

# Formato de Archivo Fuente SCE de DAAD - Referencia Completa

---

## Índice

1. [Introducción](#)
  2. [Estructura General del Archivo SCE](#)
  3. [Secciones del Archivo SCE](#)
  4. [Los Procesos: Corazón del Sistema DAAD](#)
  5. [Funcionamiento del Sistema de Procesos](#)
  6. [Comandos del Compilador](#)
  7. [Sintaxis y Reglas](#)
  8. [Ejemplos Prácticos](#)
- 

## Introducción

El archivo SCE (Source Code file) es el archivo fuente de DAAD que contiene toda la definición de un juego de aventura conversacional. Este archivo utiliza formato ASCII y contiene todas las secciones necesarias para crear una aventura completa: vocabulario, mensajes, objetos, localizaciones, conexiones y, lo más importante, las **tablas de procesos** que controlan toda la lógica del juego.

### Características del Sistema

- **Multiplataforma:** Un archivo SCE se compila para múltiples sistemas (PC, Spectrum, Atari ST, Amiga, etc.)
  - **Lenguaje de procesos avanzado:** Con características como bucles, indirección y llamadas anidadas
  - **Sistema modular:** Separación clara entre datos (vocabulario, mensajes) y lógica (procesos)
  - **Soporte multi-idioma:** Capacidad para crear versiones en diferentes idiomas
- 

## Estructura General del Archivo SCE

Un archivo SCE está organizado en **secciones obligatorias** que deben aparecer en un orden específico:

```
/CTL      ; Control - Configuración general
/VOC      ; Vocabulary - Vocabulario del juego
/STX      ; System Messages - Mensajes del sistema
/MTX      ; Messages - Mensajes del juego
/OTX      ; Object Names - Nombres de objetos
/LTX      ; Location Names - Nombres de localizaciones
/CON      ; Connections - Conexiones entre localizaciones
/OBJ      ; Objects - Definición de objetos
/PRO      ; Processes - Tablas de procesos (CORAZÓN DEL SISTEMA)
```

## Reglas de Sintaxis General

- **Comentarios:** Las líneas que comienzan con `;` son comentarios
  - **Espacios en blanco:** Se ignoran los espacios al inicio de línea (excepto en PRO)
  - **Continuación de línea:** El carácter `\` al final permite continuar en la siguiente línea
  - **Códigos de escape:** Se pueden usar secuencias como `\n` (nueva línea), `\t` (tab), etc.
  - **Sensibilidad:** El sistema es sensible a mayúsculas y minúsculas
- 

## Secciones del Archivo SCE

### 1. Sección CTL (Control)

Define parámetros globales del juego:

```
/CTL
; Número de localizaciones, objetos, mensajes, etc.
; Definición de banderas especiales
; Configuración de la aventura
```

### 2. Sección VOC (Vocabulary)

Define todas las palabras que el jugador puede usar:

```
/VOC
; Formato: PALABRA TIPO VALOR
NORTH NOUN 1
GO VERB 1
TAKE VERB 10
SWORD NOUN 50
```

#### Tipos de palabras:

- **VERB** (verbos)
- **NOUN** (sustantivos)
- **ADJECTIVE** (adjetivos)
- **ADVERB** (adverbios)
- **PREPOSITION** (preposiciones)
- **PRONOUN** (pronombres)
- **CONJUGATION** (conjugaciones)

### 3. Sección STX (System Messages)

Mensajes predefinidos del sistema:

```
/STX
"You can't go that way."
```

```
"I don't understand."  
"You can't do that."
```

#### 4. Sección MTX (Messages)

Mensajes definidos por el autor del juego:

```
/MTX  
"Welcome to the adventure!"  
"You see a rusty sword here."  
"The door is locked."
```

#### 5. Sección OTX (Object Names)

Nombres y descripciones de objetos:

```
/OTX  
"sword" "rusty sword" "An old rusty sword."  
"key" "golden key" "A small golden key."
```

#### 6. Sección LTX (Location Names)

Nombres y descripciones de localizaciones:

```
/LTX  
"Forest" "You are in a dark forest. Paths lead north and south."  
"Castle" "You stand before an imposing castle gate."
```

#### 7. Sección CON (Connections)

Define las conexiones de movimiento entre localizaciones:

```
/CON  
; Localización 0  
NORTH 1  
SOUTH 2  
; Localización 1  
SOUTH 0  
EAST 3
```

#### 8. Sección OBJ (Objects)

Define propiedades y ubicación inicial de objetos:

```

/OBJ
; objeto ubicación peso atributos
sword 1 10 N N ; espada en localización 1, peso 10
key 255 5 N N ; llave llevada por jugador (255 = CARRIED)

```

## Los Procesos: Corazón del Sistema DAAD

### ¿Qué son los Procesos?

Los **procesos** son tablas que contienen la lógica del juego. Cada proceso es una tabla numerada que contiene **entradas**, y cada entrada tiene:

1. **Verbo y Sustantivo:** Condiciones de activación
2. **CondActs:** Secuencia de condiciones y acciones a ejecutar

### Estructura de un Proceso

```

/PRO 0 ; Proceso número 0 (proceso principal)
- - ; Entrada que coincide con cualquier verbo/sustantivo
MESSAGE 0 ; Acción: mostrar mensaje 0
PARSE ; Acción: analizar entrada del jugador
DOALL 255 ; Acción: procesar todos los objetos en localización actual
PROCESS 1 ; Acción: llamar al proceso 1
RESTART ; Acción: reiniciar bucle principal

/PRO 1 ; Proceso número 1 (procesos de verbos)
TAKE -
PRESENT 100 ; Condición: objeto 100 debe estar presente
GET 100 ; Acción: tomar objeto 100
MESSAGE 10 ; Acción: mostrar mensaje de éxito
DONE ; Acción: terminar procesamiento

LOOK -
DESC ; Acción: describir localización actual
DONE ; Acción: terminar

```

### Tipos de Procesos Especiales

1. **Proceso 0:** El proceso principal (bucle principal del juego)
2. **Proceso 1:** Tradicionalmente maneja verbos comunes
3. **Proceso 2:** Tradicionalmente maneja respuestas a verbos específicos

## Funcionamiento del Sistema de Procesos

### El Bucle Principal (Main Loop)

El intérprete DAAD funciona con el siguiente flujo:

INICIO → Inicialización → Proceso 0 → Sistema Operativo/Reinicio

1. **Inicialización:** Se limpia el estado del juego, jugador en localización 0
2. **Llamada al Proceso 0:** Se ejecuta el proceso principal
3. **Retorno:** Al terminar Proceso 0, se sale al SO o reinicia

## Proceso de Búsqueda en Tablas

Cuando se ejecuta un proceso, DAAD:

1. **Examina cada entrada** de la tabla secuencialmente
2. **Compara Verbo y Sustantivo** con la Oración Lógica actual (LS)
3. **Si coinciden**, ejecuta los CondActs de esa entrada
4. **Si no coinciden**, pasa a la siguiente entrada

## Funcionamiento de las Entradas

```
VERBO  SUSTANTIVO
CONDICIÓN1 parámetros
CONDICIÓN2 parámetros
ACCIÓN1 parámetros
ACCIÓN2 parámetros
```

### Flujo de ejecución:

1. Se verifican las **condiciones** en orden
2. Si una condición **falla**, se abandona la entrada y se pasa a la siguiente
3. Si todas las condiciones **pasan**, se ejecutan las **acciones**
4. Las acciones se ejecutan hasta encontrar una acción de **salida** (**DONE**, **RESTART**, etc.)

## Comodines en Procesos

- **\_ (guión bajo):** Coincide con cualquier palabra
- **Ejemplo:** **\_ \_** coincide con cualquier combinación verbo-sustantivo
- **Uso típico:** En Proceso 0 para capturar todas las entradas

## Tipos de CondActs

### 1. Condiciones

Deben cumplirse para continuar:

```
AT 5          ; ¿Está el jugador en localización 5?
PRESENT 10    ; ¿Está presente el objeto 10?
```

```
CARRIED 15      ; ¿Lleva el jugador el objeto 15?
```

## 2. Acciones Normales

Se ejecutan y continúan:

```
MESSAGE 20      ; Mostrar mensaje 20  
GET 10          ; Tomar objeto 10  
DROP 15         ; Soltar objeto 15
```

## 3. Acciones de Salida

Terminan el procesamiento de la entrada actual:

```
DONE           ; Terminar entrada actual  
RESTART        ; Volver al inicio del Proceso 0  
END            ; Terminar juego
```

## 4. Acciones de Salida Condicional

Terminan solo si fallan:

```
GET 10          ; Si falla (no se puede tomar), termina entrada  
PUTIN 10 20     ; Si falla (no se puede meter), termina entrada
```

## 5. Acciones de Control

Cambian el flujo de ejecución:

```
PROCESS 2       ; Llamar al proceso 2  
SKIP 5          ; Saltar 5 CondActs  
GOTO etiqueta  ; Saltar a etiqueta local
```

## Indirección

Los procesos soportan **indirección** en el primer parámetro:

```
MESSAGE [100]   ; Mostrar mensaje cuyo número está en bandera 100  
GET [50]        ; Tomar objeto cuyo número está en bandera 50
```

# Comandos del Compilador

## Comandos de Compilación Condicional

```
#IFDEF símbolo      ; Si símbolo está definido
#IFNDEF símbolo     ; Si símbolo NO está definido
#ELSE               ; Alternativa
#ENDIF              ; Fin de bloque condicional
```

## Comandos de Definición

```
#DEFINE símbolo valor    ; Definir símbolo con valor
#SYMBOLS archivo         ; Incluir archivo de símbolos
```

## Comandos de Inclusión de Datos

```
#INCBIN archivo          ; Incluir archivo binario
#DEFB 1 2 3 4            ; Definir bytes
#DEFW 1000 2000          ; Definir palabras (16-bit)
#HEX FF00 AB12           ; Incluir datos hexadecimales
```

## Comandos de Direccionamiento

```
#DBADDR símbolo         ; Dar a símbolo la dirección actual
#USERPTR n              ; Vector de usuario (0-9)
```

## Comandos Específicos de 8-bit

```
#EXTERN archivo         ; Vector externo
#SFX archivo            ; Vector de efectos de sonido
#INT archivo            ; Vector de interrupción 50Hz
```

---

# Sintaxis y Reglas

## Reglas de Formato

1. **Secciones obligatorias:** Deben aparecer en el orden especificado
2. **Numeración consecutiva:** Procesos deben numerarse 0, 1, 2, ...
3. **Espacios en PRO:** En sección PRO, columna 1 indica nueva entrada
4. **Etiquetas locales:** Líneas que empiezan con \$ definen etiquetas

## Códigos de Escape

```
\n    ; Nueva línea
\t    ; Tabulación
\\    ; Barra invertida literal
\"    ; Comillas literal
\NNN  ; Carácter por código ASCII (octal)
```

## Limitaciones

- **Vocabulario:** Limitado por memoria disponible
- **Mensajes:** Máximo definido en CTL
- **Objetos/Localizaciones:** Máximo 255 cada uno
- **Procesos:** Sin límite teórico, limitado por memoria
- **Banderas:** 256 banderas (0-255)

## Ejemplos Prácticos

### Ejemplo 1: Proceso Principal Básico

```
/PRO 0
--
CLS          ; Limpiar pantalla
LOOK         ; Describir localización
NEWLINE      ; Nueva línea
MESSAGE 0    ; "What do you want to do?"
PARSE        ; Analizar entrada del jugador
PROCESS 1    ; Procesar verbos comunes
MESSAGE 1    ; "I don't understand"
RESTART      ; Volver al inicio
```

### Ejemplo 2: Proceso de Verbos

```
/PRO 1
TAKE _
  PRESENT ?    ; ¿Está presente el sustantivo?
  WEIGHT ? 10  ; ¿Pesa menos de 10?
  GET ?        ; Tomar objeto
  MESSAGE 10   ; "Taken"
  DONE         ; Terminar

LOOK _
  DESC         ; Describir localización
  DONE

INVENTORY _
```



```

    INVEN          ; Mostrar inventario
    DONE

QUIT _
    MESSAGE 20     ; "Are you sure?"
    PARSE         ; Obtener respuesta
    YES           ; ¿Es afirmativa?
    END           ; Terminar juego
    RESTART       ; Si no, continuar

```

### Ejemplo 3: Proceso con Condiciones Complejas

```

/PRO 2
OPEN DOOR
    CARRIED KEY    ; ¿Lleva la llave?
    AT 5           ; ¿Está en localización 5?
    HASNAT DOOR OPEN ; ¿La puerta NO está abierta?
    SETAT DOOR OPEN ; Marcar puerta como abierta
    MESSAGE 30     ; "The door creaks open"
    DONE

CLOSE DOOR
    AT 5           ; ¿Está en localización 5?
    HASAT DOOR OPEN ; ¿La puerta está abierta?
    HASNAT DOOR OPEN ; Marcar puerta como cerrada
    MESSAGE 31     ; "You close the door"
    DONE

```

### Ejemplo 4: Uso de Indirección

```

/PRO 3
USE _
    CARRIED ?      ; ¿Lleva el objeto?
    COPYOF ? 100   ; Copiar número de objeto a bandera 100
    EQ 100 SWORD   ; ¿Es la espada?
    PROCESS 10     ; Procesar uso de espada
    EQ 100 KEY     ; ¿Es la llave?
    PROCESS 11     ; Procesar uso de llave
    MESSAGE 40     ; "You can't use that"
    DONE

```

### Ejemplo 5: Bucle con DOALL

```

/PRO 4
LOOK _
    DESC          ; Describir localización

```

```
DOALL 255      ; Para cada objeto en localización actual
PROCESS 5      ; Procesar descripción de objeto
DONE

/PRO 5

--
CARRIED ?      ; ¿Está el objeto siendo llevado?
SKIP 2         ; Si sí, saltar descripción
MESSAGE [?]    ; Mostrar descripción del objeto (indirecto)
NEWLINE       ; Nueva línea
DONE
```

---

## Conclusión

El formato SCE de DAAD es un sistema potente y flexible para crear aventuras conversacionales. Su **sistema de procesos** basado en tablas de verbo-sustantivo con CondActs proporciona un control fino sobre la lógica del juego, mientras que su estructura modular permite una organización clara y mantenible del código fuente.

La clave para dominar DAAD está en comprender cómo funcionan los procesos y cómo se relacionan entre sí a través de las llamadas **PROCESS**, el bucle principal en el Proceso 0, y el flujo de ejecución de condiciones y acciones dentro de cada entrada de proceso.

---

*Este documento está basado en el Manual DAAD Version 2 Release 1 y proporciona una referencia completa del formato de archivo SCE y el funcionamiento del sistema de procesos.*