

机器人系统使用说明

📖 目录

- [系统概述](#)
- [配置说明](#)
- [脚本编写](#)
- [消息类型](#)
- [执行类型详解](#)
- [实战示例](#)
- [故障排查](#)

🎯 系统概述

什么是机器人系统？

机器人系统（定时任务系统）是游戏服务器的**自动化脚本执行引擎**，用于：

- 🕒 定时发送系统公告
- 🎮 自动触发游戏活动
- 📢 周期性广播信息
- ⚙️ 定时执行游戏逻辑

系统架构



核心文件位置

文件	路径	作用
配置文件	Database/System/其他配置/机器人配置.csv	定义任务执行时间表
脚本文件	Database/System/Envir/00-QFunction.txt	实现任务具体逻辑
核心代码		
主程集成		

与假人系统的区别

特性	机器人系统	假人系统
用途	定时任务、系统公告	AI虚拟玩家
账号	JokerBot251012	Bot_*
角色	假人（固定）	多个假人角色
控制	CSV配置 + 脚本	HTTP API控制
触发	时间触发	外部API触发

配置说明

CSV配置文件格式

文件路径: Database/System/其他配置/机器人配置.csv

```
StartTime,Interval,Type,Enable,Count,Key
2025/10/12,00:00:05,ROSEC,1,999999999,@_5秒测试
2025/10/12,19:00:00,RUNONDAY,1,999999999,@_怪物攻城活动开始
2025/10/12,6:20:00:00,RUNONWEEK,1,999999999,@_自动攻城
```

字段说明

1. StartTime - 开始日期

- 格式: YYYY/MM/DD
- 说明: 任务首次执行的起始日期
- 示例: 2025/10/12

2. Interval - 时间间隔

- **格式:** 根据Type类型不同而不同
 - 日时间: HH:MM:SS (如: 19:00:00)
 - 周时间: D:HH:MM:SS (如: 6:20:00:00 表示周六20点)
 - 秒间隔: 00:00:05 (表示每5秒)
- **说明:** 任务执行的具体时间或间隔

3. Type - 执行类型

执行类型决定任务如何重复执行，详见[执行类型详解](#)

类型	说明	示例
RUNONDAY	每天指定时间执行	每天19点
RUNONWEEK	每周指定时间执行	每周六20点
RUNONHOUR	每小时指定分秒执行	每小时30分
RUNONMIN	每分钟指定秒执行	每分钟30秒
RUNONSEC	每秒执行	每秒
ROHOUR	按小时间隔执行	每隔2小时
ROMIN	按分钟间隔执行	每隔5分钟
ROSEC	按秒间隔执行	每隔10秒
RODAY	按天间隔执行	每隔1天

4. Enable - 启用状态

- **取值:** 1 = 启用, 0 = 禁用
- **说明:** 控制任务是否执行

5. Count - 执行次数

- **取值:** 数字
 - 1 = 只执行一次
 - 999999999 = 无限次执行
- **说明:** 任务最多执行次数

6. Key - 脚本段名称

- **格式:** @_脚本名称
 - **说明:** 对应 00-QFunction.txt 中的脚本段
 - **注意:** 必须以 @_ 开头
-

脚本编写

脚本文件位置

文件路径: Database/System/Envir/00-QFunction.txt

脚本基本格式

[@_脚本名称]

#IF

条件判断语句

#ACT

执行动作语句

Break

脚本示例

示例1：简单广播公告

[@_5秒测试]

#IF

#ACT

GlobalMessage "【定时任务】5秒测试执行中..." TRUE

Break

示例2：活动开始公告

[@_怪物攻城活动开始]

#IF

#ACT

GlobalMessage "【系统公告】怪物攻城活动现在开始！" TRUE

GlobalMessage "活动时间：30分钟，请勇士们速速前往比奇城参加！" TRUE

Break

示例3：多条件执行

[@_整点福利]

#IF

EQUAL G100 1

#ACT

GlobalMessage "【福利公告】整点福利已发放到所有在线玩家！" TRUE

Break

常用脚本命令

消息发送命令

- GlobalMessage - 全服广播（所有玩家）
- TopMessage - 屏幕顶部消息（单个玩家）
- LocalMessage - 聊天消息（单个玩家）

详见：[消息类型](#)

其他常用命令

- MONGEN - 刷怪
- NPCGEN - 生成NPC
- MOV - 设置变量
- CALC - 数学运算
- CALL - 调用其他脚本

消息类型

三种消息对比

消息类型	接收范围	显示位置	滚动播报	应用场景
GlobalMessage	全服所有玩家	聊天窗口/屏幕顶部	✔ 支持	系统公告、活动通知
TopMessage	单个玩家	屏幕顶部	✔ 支持	重要提示、任务完成
LocalMessage	单个玩家	聊天窗口	✘ 不支持	个人消息、任务进度

1. GlobalMessage - 全服广播

语法:

```
GlobalMessage "消息内容" 是否滚动播报
```

参数说明:

- 消息内容: 字符串，要广播的内容
- 是否滚动播报: TRUE = 滚动播放，FALSE = 静态显示

实现位置: `GameServer/地图类/NPC/NPCSegment.cs:6246`

代码实现:

```
网络服务网关.发送公告(param[0], result148);
```

使用示例:

```
[@_活动通知]
#IF
#ACT
GlobalMessage "【系统公告】Boss即将刷新！" TRUE
Break
```

显示效果:

- ☒ 所有在线玩家都会收到
- ☒ 消息从右向左滚动 (如果设置TRUE)
- ☒ 记录到系统日志

2. TopMessage - 屏幕顶部消息

语法:

```
TopMessage "消息内容" 是否滚动播报
```

参数说明:

- **消息内容:** 字符串, 要显示的内容
- **是否滚动播报:** `TRUE` = 滚动播放, `FALSE` = 静态显示

实现位置: `GameServer/地图类/NPC/NPCSegment.cs:5577`

代码实现:

```
player.发送顶部公告(param[0], 全服通知: false, result61);
```

使用示例:

```
[@_任务完成]
#IF
#ACT
TopMessage "恭喜完成任务！" FALSE
Break
```

显示效果:

- ☒ 仅当前玩家看到
- ☒ 显示在屏幕顶部
- ☒ 更醒目, 适合重要提示

3. LocalMessage - 聊天窗口消息

语法:

```
LocalMessage "消息内容" FALSE
```




参数说明:

- **消息内容:** 字符串, 要发送的内容
- **第二个参数:** 通常设为 `FALSE`

使用示例:

```
[ @_获得奖励 ]
#IF
#ACT
LocalMessage "你获得了100金币奖励!" FALSE
Break
```

显示效果:

-  仅当前玩家看到
-  显示在聊天窗口
-  带有玩家等级信息

滚动播报参数说明

TRUE - 滚动播报模式:

- 消息从屏幕右侧滚动到左侧 (跑马灯效果)
- 适合活动公告、系统通知
- 更吸引玩家注意

FALSE - 静态显示模式:

- 消息直接显示, 不滚动
- 适合任务提示、个人消息
- 显示时间较短

代码实现 (`网络服务网关.cs:824`):

```
binarywriter.write((byte)(滚动播报 ? 2 : 3));
// 2 = 滚动播报模式
// 3 = 静态显示模式
```

消息类型选择建议

使用场景	推荐类型	滚动播报
系统维护公告	GlobalMessage	TRUE
活动开始/结束	GlobalMessage	TRUE
世界Boss刷新	GlobalMessage	TRUE
玩家任务完成	TopMessage	FALSE
玩家升级提示	TopMessage	FALSE
物品掉落提示	LocalMessage	FALSE
经验获得提示	LocalMessage	FALSE



执行类型详解

执行类型分类

机器人系统支持**两大类**执行模式：

1. **间隔型**（RO系列）：按固定时间间隔重复执行
2. **定时型**（RUNON系列）：在指定时间点执行

间隔型执行类型

ROSEC - 按秒间隔执行

配置格式:

```
2025/10/12,00:00:05,ROSEC,1,999999999,@_5秒测试
```

说明:

- 间隔值: 00:00:05 = 每5秒执行一次
- 适用场景: 高频监控、实时刷新

实际示例:

```
# 每10秒广播一次公告
2025/10/12,00:00:10,ROSEC,1,999999999,@_实时监控
```

ROMIN - 按分钟间隔执行

配置格式:

```
2025/10/12,00:05:00,ROMIN,1,999999999,@_5分钟任务
```

说明:

- 间隔值: 00:05:00 = 每5分钟执行一次
- 适用场景: 定期检查、周期性任务

实际示例:

```
# 每10分钟刷新Boss
2025/10/12,00:10:00,ROMIN,1,999999999,@_刷新Boss
```

ROHOUR - 按小时间隔执行

配置格式:

```
2025/10/12,02:00:00,ROHOUR,1,999999999,@_2小时任务
```


说明:

- 间隔值: 02:00:00 = 每2小时执行一次
- 适用场景: 长周期活动、定时重置

实际示例:

```
# 每3小时重置副本
2025/10/12,03:00:00,ROHOUR,1,999999999,@_重置副本
```

RODAY - 按天间隔执行

配置格式:

```
2025/10/12,1:00:00:00,RODAY,1,999999999,@_每日任务
```

说明:

- 间隔值: 1:00:00:00 = 每天执行一次
- 适用场景: 日常重置、每日活动

定时型执行类型

RUNONDAY - 每天固定时间执行

配置格式:

```
2025/10/12,19:00:00,RUNONDAY,1,999999999,@_怪物攻城活动开始
```

说明:

- 间隔值: 19:00:00 = 每天19点整执行
- 适用场景: 每日活动、定点维护

实际示例:

```
# 每天20:30开始石墓夺宝
2025/10/12,20:30:00,RUNONDAY,1,999999999,@_石墓夺宝活动开始

# 每天21:30结束活动
2025/10/12,21:30:00,RUNONDAY,1,999999999,@_石墓夺宝活动结束
```

RUNONWEEK - 每周固定时间执行

配置格式:

```
2025/10/12,6:20:00:00,RUNONWEEK,1,999999999,@_自动攻城
```

说明:

- 间隔值: 6:20:00:00

- 6 = 星期六 (1=周一, 2=周二, ..., 7=周日)
- 20:00:00 = 20点整
- 适用场景: 周末活动、攻城战

星期对照表:

数字	星期
1	周一
2	周二
3	周三
4	周四
5	周五
6	周六
7	周日

实际示例:

```
# 每周六20点开始攻城战
2025/10/12,6:20:00:00,RUNONWEEK,1,999999999,@_自动攻城

# 每周日15点开始公会战
2025/10/12,7:15:00:00,RUNONWEEK,1,999999999,@_公会战开始
```

RUNONHOUR - 每小时固定分秒执行

配置格式:

```
2025/10/12,00:30:00,RUNONHOUR,1,999999999,@_整点半公告
```

说明:

- 间隔值: 00:30:00 = 每小时的30分执行
- 适用场景: 整点公告、定时提醒

实际示例:

```
# 每小时的45分发送提醒
2025/10/12,00:45:00,RUNONHOUR,1,999999999,@_活动提醒
```

RUNONMIN - 每分钟固定秒执行

配置格式:

```
2025/10/12,00:00:30,RUNONMIN,1,999999999,@_每分钟30秒
```

说明:

- 间隔值: 00:00:30 = 每分钟的30秒执行
- 适用场景: 高频检查、实时监控

RUNONSEC - 每秒执行

配置格式:

```
2025/10/12,00:00:00,RUNONSEC,1,999999999,@_每秒任务
```

说明:

- 极高频率执行
- 适用场景: 紧急测试、实时数据

⚠ 警告: 不建议在生产环境使用, 会消耗大量性能

执行类型选择指南

使用场景	推荐类型	配置示例
每天固定时间活动	RUNONDAY	19:00:00
每周固定时间活动	RUNONWEEK	6:20:00:00
每小时整点公告	RUNONHOUR	00:00:00
每5分钟刷新	ROMIN	00:05:00
每10秒检查	ROSEC	00:00:10
每2小时重置	ROHOUR	02:00:00

🎮 实战示例

示例1: 每日活动系统

配置文件 (机器人配置.csv)

```
StartTime,Interval,Type,Enable,Count,Key
2025/10/12,19:00:00,RUNONDAY,1,999999999,@_怪物攻城活动开始
2025/10/12,19:30:00,RUNONDAY,1,999999999,@_怪物攻城活动结束
2025/10/12,20:00:00,RUNONDAY,1,999999999,@_古墓夺宝活动开始
2025/10/12,21:00:00,RUNONDAY,1,999999999,@_古墓夺宝活动结束
```

脚本文件 (00-QFunction.txt)

```
[@_怪物攻城活动开始]
#IF
#ACT
GlobalMessage "【活动公告】怪物攻城活动现在开始!" TRUE
GlobalMessage "活动时间: 30分钟, 请勇士们前往比奇城参加!" TRUE
MOV G100 1
Break

[@_怪物攻城活动结束]
#IF
#ACT
GlobalMessage "【活动公告】怪物攻城活动已结束, 感谢各位勇士的参与!" TRUE
```

```
MOV G100 0
Break

[@_石墓夺宝活动开始]
#IF
#ACT
GlobalMessage "【活动公告】石墓夺宝活动现在开始！" TRUE
GlobalMessage "大量宝箱已在石墓阵刷新，速来寻宝！" TRUE
Break

[@_石墓夺宝活动结束]
#IF
#ACT
GlobalMessage "【活动公告】石墓夺宝活动已结束！" TRUE
Break
```

示例2：定时刷新系统

配置文件

```
StartTime,Interval,Type,Enable,Count,Key
2025/10/12,00:10:00,ROMIN,1,999999999,@_刷新世界Boss
2025/10/12,02:00:00,ROHOUR,1,999999999,@_重置副本
```

脚本文件

```
[@_刷新世界Boss]
#IF
#ACT
MONGEN 沃玛教主 沃玛寺庙三层 100 100 1
GlobalMessage "【Boss刷新】沃玛教主已在沃玛寺庙三层刷新！" TRUE
Break

[@_重置副本]
#IF
#ACT
GlobalMessage "【系统公告】所有副本已重置，祝各位勇士挑战顺利！" TRUE
Break
```

示例3：每周攻城战

配置文件

```
StartTime,Interval,Type,Enable,Count,Key
2025/10/12,6:20:00:00,RUNONWEEK,1,999999999,@_攻城战开始
2025/10/12,6:22:00:00,RUNONWEEK,1,999999999,@_攻城战结束
```

脚本文件

```
[@_攻城战开始]
#IF
#ACT
GlobalMessage "_____ " TRUE
GlobalMessage "【攻城战】沙巴克攻城战正式开始！ " TRUE
GlobalMessage "【攻城战】持续时间：2小时" TRUE
GlobalMessage "【攻城战】胜利行会将获得丰厚奖励！ " TRUE
GlobalMessage "_____ " TRUE
Break

[@_攻城战结束]
#IF
#ACT
GlobalMessage "【攻城战】沙巴克攻城战已结束！ " TRUE
GlobalMessage "【攻城战】恭喜获胜行会！ " TRUE
Break
```

示例4：整点福利系统

配置文件

```
StartTime,Interval,Type,Enable,Count,Key
2025/10/12,00:00:00,RUNONHOUR,1,999999999,@_整点福利
```

脚本文件

```
[@_整点福利]
#IF
#ACT
GlobalMessage "【整点福利】整点福利已发放！在线玩家可领取经验宝箱！ " TRUE
Break
```

示例5：测试用5秒循环

配置文件

```
StartTime,Interval,Type,Enable,Count,Key
2025/10/12,00:00:05,ROSEC,1,999999999,@_5秒测试
```

脚本文件

```
[@_5秒测试]
#IF
#ACT
GlobalMessage "【定时任务】5秒测试执行中..." TRUE
Break
```

实际效果:

```
[2025年10月12日 0:59:55]: 【定时任务】5秒测试执行中...
[2025年10月12日 1:00:00]: 【定时任务】5秒测试执行中...
[2025年10月12日 1:00:05]: 【定时任务】5秒测试执行中...
...
```

🔧 故障排查

常见问题

1. 机器人任务不执行

症状: CSV配置正确，但任务没有触发

原因: 机器人系统未初始化

解决方案: 检查 `GameServer/杂项/主程.cs` 第980-982行

确保以下代码未被注释:

```
// 机器人系统初始化 - 用于定时任务调度（活动公告等）
机器人.载入数据(); // 加载机器人配置文件
机器人.初始化(); // 创建机器人专用角色（账号：JokerBot251012）
```

验证方法: 查看系统日志

```
[2025年10月12日 0:34:16]: 机器人.模板已经加载, 数量: 9
[2025年10月12日 0:34:16]: [CallDefaultNPC调试] 玩家假人调用DefaultNPC, 类型:Login, 延迟:False
```

2. 时间配置错误

症状: 任务执行时间不对

常见错误:

```
# ❌ 错误: 缺少前导0
2025/10/12,9:0:0,RUNONDAY,1,999999999,@_测试

# ✅ 正确: 标准格式
2025/10/12,09:00:00,RUNONDAY,1,999999999,@_测试
```

时间格式规范:

- 小时: `00-23` (两位数)
- 分钟: `00-59` (两位数)
- 秒数: `00-59` (两位数)

3. 脚本段名称不匹配

症状: 日志显示任务触发，但没有执行效果

常见错误:

```
# CSV配置
Key: @_怪物攻城活动开始

# 脚本文件
[@怪物攻城活动开始] ❌ 缺少下划线
```

正确写法:

```
# CSV中的Key必须与脚本段名称完全一致
CSV: @_怪物攻城活动开始
脚本: [@_怪物攻城活动开始]
```

4. Count参数设置错误

症状: 任务只执行一次就停止

原因: Count设置为1

```
# ❌ 只执行一次
2025/10/12,19:00:00,RUNONDAY,1,1,@_活动开始

# ✅ 无限次执行
2025/10/12,19:00:00,RUNONDAY,1,999999999,@_活动开始
```

5. Enable设置为0

症状: 配置存在但不执行

原因: 任务被禁用

```
# ❌ 任务被禁用
2025/10/12,19:00:00,RUNONDAY,0,999999999,@_活动开始
                                ↑

# ✅ 任务启用
2025/10/12,19:00:00,RUNONDAY,1,999999999,@_活动开始
                                ↑
```

6. 日志中没有执行记录

症状: GlobalMessage没有在日志中显示

原因: GlobalMessage默认记录日志，检查日志路径

日志位置: Database/Log/SystemLog/YYYY-MM-DD HH MM SS.txt

验证执行:

正常日志

[2025年10月12日 0:40:00]: [NPCCheck调试] 所有检查成功 - 玩家:假人

[2025年10月12日 0:40:00]: 机器人配置测试!!!

[2025年10月12日 0:40:00]: [NPC脚本Break] 页面:, 玩家:假人 - 脚本执行中断

调试技巧

1. 查看机器人加载状态

查看系统日志中的加载信息:

机器人.模板已经加载, 数量: x

2. 测试简单任务

使用5秒测试任务验证系统是否正常:

2025/10/12,00:00:05,ROSEC,1,999999999,@_测试

[@_测试]

#IF

#ACT

GlobalMessage "测试成功!" TRUE

Break

3. 检查执行时间

临时修改配置, 将执行时间设置为当前时间后2-3分钟, 观察是否执行。


4. 重新加载配置

修改CSV后, 需要重启服务器或使用GM命令重新加载:


@ReloadNPCScript

性能优化建议

1. 避免高频任务

 不推荐: 每秒执行

2025/10/12,00:00:01,ROSEC,1,999999999,@_高频任务

 推荐: 每5秒或更长

2025/10/12,00:00:05,ROSEC,1,999999999,@_正常任务

2. 合理使用Count

```
# 测试任务：执行10次后自动停止
2025/10/12,00:00:05,ROSEC,1,10,@_测试任务

# 正式任务：无限执行
2025/10/12,19:00:00,RUNONDAY,1,999999999,@_每日活动
```

3. 脚本优化

```
# ❌ 不推荐：复杂逻辑
[@_复杂任务]
#IF
复杂条件1
复杂条件2
复杂条件3
#ACT
大量操作...

# ✅ 推荐：简单高效
[@_简单任务]
#IF
#ACT
GlobalMessage "公告内容" TRUE
Break
```

附录

相关文件位置

```
项目根目录/
├─ Database/
│   └─ System/
│       └─ 其他配置/
│           └─ 机器人配置.csv          # 任务配置
│           └─ Envir/
│               └─ 00-QFunction.txt      # 脚本实现
├─ Log/
│   └─ SystemLog/
│       └─ YYYY-MM-DD HH MM SS.txt # 系统日志
```

代码文件说明

版本历史

- **v1.0** (2025-10-12): 初始版本，机器人系统修复与文档编写

技术支持

如遇到问题，请检查：

1. 系统日志： Database/Log/SystemLog/
2. 配置文件： Database/System/其他配置/机器人配置.csv
3. 脚本文件： Database/System/Envir/00-QFunction.txt