
OPEN GAME DEFINITIONS

STANDARD v0.1.e1

By Arda Erdikmen, Sep 2009

DRAFT

Table Of Contents

Overview	2
Goals	2
Terminology	2
Structure	2
GDS File type and GDS data	2
Types	3
Root element: GameDefinitionStandard	3
GameDefinitionStandard > GameDefinition	3
GameDefinitionStandard > GameDefinition > Platform	4
GameDefinitionStandard > GameDefinition > Platform > GameFileName	4
Mapping Controls (GameDefinitionStandard > GameDefinition > Platform >)	5
ALT Root tag	5
ALT > MAP tag	5
ALT > MAP > ASSIGN Sub tag	6
ALT > MAP > SWITCH sub tag	6
MACRO Operations	7
ALT > MAP > HIT Sub tag	7
ALT > MAP > WAIT sub tag	7
ALT > MAP > HOLD Sub tag	8
ALT > MAP > RELEASE Sub tag	8
ALT > MAP > LOOP element	8
ALT > MAP > SETUP ELEMENT	8
ALT > MAP > GROUP element	9
KEY Constants	10
Basic Controls	10
Additional Buttons	10
Virtual Keys	11

Planned Virtual Keys	11
Emulator functions	12
Game Information Group	12
Game Information	12
Game Information > Name (BoundedString, optional)	13
Game Information > Description(BoundedString, Optional)	13
Game Information > ReleaseDate(BoundedString, optional)	13
Game Information > Version(TwoPartNumber, optional)	13
Game Information > Developers / Publishers.....	13
Game Information > People.....	13
Example GDS Files	13
Minimalistic Example	13
Detailed Example.....	14
A GDS Example with Multiple Game Entries.....	15

Overview

Game definition standard (GDS) is a standard for creating key and button definitions for emulators. When you use an emulator on a console or a cell phone, there is always a problem about controls. Even on a PC with a hundred keys on the keyboard, setting up controls remains as a problem. You have to configure host machine's control buttons, keys or joysticks to emulated machine's accessories. Even this might not be enough, because not every game uses same way to control a character, navigate into menus or start the game.

GDS provides convenient method for generalize controls over different host platforms and on different emulated software by using XML engine.

Goals

****to do****

Terminology

****to do****

Structure

GDS File type and GDS data

GDS, "Game Definition Standard" is not only a key assignment library, but it's also stores comprehensive information about a game.

GDS files must conform to XML 1.0 standard.

A GDS file may contain one or more games in it. Common usage of the GDS files like this:

```
It_came_from_desert.tzx
It_came_from_desert.gds
```

You provide a GDS file for every game image in the folder. Yet you can merge all those files to form a single big GDS file.

If you want to embed a GDS info into a TZX file, please use this guide to create a Custom info block:

Tzx Custom Info Block (id:35):

Offset 0- CHAR:"GDS"

Offset 3-DWORD:GDS data length(4 Byte)

Offset 7-CHAR:UTF8 encoded XML formatted GDS data

Types

****to do****

Type	Description
TwoPartNumber	A simple type contains version information eg. "1.2" (0-65535.0-65535)
path	Windows based path string.
MD5	128 bit MD5 hash.
BoundedString	A string which preserves white space and max. 32768 chars long.
BASE64	Base64 encoded data.

Root element: GameDefinitionStandard

GameDefinitionStandard, is the root element and it contains one or more game definitions.

```
< GameDefinitionStandard>
  ...One or more game definitions...
</ GameDefinitionStandard>
```

GameDefinitionStandard > GameDefinition

This is the main element that holds all the data about a game and key bindings. This element has five attributes:

GameHASH , GameID, GameFileName , Version

Name	Type	Mandatory	Description
GameHASH	MD5	No	A MD5 hash of the game file created by a generator. If this attribute specified, this definition entry will be kept separate from the game group. If omitted, same game from different platforms may be merged

			together under the same GameDefinition element with a couple of platform element.
GameID	Bounded String	No	GameID is uniquely identifies your game to the emulator. It may be taken from a big database, or may be pre-set by the creator of the game.
Version	TwoPart Number	Yes	GDS definition version for this entry. It starts from 0.1 as of this document.

Example

```
<GameDefinition GameHASH=" 30b7ed5304dc0167076823434f43d7b1" GameID="101104",
Version="0.1">
```

GameDefinitionStandard > GameDefinition > Platform

This element specifies the definition file is specially to be played on a specific emulated machine.

It has a set of optional sub items with additional attributes: "Hardware", "Model", "Peripherals", "Version".

Example

```
<Platform Brand="Sinclair" Model="ZX Spectrum + Peripherals="Multiface 3"
Version="Issue 3B">
```

... Definitions go here...

```
</Platform>
```

GameDefinitionStandard > GameDefinition > Platform > GameFileName

This attribute may be required to identify game to the emulator, yet may be ignored by it. This attribute is not only recommended, it's mandatory. GDS files may be merged together to form a big definition file, so this entry separates every entry. A Platform element may include more than one GameFileName element under it.

Example

```
<Platform Brand="Sinclair" Model="ZX Spectrum + Peripherals="Multiface 3"
Version="Issue 3B">
```

```
  <GameFileName>somegame.tzx</Gamefilename>
  <GameFileName>somegame.z80</Gamefilename>
  <GameFileName>somegame.sna</Gamefilename>
  <GameFileName>somegame.szx</Gamefilename>
```

... Definitions go here

```
</Platform>
```

If no hardware element is given, the definitions are no bound to match a specific brand or model.

When merging different GDS files into a single big GDS file, games from different platforms may or may not be merged together under a <gamedefinition> tag.

Mapping Controls (GameDefinitionStandard > GameDefinition > Platform >)

ALT Root tag

GDS supports different setups and ways to cycle between them. ALT tag sets a new set of controls to each host button. Every mapping set starts with an "ALT" tag. Attributes follows.

Name	Type	Mandatory	Description
ID	String	Yes	A unique id for the button set.
Description	String	No	A description of the set of keys.

Examples

```
<ALT id="main" description="Movement">
  <MAP button="B_FIRE" description="Fire Gun">
    <ASSIGN key=VK_SPACE/>
  </MAP>
</ALT>
<ALT id="inv" description="inventory keys">
  <MAP button="B_FIRE" description="Pickup Item">
    <ASSIGN key=VK_T/>
  </MAP>
</ALT>
```

ALT > MAP tag

Map tag, assigns a virtual key or a macro to a host button.

Name	Type	Mandatory	Description
Button	Button	Yes	Host button to assign
Modal	Bool	No	Modal attribute locks the input buttons until macro is completed. Until macro completes, user cannot interact with the emulator. Modal macros can lock the emulator if all of the buttons on the host machine are controlled by GDS system.
Description	String	No	Assigns a description to assigned key. If you don't provide a description, default one will be used. Default descriptions are the same as the element name but without the "B_" prefix. Eg. "B_START"'s default description is "START", "B_UP"'s is "UP"...

Examples

```
<MAP button="B_UP">
```

```
...mapped virtual key assignments and/or macros...
</MAP>
```

```
<MAP> button="B_FIRE1" Description="Smart Bomb">
...assignments...
</MAP>
```

```
<MAP> button="B_FIRE1" Description="Hadouken Combo" modal="True">
...a macro...
</MAP>
```

ALT > MAP > ASSIGN Sub tag

Copies the host button behavior to virtual key. If user presses the button for 1 sec, Virtual key will be pressed for exactly 1 second. This is the most basic assignment command. "Assign" generally is not used in the macros as it creates an interactive assignment and may produce unexpected results.

Name	Type	Mandatory	Description
Key	Vkey	Yes	Virtual key to be assigned

Example

```
<MAP button="B_UP">
  <ASSIGN key="VK_Q"/>
</MAP>
```

ALT > MAP > SWITCH sub tag

Switches settings between alternate setups. Switch is a way to assign a button to jump to a different "ALT".

Name	Type	Mandatory	Description
Button	Button	Yes	Host Button to assign
Alt	String	Yes	Alternate Location ID
Clear	Bool	No	Clear attribute clears all assignments to GDS_DUMMY before making a switch. Default setting for this attribute is "False". So assignments will be carried to alternate setups if they are not set in the alternatives.

Examples

```
<ALT id="main" description="Movement">
  <MAP button="B_FIRE" description="Fire Gun">
    <SWITCH button="B_SELECT" alt="inv"/>
    <ASSIGN key="VK_SPACE"/>
  </MAP>
</ALT>
```

```

<ALT id="inv" description="inventory keys">
  <MAP button="B_FIRE" description="Pickup Item">
    <SWITCH key="B_SELECT" alt="main"/>
    <ASSIGN key="VK_T"/>
  </MAP>
</ALT>

```

MACRO Operations

ALT > MAP > HIT Sub tag

If you are trying to create a combo or macro script, HIT makes the button pressed and released in N t-state. Even if player press&hold the button on the host machine, Virtual key will be released after given time. Default unit is 1ts.

Name	Type	Mandatory	Description
Key	Vkey	Yes	Virtual key to be assigned
Seconds	Twopartnumber	No	Seconds with dot separated fraction eg."0.035"
Tstate	UnsignedInteger	No	T-states for exact timing.
Frame	UnsignedInteger	No	Depends on the emulated machine.

Examples

```

<MAP button="B_UP">
  <HIT key="VK_SPACE"/>
</MAP>

```

```

<MAP button="B_UP" Description="Fly for 1.5 seconds">
  <HIT key="VK_SPACE" Seconds="1.5"/>
</MAP>

```

ALT > MAP > WAIT sub tag

In a macro or combo, waits for a given amount of time.

Name	Type	Mandatory	Description
Seconds	Twopartnumber	No	Seconds with dot separated fraction eg."0.035"
Tstates	UnsignedInteger	No	T-states for exact timing.
Frames	UnsignedInteger	No	Depends on the emulated machine.

Example

```

<MAP button="B_UP" Modal="True">
  <HIT key="VK_Q"/>
  <WAIT seconds="1"/>
  <HIT key="VK_U"/>
</MAP>

```

ALT > MAP > HOLD Sub tag

Makes the assigned key to stick until a hit or release tag occurred in the script.

Name	Type	Mandatory	Description
Key	Vkey	Yes	Virtual key to HOLD

Example

```
<MAP button="B_UP">
  <HOLD key="VK_Q"/>
</MAP>
```

ALT > MAP > RELEASE Sub tag

Makes the assigned key to release from a hold instruction.

Name	Type	Mandatory	Description
Key	Vkey	Yes	Virtual key to be released.

Example

```
<MAP button="B_UP">
  <HOLD button="VK_Q"/>
  <WAIT frames="5"/>
  <RELEASE key="VK_Q"/>
</MAP>
```

ALT > MAP > LOOP element

Loops can be set to assign multiple set of keys to a button. LOOP has no attributes.

Examples

```
<MAP button="B_START" Description="Press 1,2,3 in succession">
  <LOOP>
    <ASSIGN key="VK_1"/>
    <ASSIGN key="VK_2"/>
    <ASSIGN key="VK_3"/>
  </LOOP>
</MAP>
```

When you press Start button first time, it simulates the "1" key, when you tap it again it presses "2" this time. It will loop through the list forever.

ALT > MAP > SETUP ELEMENT

Setup can be set to assign multiple set of keys to a button. Unlike LOOP, setup assigns only run once.

Name	Type	Mandatory	Description
Reset	Bool	No	Specifies if Setup is performed again when alternate setup is selected. If you omit the RESET attribute, setup sequence will only run once through the entire game session.

Examples

```
<MAP button="B_START" Description="Press 1,2,3 in succession">
  <SETUP>
    <ASSIGN key="VK_1"/>
    <ASSIGN key="VK_2"/>
    <ASSIGN key="VK_3"/>
  </SETUP>
</MAP>
```

When you press Start button first time, it simulates the "1" key, when you tap it again it presses "2" this time. And at the third or more times it will keep pressing "3" key. it will never press 1 or 2 again, until a SWITCH executed *and* RESET attribute is set.

ALT > MAP > GROUP element

Creates groups of assignments in MAP tag. Useful when creating LOOPing buttons and SETUPS as it also has a description attribute.

Name	Type	Mandatory	Description
Description	String	No	Replaces the description string in MAP tag.

```
<MAP button="B_START" Description="Press 1,2,3 in succession">
  <LOOP>
    <GROUP description="Build up energy">
      <HOLD button="VK_Q"/>
      <WAIT frames="5"/>
      <RELEASE button="VK_Q/>
    </GROUP>

    <GROUP>
      <ASSIGN key="VK_T"/>
    </GROUP>

    <GROUP description="Heal targets">
      <HOLD key="VK_SYMBOLSHIFT">
      <HOLD key="VK_CAPS">
      <Hit button="VK_H"/>
      <RELEASE key="VK_SYMBOLSHIFT">
      <RELEASE key="VK_CAPS">
    </GROUP>
  </LOOP>
</MAP>
```

KEY Constants

Basic Controls

The basic controls include a directional pad, two action buttons, a select and a start button.

Name	Recommended Usage
B_SELECT	Switching between alternate configurations, options, menus or pause.
B_START	To start a game. Provide a macro if needed.
B_UP	Cursor/Character Movement
B_DOWN	Cursor/Character Movement
B_LEFT	Cursor/Character Movement
B_RIGHT	Cursor/Character Movement
B_FIRE1	Main Action Button
B_FIRE2	Secondary action button

Don't try to use other buttons if possible. Do not use other directional buttons specific to niche devices, leave this to emulator writers.

Additional Buttons

Use additional buttons if the game only and ultimately requires more buttons, even in this case, try to fit most basic functions to above keys.

B_FIRE3...31
B_LTHUMB1
B_LTHUMB2
B_LTHUMB3
B_RTHUMB1
B_RTHUMB2
B_RTHUMB3

CELL PHONE SPECIFIC CONTROLS

B_YES
B_NO
B_CPAD_UP
B_CPAD_DOWN
B_CPAD_LEFT
B_CPAD_RIGHT
B_MENU

THE DEVICES WITH KEYBOARD

B_KEY_A...Z
B_KEY_0...9

Virtual Keys

This keys represent the emulated key, for example, a key press event on a emulated machine. There may be different keys in different machines, e.g. Apple key on an old Macintosh computer. Or set of control keys on an atari800XL or c64 Platform. Those keys will be added in later revisions of this document.

As first version intended to be used in Zx Spectrum Platform Emulators, only special spectrum keys are listed here:

Name	Description
VK_0..9	Numbers on keyboard above the letters, not the numeric part of the keyboard.
VK_A..Z	Keys of the keyboard, A to Z (ANSI Characters only)
VK_CAPSSHIFT	Caps Shift key on ZX Spectrum
VK_SYMBOLSHIFT	Symbol Shift key on ZX Spectrum
VK_BREAK	Break Key on Zx Spectrum
VK_ENTER	Enter Key on Zx Spectrum
VK_KEYCODE_0...255	Keyboard input codes on emulated machine. Those constants are for special keys, not listed on this listing.
VK_JOY1_UP	Joystick PORT 1 on emulated machine (Sinclair in this case)
VK_JOY1_DOWN	"
VK_JOY1_LEFT	"
VK_JOY1_RIGHT	"
VK_JOY1_FIRE1	"
VK_JOY1_FIRE2	"
VK_JOY2_UP	Joystick PORT 2 on emulated machine (Sinclair in this case)
VK_JOY2_DOWN	"
VK_JOY2_LEFT	"
VK_JOY2_RIGHT	"
VK_JOY2_FIRE1	"
VK_JOY2_FIRE2	"
VK_KEMPSTON_UP	Kempston Joystick on a ZX Spectrum
VK_KEMPSTON_DOWN	"
VK_KEMPSTON_LEFT	"
VK_KEMPSTON_RIGHT	"
VK_KEMPSTON_FIRE1	"
VK_KEMPSTON_FIRE2	"
VK_KEMPSTON_FIRE3	"
VK_MOUSE_X	Generic Mouse on a emulated machine X axis
VK_MOUSE_Y	", Y axis
VK_MOUSE_L	Left mouse button
VK_MOUSE_R	Right mouse button
VK_MOUSE_M	Middle mouse button

Planned Virtual Keys

Following keys are not need to be supported right now, but it will be added to the later revisions of GDS format. You can use them if you need any.

VK_SHIFT	Generic Shift key
VK_F1..F12	Generic F-keys

VK_START	Start button on emulated machine
VK_SELECT	Select button on a emulated machine
VK_RESET	Reset button on emulated machine
VK_OPTION	Option button on emulated machine
VK_HELP	Help button on emulated machine
VK_ESC	Esc button on emulated machine
VK_CAPSLOCK	Caps Lock button on emulated machine

Emulator functions

These functions may not be supported by all emulators!

Name	Description
GDS_QUICKSAVE	Makes emulator to make a quick save operation
GDS_QUICKLOAD	Makes emulator to make a quick load operation
GDS_SAVE	Makes emulator to execute a save operation
GDS_LOAD	Makes emulator to execute a save operation
GDS_OPTIONS	Makes emulator to show it's options dialog
GDS_ESC	Makes emulator to performs an ESC action (eg. While in menu's)
GDS_OK	Accept button for dialogues on emulator
GDS_CANCEL	Cancel button for dialogues on emulator
GDS_SELECT	Select button on emulator
GDS_BACK	Makes emulator to performs an BACK action (eg. While in menu's)
GDS_QUIT	Makes emulator to quit.
GDS_RESET	Makes emulator to perform a reset.
GDS_VIEW_DESCRIPTIONS	Makes emulator to show key descriptions on screen.
GDS_DUMMY	A dummy keyword. Does nothing.
GDS_ASK*	Makes emulator to ask a question to user.
GDS_EVALUATE*	Makes emulator to perform an evaluation.

*These functions are not yet defined.

Game Information Group

While not comprehensive, Open Game Definition Standard supports to include basic information about a game.

Game Information

This group's root element is <GameInformation>.

Example

```
<GameInformation>
  ...informational elements...
</ GameInformation>
```

Game Information > Name (BoundedString, optional)

Declares game's official name. while this element is also optional, declaring a name for a game is recommended.

```
<Name>Chuckie Egg</Name>
```

Game Information > Description(BoundedString, Optional)

A description about a game.

```
<Description>Fast paced platformer</Description>
```

Game Information > ReleaseDate(BoundedString, optional)

```
<ReleaseDate>2006-08-07</ReleaseDate>
```

Game Information > Version(TwoPartNumber, optional)

Specifies the version of the defined game.

```
<Version>1.0 </Version>
```

Game Information > Developers / Publishers

These two tags may have URLs to respective sites.

```
<Developers>  
<Developer URL="HTTP://www.somesite.com" >SomeSoftware House</Developer>  
</Developers>
```

Game Information > People

The people who worked on the game.

The sub elements are:

Code, Additional Code, Graphics, Sound, Music, Concept, Design

```
<People>  
  <Code>James Cook</Code>  
  <Code>Matt Winston</Code>  
  <Design>Rosetta Rock</Design>  
  <Music>Eth Baltaci</Music>  
  <Graphics>Rim Kavcar</Graphics>  
</People>
```

Example GDS Files

Minimalistic Example

```

<GameDefinitionFile>
  <GameDefinition Version="0.1">
    <Platform>
      <GameFileName>somegame.tzx</GameFileName>
      <ControlAssignment>
        <ALT ID="1">
          <MAP button="B_UP" key="VK_Q"/>
          <MAP button="B_DOWN" key="VK_A"/>
          <MAP button="B_LEFT" key="VK_O"/>
          <MAP button="B_RIGHT" key="VK_P"/>
          <MAP button="B_FIRE1" key="VK_Q"/>
          <MAP button="B_START" key="VK_S"/>
          <MAP button="B_SELECT" key="VK_H"/>
        </ALT>
      </ControlAssignment>
    </Platform>
  </GameDefinition>
</GameDefinitionFile>

```

Detailed Example

```

<GameDefinitionFile>
  <GameDefinition GameHASH=" 30b7ed5304dc0167076823434f43d7b1" GameID="101104",
Version="0.1">
    <Platform Brand="Sinclair" Model="Zx Spectrum+">

      <GameFileName>somegame.tzx</GameFileName>
      <GameFileName>somegamewithdifferentname.sna</GameFileName>

      <GameInformation>

        <Name>Some Game</Name>
        <Description>Fast paced platformer</Description>
        <ReleaseDate>2006-08-07</ReleaseDate>
        <Version>1.0 </Version>
        <Developers>
          <Developer URL="HTTP://www.somesite.com" >SomeSoftware
House</Developer>
        </Developers>
        <Publishers>
          < Publisher URL="HTTP://www.somesite.com" >SomeSoftware
House</ Publishers>
        </Publishers>
        <People>
          <Code>James Cook</Code>
          <Code>Matt Winston</Code>
          <Design>Rosetta Rock</Design>
          <Music>Eth Baltaci</Music>
          <Graphics>Rim Kavcar</Graphics>
        </People>

```

```

</GameInformation>

<ControlAssignment>
  <ALT ID="1">
    <MAP button="B_UP" key="VK_Q"/>
    <MAP button="B_DOWN" key="VK_A"/>
    <MAP button="B_LEFT" key="VK_O"/>
    <MAP button="B_RIGHT" key="VK_P"/>
    <MAP button="B_FIRE1" key="VK_Q"/>
    <MAP button="B_START" key="VK_S"/>
    <MAP button="B_SELECT" key="VK_H"/>
  </ALT>
</ControlAssignment>
</Platform>
</GameDefinition>

</GameDefinitionFile>

```

A GDS Example with Multiple Game Entries

(Note that Game Definitions from different versions of GDS can be used together in a GDS container.)

```

<GameDefinitionFile>

  <GameDefinition GameFileName="Somegame.tzx" Version="0.2">
    <Platform>
      <GameFileName>somegame.tzx</GameFileName>
      <ControlAssignment>
        <ALT ID="1">
          <MAP button="B_UP" key="VK_Q"/>
          <MAP button="B_DOWN" key="VK_A"/>
          <MAP button="B_LEFT" key="VK_O"/>
          <MAP button="B_RIGHT" key="VK_P"/>
          <MAP button="B_FIRE1" key="VK_Q"/>
          <MAP button="B_START" key="VK_S"/>
          <MAP button="B_SELECT" key="VK_H"/>
        </ALT>
      </ControlAssignment>
    </Platform>
  </GameDefinition>

  <GameDefinition Version="0.1">
    <Platform>
      <GameFileName>anothergame.tzx</GameFileName>
      <ControlAssignment>
        <ALT ID="1">
          <MAP button="B_UP" key="VK_Q"/>
          <MAP button="B_DOWN" key="VK_A"/>
          <MAP button="B_LEFT" key="VK_O"/>
          <MAP button="B_RIGHT" key="VK_P"/>
        </ALT>
      </ControlAssignment>
    </Platform>
  </GameDefinition>

```

```
<MAP button="B_FIRE1" key="VK_Q"/>
<MAP button="B_START" key="VK_S"/>
<MAP button="B_SELECT" key="VK_H"/>
</ALT>
</ControlAssignment>
</Platform>
</GameDefinition>

</GameDefinitionFile>
```