

# DBTool使用说明

---

文件标识：RK-SM-YF-388

发布版本：V1.0.0

日期：

文件密级：☐绝密 ☐秘密 ☐内部资料 ☒公开

## 免责声明

本文档按“现状”提供，瑞芯微电子股份有限公司（“本公司”，下同）不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因，本文档将可能在未经任何通知的情况下，不定期进行更新或修改。

## 商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标，归本公司所有。

本文档可能提及的其他所有注册商标或商标，由其各自拥有者所有。

版权所有 © 2020 瑞芯微电子股份有限公司

超越合理使用范畴，非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址：福建省福州市铜盘路软件园A区18号

网址：[www.rock-chips.com](http://www.rock-chips.com)

客户服务电话：+86-4007-700-590

客户服务传真：+86-591-83951833

客户服务邮箱：[\[fae@rock-chips.com\]](mailto:fae@rock-chips.com)

前言

概述

本文介绍如何通过修改web数据库。

产品版本

芯片名称	内核版本
RV1109, RV1126, RK1808, RK1806	Linux 4.19

读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

修订记录

版本号	作者	修改日期	修改说明
V1.0.0	陈茂森	2020-09-23	初始版本

# 目录

## DBTool使用说明

1. 编译
2. 命令介绍
  - 2.1 帮助命令
  - 2.2 模式设定
  - 2.3 文件路径设置
  - 2.4 自动模式
  - 2.5 字符模式
3. 使用实例
  - 3.1 json转db
    - 3.1.1 基础模式
    - 3.1.2 自动模式
  - 3.2 db转json
    - 3.2.1 普通模式
    - 3.2.2 字符模式
  - 3.3 json文件规范
    - 3.3.1 普通模式
    - 3.3.2 其他模式
  - 3.4 getdiff模式
  - 3.5 patchdiff模式
  - 3.6 diffwork模式
4. json文件说明
  - 4.1 普通json文件
    - 4.1.1 增加行数据
    - 4.1.2 增加列数据
    - 4.1.3 default介绍
  - 4.2 json能力集说明
    - 4.2.1 能力集简介
    - 4.2.2 能力集para详细介绍
      - 4.2.2.1 static/dynamic
      - 4.2.2.2 disabled
  - 4.3 WebPage
5. 命名规范
6. 常见错误提示

# 1. 编译

---

如下演示不适用buildroot编译dbtool方法。

```
# 在dbtool目录下
mkdir build && cd build # 新建目录管理build文件
cmake .. # cmake
make # 在build目录生成dbtool工具
```

## 2. 命令介绍

---

### 2.1 帮助命令

命令：--help [option] / -h [option]。

介绍基本指令的运用：当option为空选项可直接显示帮助内容。

显示db规范：--help db / -h db。

显示json文件规范：--help json / -h json。

显示能力集规范：--help sys / -h sys。

### 2.2 模式设定

使用--mode <option> / -m <option> 对DBTool运行模式进行设置。

默认模式，在未使用模式设定指令时，DBTool默认模式为json文件转db文件。

json文件转db文件：--mode js2db / -m js2db。

db文件转json文件：--mode db2js / -m db2js。

json文件规范化：--mode js2js / -m js2js。

获取记录文件与原文件的json差异diff文件：--mode getdiff / -m getdiff。

根据diff文件自动修改json文件生成patch文件：--mode getdiff / -m getdiff。

根据file.json中记录的文件，自动查找diff文件并执行patch：--mode diffwork / -m diffwork。

### 2.3 文件路径设置

使用 --jspath | -j [option] 设置json文件路径，其中option为路径。json文件默认路径"sysconfig.json"。

使用 --dbpath | -d [option] 设置db文件路径，其中option为路径。db文件默认路径"./sysconfig.db"。

使用 `--comparepath | -c [option]` 设置对比json文件路径，其中option为路径，对比文件默认路径 `"/compare.json"`。

使用 `--diffpath | -f [option]` 设置diff文件路径，其中option为路径，diff默认路径 `"/sysconfig.json.diff"`。

使用 `--difffile | -df [option]` 设置diff记录文件路径，其中option为路径，diff默认路径 `"/diff/file.json"`。

## 2.4 自动模式

在自动校准模式下，当json文件出现与能力集规范不同时，将不会进行询问，直接对数据进行校准。

在DBTool运行模式设置为json文件转db文件时，增加 `--auto | -a`可进入自动校准模式。

## 2.5 字符模式

字符模式下，生成的json文件中，能力集的para参数将以字符串形式展示，而非json对象。

生成的字符串可用于dbserver。

在DBTool运行模式设置为db文件转json文件或json文件规范化时，增加 `--string | -s`可进入字符模式。

# 3. 使用实例

以下实例均为在linux环境下使用，DBTool 文件名为dbtool。

## 3.1 json转db

### 3.1.1 基础模式

在DBTool所在路径输入如下命令，可将同路径下的sysconfig.json文件转为sysconfig.db文件。

```
./dbtool -j sysconfig.json -d sysconfig.db
```

若该路径下已存在sysconfig.db文件，新生成的sysconfig.db文件将会将原文件覆盖。

在生成db文件过程中，若sysconfig.json文件中的数据与能力集设置冲突，将会有如下提示：

```
in video id 2,sStreamType="thirdStream" is wrong,
autoModify(a)/ignore(i)/delete(d)?
```

其中video为数据库表名，

id为数据库中行的id，

sStreamType="thirdStream"为与能力集冲突的项。

此时用户将有三个选项，autoModify|a, ignore|i, delete|d。

若输入autoModify|a，将会根据能力集自动调整冲突的项，调整规则见。

若输入ignore|i，将会无视此冲突。

若输入delete|d，将会包含冲突项的整行数据删除。

### 3.1.2 自动模式

在DBTool所在路径输入如下命令，可将同路径下的sysconfig.json文件转为sysconfig.db文件。

```
./dbtool -j sysconfig.json -d sysconfig.db -a
```

若该路径下已存在sysconfig.db文件，新生成的sysconfig.db文件将会将原文件覆盖。

在生成db文件过程中，若sysconfig.json文件中的数据与能力集设置冲突，将不会有如[调整提示](#)，默认选择自动调整。

自动处理规则（类型介绍见能力集para详细介绍）：

当类型为range时，当实际值大于最大值时，将调整实际值为最大值；当实际值小于最小值时，将调整实际值为最小值。

当类型为options时，当实际值不在options内，将调整实际值为options的第一个选项。

当类型为dynamicRange时，同类型为range。

当类型为options/dynamicRange时，当实际值大于最大值时，将调整实际值为最接近最大值的合格选项；当实际值小于最小值时，将调整实际值为最接近最小值的合格选项。

## 3.2 db转json

### 3.2.1 普通模式

在DBTool所在路径输入如下命令，可将同路径下的sysconfig.db文件转为sysconfig.json文件。

```
./dbtool -m db2js -j sysconfig.json -d sysconfig.db
```

该模式下生产的json文件中能力集中para参数将为json形式，如下。

```
{
  "id": 1,
  "name": "screenshotSchedule",
  "para": [
    {
      "color": "#87CEEB",
      "name": "timing"
    }
  ]
}
```

### 3.2.2 字符模式

```
./dbtool -m db2js -j sysconfig.json -d sysconfig.db -s
```

该模式下生产的json文件中能力集中para参数将为json字符串形式，如下。

```
{
  "id": 1,
  "name": "screenshotSchedule",
  "para": "[{\"color\": \"#87CEEB\", \"name\": \"timing\"}]"
}
```

## 3.3 json文件规范

json文件规范化实质是将json文件转为db文件后，在将对应db文件转为json文件。

### 3.3.1 普通模式

在DBTool所在路径输入如下命令，可将同路径下的sysconfig.json文件转为sysconfig.json.modify文件。

```
./dbtool -m js2js -j sysconfig.json -d sysconfig.db
```

### 3.3.2 其他模式

因 json文件规范实质为json->db->json，因此json转db的自动模式，db转json的字符模式，均适用。

具体使用见对应实例。

## 3.4 getdiff模式

在DBTool所在路径输入如下命令，将比较compare.json文件与sysconfig.json的差异生成，compare.json.diff文件。

```
./dbtool -m getdiff -j sysconfig.json -c compare.json
```

此模式下会将原有json数组格式的sysconfig.json与compare.json先转换为json对象再进行对比。因此请勿随意修改生成的diff文件。

## 3.5 patchdiff模式

在DBTool所在路径输入如下命令，根据[先前生成的diff文件](#)修改sysconfig.json，生成sysconfig.json.patch。sysconfig.json.patch的实际内容与compare.json一致。

```
./dbtool -m patchdiff -j sysconfig.json -c compare.json.diff
```

此模式会将sysconfig.json先转换为json对象进行与diff文件的patch，后再转换为json数组存储到sysconfig.json.patch文件中。

## 3.6 diffwork模式

diffwork模式下将根据file.json中记录的文件，批量进行patchdiff工作。

```
./dbtool -m diffwork -j sysconfig.json -df ../build/file.json
```

## 4. json文件说明

### 4.1 普通json文件

json文件为一个json数组，数组中每一个单元为一个json对象，该对象有三个属性：tableName、items、default。

1.tableName: 类型为字符串，内容为表名，对应数据库的表名。

2.items: 类型为json数组，每一个单元为数据项，对应数据库中一行的数据。

3.default: default用于创建数据表。类型为json数组，每一个单元为json对象，其columnName对应数据库中的列名，其setting为该列的数据类型，默认值等设置。

实例如下：

```
[
  {
    "1.tableName": "EventSchedules",
    "2.items": [
      {
        "id": 0,
        "sSchedulesJson": ""
      }
    ],
    "3.default": [
      {
        "columnName": "id",
        "setting": "INTEGER PRIMARY KEY AUTOINCREMENT"
      },
      {
        "columnName": "sSchedulesJson",
        "setting": "TEXT"
      }
    ]
  }
]
```



### 4.1.1 增加行数据

在items中增加json对象，对象key为数据表的列名，对象value为对应列数据。即可在新生成的数据库中增加行数据。实例如下，增加数据id为1，sSchedulesJson为空。

```
[
  {
    "1.tableName": "EventSchedules",
    "2.items": [
      {
        "id": 0,
        "sSchedulesJson": ""
      },
      {
        "id": 1,
        "sSchedulesJson": ""
      }
    ],
    "3.default": [
      {
        "columnName": "id",
        "setting": "INTEGER PRIMARY KEY AUTOINCREMENT"
      },
      {
        "columnName": "sSchedulesJson",
        "setting": "TEXT"
      }
    ]
  }
]
```

### 4.1.2 增加列数据

在原有的数据表json的基础上增加列，若未设定默认值，需要同时修改items，default。若设定默认值，仅需修改default。如下，不设定默认值的情况如下增加sName列，需在items内增加sName的定义。

```
[
  {
    "1.tableName": "EventSchedules",
    "2.items": [
      {
        "id": 0,
        "sSchedulesJson": "",
        "sName": "test1"
      }
    ],
    "3.default": [
      {
        "columnName": "id",
        "setting": "INTEGER PRIMARY KEY AUTOINCREMENT"
      },
      {
        "columnName": "sSchedulesJson",
        "setting": "TEXT"
      }
    ]
  }
]
```

```

    },
    {
      "columnName": "sName",
      "setting": "TEXT"
    }
  ]
}
]

```

如下，在设定默认值时，若未在items内定义新增的sName，则使用默认值。若定义sName则使用定义

```

[
  {
    "1.tableName": "EventSchedules",
    "2.items": [
      {
        "id": 0,
        "sSchedulesJson": ""
      }
    ],
    "3.default": [
      {
        "columnName": "id",
        "setting": "INTEGER PRIMARY KEY AUTOINCREMENT"
      },
      {
        "columnName": "sSchedulesJson",
        "setting": "TEXT"
      },
      {
        "columnName": "sName",
        "setting": "TEXT DEFAULT 'TEST'"
      }
    ]
  }
]

```

### 4.1.3 default介绍

default为建表属性，columnName规定列名，setting规定数据类型、默认值以及特别属性。

列名的规范见[命名规范](#)。

数据类型：目前表支持两种数据类型，TEXT字符型，NUMBER数字型，使用其他类型，可能存在与cgi或其他应用的冲突。数字类型推荐置于default 头部。

默认值：使用DEFAULT开头后跟默认值，若列的数据类型为字符型默认值需要使用单引号包括，若为数字型则不需要。实例如下：

```

# 未设置主键
"setting": "TEXT DEFAULT 'test1'"
"setting": "NUMBERL DEFAULT 0"

```

特别属性：同其他sql的特别属性设置，本篇仅介绍主键，以及自增属性。

主键：每个数据表必带项，不可有重复值，若未设定将会由数据库自动生成（不推荐）。一般使用id作为

主键。使用如下：在规定好的setting后加入PRIMARY KEY则将此列设置为主键。

```
# 未设置主键
{
  "columnName": "id",
  "setting": "INTEGER"
}
# 设置主键
{
  "columnName": "id",
  "setting": "INTEGER PRIMARY KEY"
}
```

自增属性：仅数字型的列可设置该属性，在属性最后添加AUTOINCREMENT以设置此属性，在新增items时，若未规定该列属性，则会在上一个的结果上+1。实例如下，在增加第二个items时，由于未规定id，id会在上个items的基础上+1，即最后第二项的id为1。

```
[
  {
    "1.tableName": "EventSchedules",
    "2.items": [
      {
        "id": 0,
        "sSchedulesJson": ""
      },
      {
        "sSchedulesJson": ""
      }
    ],
    "3.default": [
      {
        "columnName": "id",
        "setting": "INTEGER PRIMARY KEY AUTOINCREMENT"
      },
      {
        "columnName": "sSchedulesJson",
        "setting": "TEXT"
      }
    ]
  }
]
```

## 4.2 json能力集说明

## 4.2.1 能力集简介

tableName为SystemPara的为能力集，能力集必须位于json文件数组的第一位。

能力集的构成与其他表相同，每项能力集有id，name，para三个参数。

id: 为该能力集在SystemPara中的序号。

name: 为能力集所限制的表的表名，若不存在与name同名的表，则该能力集不生效，web端用于映射的能力集均无同名表。

para: json对象，包含能力集具体参数。

para对象有static、dynamic、relation、disabled、layout这5种属性。

static: 静态属性，在任何情况下表单均需满足的条件。如下，表明iImageQuality仅允许为1、5、10。

```
"static": {
  "iImageQuality": {
    "options": [
      1,
      5,
      10
    ],
    "type": "options"
  }
}
```

dynamic: 动态属性，在规定的情况下需要满足的条件。如下表示，当id为0时，iShotInterval列的最大值为604800000，最小值为1000。

```
"dynamic": {
  "id": {
    "0": {
      "iShotInterval": {
        "for": "timing",
        "range": {
          "max": 604800000,
          "min": 1000
        },
        "type": "range"
      }
    }
  }
}
```

relation: 映射关系，仅web前端映射使用，无实际限制功能。如下表示，在web端，当iImageQuality为1时，显示的值为low。

```
"relation": {
  "iImageQuality": {
    "1": "low",
    "5": "middle",
    "10": "high"
  }
}
```

**disabled:** 禁用条件。与dynamic类似，在规定的情况下，限制值。禁用功能为web前端使用。如下表示，当sStreamType为subStream时，sOutputDataType限定H.264，并被禁止变更。

```
"disabled": [
  {
    "name": "sStreamType",
    "options": {
      "subStream": {
        "sOutputDataType": "H.264",
        "sSmart": "close"
      },
      "thirdStream": {
        "sSmart": "close"
      }
    },
    "type": "disabled/limit"
  }
]
```

**layout:** web前端属性，用于布局。如下，web前端将按数组顺序来展示词条。

```
"layout": {
  "encoder": [
    "sStreamType",
    "sVideoType",
    "sResolution",
    "sRCMode",
    "sRCQuality",
    "sFrameRate",
    "sOutputDataType",
    "sSmart",
    "sH264Profile",
    "sSVC",
    "iMaxRate",
    "iMinRate",
    "iGOP",
    "iStreamSmooth"
  ]
}
```

## 4.2.2 能力集para详细介绍

### 4.2.2.1 static/dynamic

static以及dynamic的最小单元如下：

```
<column_name>: {
  <type>: type_name,
  [for]: for_name,
  <type_name>: detail
}
```

最外层的key<column\_name>对应需要限制的列名；内层<type>为限制的类型；[for]为可选项，说明限制目的，无实际作用；已限制类型明明为key的<type\_name>，存放具体限制内容detail。

1.type: 主要有options、range、dynamicRange、options/dynamicRange、refer这五种。

options: 选项型，column内的值必须在options的数组内。如下，iImageQuality仅允许为1、5、10。

```
"static": {
  "iImageQuality": {
    "options": [
      1,
      5,
      10
    ],
    "type": "options"
  }
}
```

range: 取值范围型，在对应detail中的属性有，min、max、step（可选）。当step规定时，在web前端将会被展示为滑动条；当step未规定时，在web端展示为数字输入框。实例如下：

```
"iShotInterval": {
  "for": "timing",
  "range": {
    "max": 604800000,
    "min": 1000
  },
  "type": "range"
}
```

dynamicRange: 动态取值型，如下，iMinRate为动态取值，其最大值为同row的iMaxRate的1倍，其最小值为100的1倍。若仅有最小值无最大值，可不给出max和maxRate。

```
"iMinRate": {
  "dynamicRange": {
    "max": iMaxRate,
    "maxRate": 1,
    "min": 100,
    "minRate": 1
  },
  "type": "dynamicRange"
}
```

options/dynamicRange: 选项并规定取值范围。如下：sFrameRate的值必须在options中规定，且最大值为同row的sFrameRateIn的一倍。options内必须全为字符串，或全为数字。当options为字符串时支持使用分数。

```
"sFrameRate": {
  "dynamicRange": {
    "max": "sFrameRateIn",
    "maxRate": 1
  },
  "options": [
    "1/16",
    "1/8",
  ]
}
```

```

        "1/4",
        "1/2",
        "1",
        "2",
        "4",
        "6",
        "8",
        "10",
        "12",
        "14",
        "16",
        "18",
        "20",
        "25",
        "30"
    ],
    "type": "options/dynamicRange"
}

```

refer: 引用型，将会根据refer数组内的内容检索限制条件。如下，sStreamType的限制条件，将会与SystemPara表中id为4的表，para列中，static类型下的sStreamType相同。即在SystemParajson化后按照refer中的顺序以及key进行检索，将检索到的条件赋给发起检索的对象。

```

"sStreamType": {
    "refer": [
        4,
        "para",
        "static",
        "sStreamType"
    ],
    "type": "refer"
}

```

#### 4.2.2.2 disabled

disabled内的type仅两种：disabled、disabled/limit。

disabled: 仅web端使用，当满足条件时将禁用选项。

disabled/limit: 禁用的同时，将对某些column进行限制。在文件转换过程中生效。实例见能力集简介[disabled](#)。

## 4.3 WebPage

WebPage为特殊能力集用于规范web各个功能权限以及显示与否。基本单元如下。

auth: 权限等级，数字越大要求权限越低，拥有大于等于auth权限即可访问对应功能界面。-1为禁止访问，一般为该产品不具备的功能；0为管理员；1为操作员；2为普通用户；3为预留；4为任意用户；

name: 单元名；

item: 可选项，子单元；

说明: 若未满足上层权限要求，则直接不可访问子单元。

# 基本单元

```
{
  "auth": 4,
  "item": [],
  "name": ""
}
```

# 实际展示

```
{
  "id": 0,
  "name": "webPage",
  "para": {
    "auth": 4,
    "item": [
      {
        "auth": 4,
        "name": "preview"
      },
      {
        "auth": 4,
        "item": [
          {
            "auth": 4,
            "item": [
              {
                "auth": 0,
                "name": "delete"
              }
            ],
            "name": "videoRecord"
          },
          {
            "auth": 4,
            "item": [
              {
                "auth": 0,
                "name": "delete"
              }
            ],
            "name": "pictureRecord"
          }
        ],
        "name": "download"
      }
    ],
    "auth": 4,
    "item": [
      {
        "auth": 4,
        "item": [
          {
            "auth": 4,
            "item": [
              {
                "auth": 1,
                "name": "modify"
              }
            ]
          }
        ]
      }
    ]
  }
}
```



```

        "name": "ListManagement"
    },
    {
        "auth": 1,
        "name": "AddOne"
    },
    {
        "auth": 1,
        "name": "BatchInput"
    }
],
"name": "MemberList"
},
{
    "auth": 4,
    "item": [
        {
            "auth": 4,
            "item": [
                {
                    "auth": 0,
                    "name": "modify"
                }
            ],
            "name": "SnapShot"
        }
    ],
    "name": "SnapShot"
},
{
    "auth": 4,
    "item": [
        {
            "auth": 4,
            "item": [
                {
                    "auth": 0,
                    "name": "modify"
                }
            ],
            "name": "Control"
        }
    ],
    "name": "Control"
},
{
    "auth": 1,
    "item": [
        {
            "auth": 1,
            "name": "ParaConfig"
        }
    ],
    "name": "Config"
}
],
"name": "face"
},

```

```

{
  "auth":-1,
  "item":[
    {
      "auth":1,
      "item":[
        {
          "auth":1,
          "name":"FacePara"
        },
        {
          "auth":1,
          "name":"ROI"
        }
      ],
      "name":"Config"
    }
  ],
  "name":"face-para"
},
{
  "auth":-1,
  "item":[
    {
      "auth":4,
      "item":[
        {
          "auth":4,
          "item":[
            {
              "auth":1,
              "name":"modify"
            }
          ],
          "name":"MemberList"
        }
      ],
      "name":"AddOne"
    },
    {
      "auth":1,
      "name":"BatchInput"
    },
    {
      "auth":4,
      "item":[
        {
          "auth":0,
          "name":"modify"
        }
      ],
      "name":"SnapShot"
    },
    {
      "auth":4,
      "item":[
        {

```



```

        "auth":-1,
        "name":"securityService"
    }
],
    "name":"Security"
},
{
    "auth":0,
    "name":"User"
}
],
    "name":"System"
},
{
    "auth":1,
    "item":[
        {
            "auth":1,
            "item":[
                {
                    "auth":1,
                    "name":"TCPIP"
                },
                {
                    "auth":-1,
                    "name":"DDNS"
                },
                {
                    "auth":-1,
                    "name":"PPPoE"
                },
                {
                    "auth":1,
                    "name":"Port"
                },
                {
                    "auth":-1,
                    "name":"uPnP"
                }
            ],
            "name":"Basic"
        },
        {
            "auth":1,
            "item":[
                {
                    "auth":1,
                    "name":"Wi-Fi"
                },
                {
                    "auth":-1,
                    "name":"SMTP"
                },
                {
                    "auth":-1,
                    "name":"FTP"
                },
                {

```

```

        "auth":-1,
        "name":"eMail"
    },
    {
        "auth":-1,
        "name":"Cloud"
    },
    {
        "auth":-1,
        "name":"Protocol"
    },
    {
        "auth":-1,
        "name":"QoS"
    },
    {
        "auth":-1,
        "name":"Https"
    }
],
    "name":"Advanced"
}
],
    "name":"Network"
},
{
    "auth":1,
    "item":[
        {
            "auth":1,
            "name":"Encoder"
        },
        {
            "auth":1,
            "name":"AdvancedEncoder"
        },
        {
            "auth":1,
            "name":"ROI"
        },
        {
            "auth":1,
            "name":"RegionCrop"
        }
    ],
    "name":"Video"
},
{
    "auth":1,
    "item":[
        {
            "auth":1,
            "name":"AudioParam"
        }
    ],
    "name":"Audio"
},
{

```

```

        "auth":1,
        "item":[
            {
                "auth":1,
                "name":"DisplaySettings"
            },
            {
                "auth":1,
                "name":"OSDSettings"
            },
            {
                "auth":1,
                "name":"PrivacyCover"
            },
            {
                "auth":1,
                "name":"PictureMask"
            }
        ],
        "name":"Image"
    },
    {
        "auth":1,
        "item":[
            {
                "auth":1,
                "name":"MotionDetect"
            },
            {
                "auth":1,
                "name":"IntrusionDetection"
            },
            {
                "auth":-1,
                "name":"AlarmInput"
            },
            {
                "auth":-1,
                "name":"AlarmOutput"
            },
            {
                "auth":-1,
                "name":"Abnormal"
            }
        ],
        "name":"Event"
    },
    {
        "auth":1,
        "item":[
            {
                "auth":1,
                "item":[
                    {
                        "auth":1,
                        "name":"VideoPlan"
                    },
                    {

```

```

        "auth":1,
        "name":"ScreenshotPlan"
    },
    {
        "auth":1,
        "name":"ScreenshotPara"
    }
],
"name":"PlanSettings"
},
{
    "auth":1,
    "item":[
        {
            "auth":1,
            "name":"HardDiskManagement"
        },
        {
            "auth":-1,
            "name":"NAS"
        },
        {
            "auth":-1,
            "name":"CloudStorage"
        }
    ],
    "name":"StorageManage"
}
],
"name":"Storage"
},
{
    "auth":1,
    "item":[
        {
            "auth":1,
            "name":"MarkCover"
        },
        {
            "auth":-1,
            "name":"MaskArea"
        },
        {
            "auth":-1,
            "name":"RuleSettings"
        },
        {
            "auth":-1,
            "name":"AdvancedCFG"
        }
    ],
    "name":"Intel"
},
{
    "auth":-1,
    "item":[
        {
            "auth":-1,

```

```
        "name": "GateConfig"
    },
    {
        "auth": -1,
        "name": "ScreenConfig"
    }
],
"name": "Peripherals"
}
],
"name": "config"
},
{
    "auth": 4,
    "name": "about"
}
],
"name": "header"
}
}
```

## 5. 命名规范

表名：优先使用大驼峰命名，如TableName。

列名：必须使用小驼峰命名。若数据类型为number，必须使用i开头，如：iPeopleNumber。当数据类型为TEXT时，推荐使用s开头，如：sName。暂不支持其他类型的数据。

## 6. 常见错误提示

```
XXX isn't a table
```

当json文件中缺少items、default中任意一个时，提示如上，XXX为表名。

```
json file doesn't existst!
```

当json文件不存在时提示如上，仅在js转db模式以及json规范化模式中出现。

```
db is empty!
```

当数据库文件为空时，提示如上。

```
json file is empty
```

当json文件为空时提示如上，仅在js转db模式以及json规范化模式中出现。

```
json file lose SystemPara
```



当json文件第一个项不是SystemPara时，提示如上。