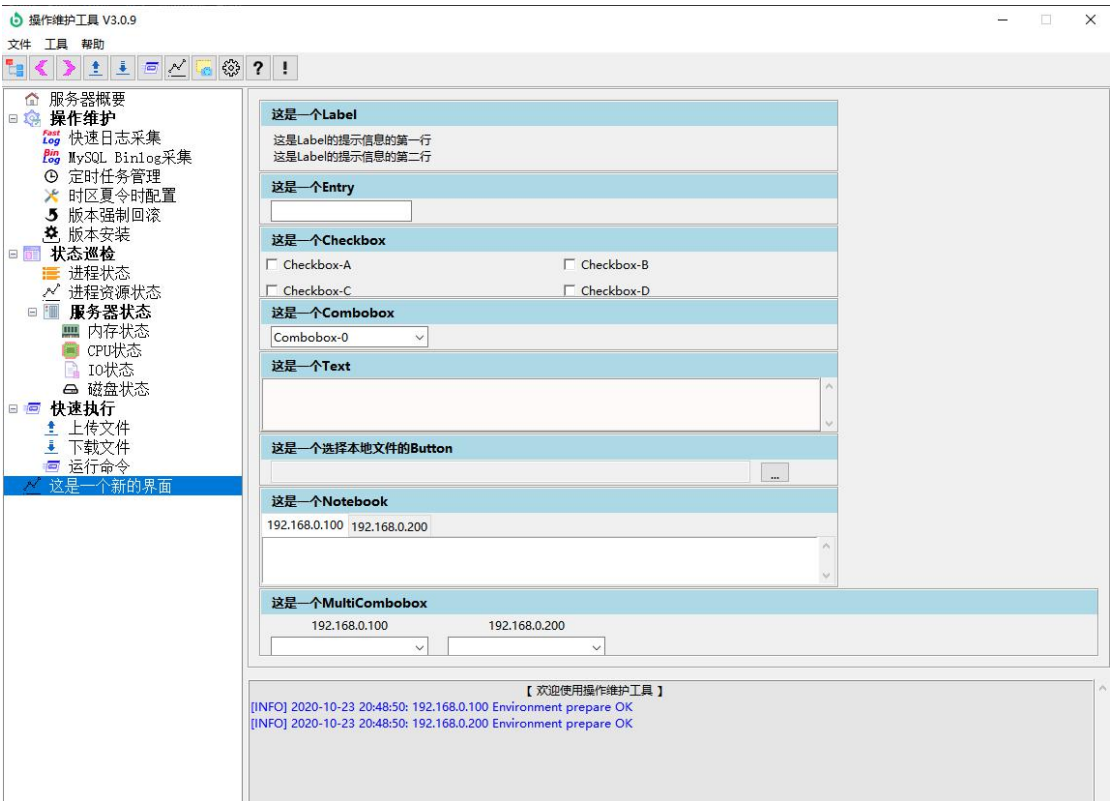
```

```
"WidgetType": "Checkbox",
"WidgetTips": ["这是一个 Checkbox"],
"WidgetValues": ["Checkbox-A", "Checkbox-B", "Checkbox-C", "Checkbox-D"],
"WidgetParams": {"Size": [600, 50, 40, 1]},
"WidgetAttrs": [],
"WidgetActions": []
},
{
  "WidgetType": "Combobox",
  "WidgetTips": ["这是一个 Combobox"],
  "WidgetValues": ["Combobox-0", "Combobox-1", "Combobox-2"],
  "WidgetParams": {"Size": [600, 30, 20, 1]},
  "WidgetAttrs": [],
  "WidgetActions": []
},
{
  "WidgetType": "Text",
  "WidgetTips": ["这是一个 Text"],
  "WidgetValues": [],
  "WidgetParams": {"Size": [600, 60, 600, 4]},
  "WidgetAttrs": [],
  "WidgetActions": []
},
{
  "WidgetType": "Button",
  "WidgetTips": ["这是一个选择本地文件的 Button"],
  "WidgetValues": ["ChooseFile", "*"],
  "WidgetParams": {"Size": [600, 30, 10, 1]},
  "WidgetAttrs": [],
  "WidgetActions": []
},
{
  "WidgetType": "Notebook",
  "WidgetTips": ["这是一个 Notebook"],
  "WidgetValues": [],
  "WidgetParams": {"Size": [600, 80, 600, 80]},
  "WidgetAttrs": [],
  "WidgetActions": []
},
{
  "WidgetType": "MultiCombobox",
  "WidgetTips": ["这是一个 MultiCombobox"],
  "WidgetValues": ["", "MultiCombobox-A", "MultiCombobox-B",
"MultiCombobox-C"],
```

```
"WidgetParams": {"Size": [900, 50, 20, 1]},
"WidgetAttrs": [],
"WidgetActions": []
}
```

字段	字段类型	说明
WidgetType	字符串	控件类型； 目前支持的控件有：'Label'，'Checkbox'，'Combobox'，'Entry'，'Text'，'Button'，'Notebook'，'MultiCombobox'。 当需要自己实现一些复杂的界面时，该项值为 Self，同时需要自己实现界面类，参考 TimezonePage（只限工具研发人员）
WidgetTips	数组	控件提示信息；默认一行。
WidgetValues	数组	控件的值； 如果是 Label 或 Entry，则每个元素为一行，显示该值； 如果是 Checkbox，则每个元素为一个复选项； 如果是 Combobox，则每个元素为一个单选项； 如果是 Text 或 Notebook，则元素无意义，不处理； 如果是 Button，第一个元素为 Button 类型，第二个元素为类型参数；（目前只支持 ChooseFile 类型） 当控件中的值不是固定内容时，例如需要显示服务器的 IP 和网卡时，可使用内部变量：__IP__ 和 __NETCARD__，分别代表 IP 和网卡数组，在工具登录服务器后自动循环更新这两个内部变量的值，在进入界面后自动替换该内部变量为具体的值。
WidgetParams	对象	控件布局参数； Size : 控件的画布大小和控件大小，其元素依次为画布的宽，高，控件的宽，高。其中控件高度的单位在控件是 Notebook、Label 时为像素，控件为 Entry、Checkbox、Combobox、Text、Botton、MultiCombobox 时为文本行数。
WidgetAttrs	数组	控件的属性； ShowEnterResult : 表明进入界面后自动执行脚本（此时参数为 'ENTER'）并把结果布局到该控件上。目前只支持 Notebook/Label 布局自动执行脚本得到的结果。 CanBeNull : 表明该控件的输入框内可以不输入内容便可开始执行任务脚本，服务器选择栏点击“执行”后开始校验。
WidgetActions	数组	服务器选择栏点击“执行”后开始执行任务脚本之前，需要处理的动作； UploadFile : 表明执行前，需要先上传该控件输入框中输入的文件到服务器上。

效果图：



6.6. 编写功能脚本

脚本需要支持以下几个条件：

- 1) 固定参数“ENTER”：用于支持界面进入后自动执行脚本所走的分支，通常用于获取某项功能当前的状态；
- 2) 控件输入中的参数：按照布局的控件顺序一个控件一个参数，各控件参数说明如下：

控件名	参数值说明
Checkbox	复选框选中的为 1，为选中的为 0，一个 checkbox 控件的参数组合成类似 1@0@0@1 这样的一个参数，需要脚本自行对应解析。
Combobox	参数值为所在 WidgetValues 数组中的序号，需要脚本自行对应。
Entry	参数值为输入的字符串，但当 WidgetActions 含有 UploadFile

	动作时，参数为输入的文件名，不含 windows 的路径；
Text	由于该控件输入内容为多行文本，无法当成一个参数传递给脚本，工具默认把该控件输入内容写入一个文件，并自动上传该文件到服务器，并把该文件的文件名传递给脚本，上传的服务器路径为 g_upload_dir。
Button	ChooseFile 类型的 Button 实际上参数为 Entry 控件中的本地文件路径，参数值同样为不含路径的文件名。
MultiCombobox	参数值为该控件选中的字符串。

7. 技巧及其他

脚本公共函数说明：

report_info：脚本执行的进度和进度信息，用于与工具内部交互，回显进度条；当进度为 100 时，第二个参数为需要下载的文件路径，工具会自动下载该文件，如无需下载，则设置为‘NULL’；

report_err：脚本执行失败后的进度和错误信息，该函数将立即退出脚本，工具则回显失败和失败信息；

compress：统一的打包压缩函数；