

# aardio 范例: aria2 下载 —— 如果 BT 下载没速度, 先找个热门种子下载就可以了

```
//BT 下载 / WinForm
import win.ui;
/*DSG{*/
var winform = win.form(text="aria2 下载 —— 如果 BT 下载没速度, 先找个热门种子下载就可以了";right=921;bottom=537;bgcolor=16777215;)
winform.add(
    btnAdd={cls="button";text="调用 aria2 下载文件";left=605;top=332;right=747;bottom=368;db=1;dr=1;z=3;};
    listview={cls="listview";left=12;top=7;right=912;bottom=318;bgcolor=16777215;db=1;dl=1;dr=1;dt=1;edge=1;fullRow=1;z=1;};
    txtData={cls="edit";left=32;top=336;right=585;bottom=367;align="right";db=1;dl=1;dr=1;edge=1;z=2;};
    txtMessage={cls="edit";left=8;top=380;right=909;bottom=523;db=1;dl=1;dr=1;edge=1;hscroll=1;multiline=1;vscroll=1;z=4;};
)
/*}*/

//启动 aria2 服务器
import process.aria2;
var aria2 = process.aria2();

//注意看错误信息。
aria2.onError = function(errMsg, rpcErr){
    winform.txtMessage.print(errMsg);
}

//启动下载服务器
aria2.startServer( maxConcurrentDownloads = 10 );

winform.listview.insertColumn("文件",250);
winform.listview.insertColumn("速度",80);
winform.listview.insertColumn("已下载",120);
winform.listview.insertColumn("状态",-1);
winform.listview.insertColumn("连接数",120);

var aria2Ui = {
    downloadData = {};

    getGid = function(item){
        return table.find(owner.downloadData,item);
    }

    addTaskInfo = function(gid,url,status){
        if(!gid){
            var name = bencoding.magnet.getName(url);
            winform.listview.addItem({
                name,null,null,"出错了: "+ (status:"未知错误");
            });
        }
        else {
            var taskName = aria2.taskName(gid)

            var item = owner.downloadData[gid];
            if(!item){
                var item = winform.listview.addItem({
                    taskName,null,null,status || "正在下载"
                });
                owner.downloadData[gid] = item;
            }
            else {
                winform.listview.setItemText(taskName,item,1);
                winform.listview.setItemText(status : "正在下载",item,4);
            }
        }
    }

    updateStatus = function(gid,status){
        var item = owner.downloadData[gid]
        if(!item){
            owner.addTaskInfo(gid);
            item = owner.downloadData[gid]
        }

        winform.listview.setItemText(status,item,4);
    }

    updateProgress = function(gid,progress){
        var item = owner.downloadData[gid]
        if(!item){
            owner.addTaskInfo(gid);
            item = owner.downloadData[gid]
        }

        if(progress){
```

```

        winform.listView.setItemText(progress.connections,item,5);
        winform.listView.setItemText(math.size64(progress.downloadSpeed,item).format() + "/s",item,2);
        winform.listView.setItemText(math.size64(progress.completedLength).format()
            + "/" + math.size64(progress.totalLength).format(),item,3);
    }
    else {
        winform.listView.setItemText("",item,2);
        winform.listView.setItemText("",item,5);
        var progress = aria2.tellStatus(gid,"totalLength");
        winform.listView.setItemText( math.size64(progress.totalLength).format(),item,3);
    }
}

//监听 aria2 事件
aria2.onDownloadStart = function(task){
    aria2Ui.addTaskInfo(task.gid,,"正在下载")
}

aria2.onDownloadPause = function(task){
    aria2Ui.updateStatus(task.gid,"暂停下载");
}

aria2.onDownloadStop = function(task){
    aria2Ui.updateStatus(task.gid,"已停止");
}

aria2.onDownloadComplete = function(task){
    aria2Ui.updateStatus(task.gid,"已完成");
    aria2Ui.updateProgress(task.gid);
}

aria2.onDownloadError = function(task){
    var errMsg = aria2.taskErrorMessage(param.gid);
    aria2Ui.updateStatus(task.gid,errMsg);
}

//调用 aria2 下载
winform.btnAdd.oncommand = function(id,event){
    var url = winform.txtData.text;
    if(!#url){
        winform.msgboxErr("请输入下载地址或种子文件路径")
    }

    aria2.taskAdd(url,function(gid,err){

        if(err){
            aria2Ui.addTaskInfo(null,url,err[["message"]]);
        }
        else {
            aria2Ui.addTaskInfo(gid,url,"添加成功" );
        }
    });
}

winform.txtData.setCueBannerText("请输入下载地址或种子文件路径");

//获取进度
//https://aria2.github.io/manual/en/html/aria2c.html#aria2.tellStatus
updateDownloadStatus = function(){

    aria2.tellActive(function(result,err){
        if(result) {
            for(k,v in result){
                aria2Ui.updateProgress(v.gid,v);
            }
        }
    }, "gid", "status", "connections", "downloadSpeed", "totalLength", "completedLength");

    aria2.tellWaiting(0,10,function(result,err){
        if(result) {
            for(k,v in result){
                aria2Ui.updateProgress(v.gid,v);
            }
        }
    }, "gid", "status", "connections", "downloadSpeed", "totalLength", "completedLength");
}

//启动就绪执行
aria2.ready(
    function(){

        //查看默认配置。
        //winform.txtMessage.print(aria2.getGlobalOption());

        //创建定时器，更新下载进度
        winform.setInterval(updateDownloadStatus,1500);
    }
)

```

```

//下载任务右键管理菜单
import win.ui.menu;
winform.listView.onRightClick = function(item,subItem,nmListView){

    //创建弹出菜单
    var gid = aria2Ui.getGid(nmListView.iItem);
    if(!gid) return;

    var statusInfo = aria2.tellStatus(gid,"status","belongsTo");
    if(!statusInfo) return;

    var popmenu = win.ui.popmenu(winform);
    popmenu.add('移除',function(id){
        aria2.taskRemove(gid);
        winform.listView.delItem(nmListView.iItem);
    })

    if(statusInfo.status == "active"){
        popmenu.add('暂停',function(id){
            var gid = aria2Ui.getGid(nmListView.iItem);
            aria2.taskPause(gid);
        })
    }
    if(statusInfo.status == "paused"){
        popmenu.add('开始下载',function(id){
            var gid = aria2Ui.getGid(nmListView.iItem);
            aria2.taskUnpause(gid);
        })
    }

    var path = aria2.taskFilePath(gid)
    if(path){
        popmenu.add('浏览文件',function(id){
            process.exploreSelect(path)
        })
    }

    popmenu.add('复制链接',function(id){
        var url = aria2.taskUrl(gid);
        if(url){
            import win.clip;
            win.clip.write(url);
        }
    })

    popmenu.popup();
}

winform.listView.enableDoubleBuffering();

winform.show();
win.loopMessage();

```

[Markdown 格式](#)