

# Sepics Guide New and Improved Second Edition

By Molten Thinker

Email: [moltenthinker@moltenstudios.com](mailto:moltenthinker@moltenstudios.com)

Website: <http://oni.moltenstudios.com/>

## **Disclaimer:**

We are not responsible for any damage this guide could have caused in any way. Use at your own risk. This guide may be reproduced and redistributed in every or anyway without modification. Any results based on this guide should have proper referencing.

# Table of Contents

## **I. Sepics Notes.**

1. Introduction
2. Hex Editing
3. Structure
4. Level Zero

## **II. Resources:**

1. CJBO (Object)
2. RAHC (Character)
3. SNOC (Console)
4. ROOD (Door)
5. GALF (Flag)
6. NRUF (Furniture)
7. TUEN (Neutral)
8. TRAP (Part)
9. RTAP (Patrol)
10. URWP (Powerup)
11. GDNS (Sound Generator)
12. VGRT (Trigger Volume)
13. GIRT (Trigger)
14. RRUT (Turret)
15. PAEW (Weapon)

# **I. Sepics Notes**

## **1 Introduction**

Sepics is the manipulation of \*.sep files (they are in the GameDataFolder). These files are responsible for the characters, doors, consoles, patrols and other game values. \*.dat files on the other hand, implement them. For example, \*.sep files contains the door data, but the \*.dat file contains the texture and skin of it.

The file structure of PCs and Macs is different in Oni. However, PC users do have the same \*.sep resources in \*.raw files.

## **2 Hex Editing**

To perform Sepics, open up a \*.sep file in a Hex Editor. You have to preserve length, i.e your initial file must have the same length as your final file. This means you would have to add or delete (.)s to maintain length. If you do not preserve length or mess up the \*.sep file otherwise then Oni will crash while loading the level with that \*.sep file. If you made a spelling error, a log file will open.

## **3 Structure**

Sep files like \*.dat files are made out of resources, only they are much less readable. They come in a hierarchy. After each resource, there is a character which identifies the number of the resource. The characters go in ASCII order in the list. Resources come in different types. Type is a four letter code. To understand what the type of resource does simply read it backwards.

## 4 Level Zero

Like the Level Zero data file (level0\_Final.dat), Level Zero sep file (level0\_Final.sep) is pretty special. If you mess it up, the game would not start. It is loaded at the launch of the game. It contains most of the materials and global objects of the game. It has its own \*.sep resources, for instance: **EINO**, **3RAP**, **3rGSO**, **mASO**, **mISO**, **DBAS** and **SAVT**.

## II. Standard Resources

### 1 CJBO (Object)

**CJBO** is an Object Collection or a gathering of a few types of data in an ordered list. Each **CJBO** is a collection of data for its own \*.sep resource. In fact, all sets of resources are stored in there own **CJBO**.

### 2 RAHC (Character)

**RAHC** is responsible for character data. Lets consider a few examples:

# RAHC Example I

[

...

**RAHC**.....**INVC**..**mutantmuro**

**MutantMuro**

**muro\_dummy**

]

This **RAHC** is about the character **MutantMuro** of type **mutantmuro** with an **INVC** powerup and at death the scripting function **muro\_dummy** is executed. Now, take a look at the Compound scripting file. If you remove the **muro\_dummy** function in scripting, then level will not run because **muro\_dummy** is used in the \*.sep file.

```
# RAHC Example II
[
...
RAHC.....
elite_hard_3D_E83
none
]
```

Here we have a **D\_E83** of type **elite\_hard\_3** with the weapon **none**. This cannot possibly be hard. If you still didn't get it, let's consider another example:

```
# RAHC Example III
[
...RAHC.....
.....striker_med_3A_E5bw2_sap
]
```

Here we are talking a medium striker with a small gun. Notice the scripting name tells us that he is in region A or at the beginning of the level. Thank the game creator for this.

### 3 SNOC (Console)

**SNOC** is the console resource. Lets consider an example:

```
# SNOC Example
[
SNOC....
.....console_alarm._con_ALARM_SLEEP_con_ALARM_SLEEP_con_ALARM_ON...run1_alarm
]
```

This just happens to be an alarm console. When it is used, it makes a few noises and animations. Then, the scripting function **run1\_alarm** is executed.

### 4 ROOD (Door)

**ROOD** is the door resource: you know the things you open and close? Yes, yes, those ones. Anyway, lets take a look:

```
# ROOD Example I
[
ROOD.....4BP.....CM_1.....BP.....
]
```

```
# ROOD Example II
[
ROOD.....CM_1.....D_DOOR_CM_1CON2
]
```

Doors can trigger some functions as well when they are opened (or closed).

## 5 GALF (Flag)

**GALF** – flag resource. Flags are ‘special’ places in the game. If you do, scripting, they are four-digit codes where you teleport your guys to. Example below shows a flag for a place in the region **4B**.

# GALF Example

```
[  
..GALFU.....4B.....W.....XM.  
]
```

## 6 NRUF (Furniture)

**NRUF** – furniture: some object in the game which just takes up space, clutter.

# NRUF Example I

```
[  
tNRUF.....U_trailer_2  
]
```

# NRUF Example II

```
[  
tNRUF6.....CRF_antenna4  
]
```

## 7 TUEN (Neutral)

**TUEN** – You may chat with Neutral characters thanks to TUENs.

For instance,

```
# TUEN Example I
```

```
[  
P.TUENB.....DGive 2 Ammo (CivMale1)  
.....  
BHCåBcivmale1_triggercivmale1_abortcivmale1_enemynone.....  
civmale1_gift_ammo.talking  
]
```

You are talking to CivMale1. He will Give 2 Ammo. You will trigger him by civmale1\_trigger. Since he will be giving you ammunition, he will use the animation civmale1\_gift\_ammo. Then, he will abort and wouldn't give you anything anymore. Here is another example:

```
# TUEN Example II
```

```
[  
P.TUEN].....DGive TCTF Autopistol  
(CivFem).....BHCåBcivfem_triggercivfem_abortcivfe  
m_enemyw1_tap.civfem_gift_weapon.....  
]
```

## 8 TRAP (Part)

As an example, lets take this environmental effect (part) in region 4C. That means at 4C, we would find snow snow falling down. If we change it to something else, it would find another effect going on.

```
# TRAP Example
```

```
[  
Ä?Ä?†TRAP.....4Cenv_snow_e01snow  
]
```



## 9 RTAP (Patrol)

**RTAP** is for patrols. These make characters moving back and forth along a certain path patrolling the area.

# RTAP Example

```
[  
  RTAP.....Dpatrol_03.....  
]
```

This explains what the patrol [Dpatrol03](#) is. Switching the patrol code would make characters do a different patrol.

## 10 URWP (Powerup)

**URWP** is stored in the same **CJBO** listing as **TRAPs** and **RTAPs**. I copied all the **URWP** resources from a file:

# URWP Example

```
[  
  AMME(URWP.  
    ....4B....HYPO(URWP{.....HYPO(URWP.....  
    .....INVI(URWP.....4B....SHLD(URWP.....  
    ...4B®:»DHYPO(URWP.....INVI(URWP.....  
    .....4B.....AMME(URWP.....AMME(U  
  RWP.....AMMB(URWP.....)C4....INVI(URWP  
    .....)INVI(URWP2.....HYPO(URWP...  
    .....HYPO(URWP.....[...DAMME(UR  
  WP.....DAMMB(URWP.....INVI(URWP  
    .....AMMB(URWP.....HYPO(URWP!.....  
    .....AMME(URWP.....HYPO(URWP#.....  
    .....AMMB(URWP.....INVI.....  
]
```

This is really a collection of power-ups on the level.

## 11 GDNS (Sound Generator)

**GDNS** is a sound generator. They come in two types: **RHPS** (sphere) and **EMLV** (volume). These are two different ways to spread the volume over a region.

```
# GDNS Example I
```

```
[  
hGDNS{.....windRHPS.....
```

```
# GDNS Example II
```

```
[  
hGDNS.....windEMLV.....h  
]
```

## 12 VGRT (Trigger Volume)

**VGRT** triggers volumes (changes the background music or how loud the sounds are). Example:

```
# VGRT Example I
```

```
[  
VGRTJ.....4B.....trigger_volume_08_copyt8.....  
]
```

Each VGRT triggers volume (music or sound). You can have other scripting code than just a trigger function in there too. For example:

```
# VGRT Example II
```

```
[  
....VGRT.....4B.....trigger_volume_32set_objective_2  
]
```

## 13 GIRT (Trigger)

**GIRT** is a classic trigger resource. It controls those triggers that you know in scripting.

```
# GIRT Example
```

```
[  
GIRT.....  
.....4Ctrigger_short.....  
]
```

You could have [trigger\\_short](#) or [trigger\\_long](#).

## 14 RRUT (Turret)

**RRUT** is the responsible for the turrets.

```
# RRUT Example
```

```
[  
RRUT.....4B.....sap_turret_floor...  
]
```

## 15 PEAW (Weapon)

**PEAW** is the responsible for the weapons you can find on the ground. For instance,

# WEAP Example

```
[  
PAEW0.....Dw5_sbgD  
]
```

A super-ball gun is somewhere in the level for you to pick up and blast your enemies.

***That's all folks!***

*Then the \*.sep file breaks into nothingness...*

*Thousands of hex with meaning to be discovered...*