

WOLF RPG Editor User Manual 2.24Z

English Translation by
Verella Himmel (vgperson)

Edited by
Jeffrey Casey (Widderune)

*** The user manual has been updated,
but some of the website links are nonexistent and have been marked.**

[Update Log](#)

[Frequently Asked Questions](#)

[Opening / Usage Agreement](#)

[Features and Functions](#)

[File Explanations](#)

[Basic Controls](#)

[Debug Command List](#)

[Main Window Operation](#)

[Basic Game Settings](#)

[Create Game Data](#)

[Tileset Settings](#)

[Map Settings](#)

[Common Event Settings](#)

[Map Select](#)

[Map Creation Flow](#)

[1. Creating/Registering New Map](#)

[2. Set Start Position](#)

[Registering Positions](#)

[Auto-Create Map Function](#)

[Creating Events](#)

[Event Window](#)

[Move Route](#)

[File Select](#)

[Formula Explanation / Variable Names](#)

[Event Command List](#)

[Show Message](#)

[Show Choice](#)

[Set Variable](#)

[DB Management](#)

[Set Variable+](#)

[Set String](#)

[Condition \(Number\)](#)

[Condition \(String\)](#)

[Input Key](#)

[Picture](#)

[Effect](#)

[Sound](#)

[Save/Load Control](#)

[Party Graphics](#)

[Chip Management](#)

[Transfer](#)

[Event Control](#)

[Common Event](#)

[Download](#)

[Databases](#)

[Type Definition](#)

[System Database](#)

[Type Explanations](#)

[User Database](#)

[Variable Database](#)

[Resource Standards](#)

[System Variable/String List](#)

[Special Characters Reference](#)

[Variable Reference List](#)

[Helpful Tools](#)

[WOLF RPG Editor Manual]

[On the Usage of This Text]

Copyright on this manual belongs to the author, SmokingWOLF.
However, the text may be freely cited for purposes such as making explanations.

[Update Log]

06/22/2018

- Added missing parts to the System Variable List.

06/02/2018 Updated for Editor Ver. 2.22.

- Updated Game Settings. Added to "Basic Game Settings." "Config" details made up to date.
- Added to Command "Set Variable+".
- Added to Resource Standards (font-related).
- Updated System Variable/String List.

01/30/2018

- Event Command "Input Key": Take Input said nothing about mouse processing, so added a section.

11/22/2016 Updated for Editor Ver. 2.20.

- Updated System Variable/String List.
- Updated System Database Type Explanations.
- Updated Game Settings.
- Updated Common Event Settings.
- Added other missing notes about behaviors.

07/23/2013 Updated for Editor Ver. 2.10.

- Added Frequently Asked Questions 13 and 14.
- Updated Game Settings, Formula Explanation, and System Variable/String List.
- Edited event commands "Set Variable+", "DB Management," and "Picture."
- Added new event command "Download."
- Note in "Resource Planning" about map chips not displaying in some cases.

05/07/2012

- Added missing descriptions.

03/21/2012 Updated for Editor Ver. 2.02.

- Added description for #42 in System Variables.
- Additional explanations in "Event Window," "Set Variable," "Transfer," and "Picture."

11/17/2011 Fixed missing images and broken links.

07/19/2011 Updated for Editor Ver. 2.00.

- Changes were made with major assistance from K:W!, Pipo, and Tochi. Thank you very much.

09/25/2010 Updated for Editor Ver. 1.31.

- Added more key code information to event command [Event Control].
- Other small fixes.

07/02/2010 Updated for Editor Ver. 1.30.

- Replaced Event Window with new version.
- Replaced Tileset Settings with new version.
- Other fixes to outdated information.

04/29/2010 Updated for Editor Ver. 1.20.

- Added explanation of Event Window event conditions.
- Added function explanations to Basic Game Settings.
- Additional explanations in Basic Controls and event commands [Set Variable+] and [Other 2].

02/08/2010 Updated for Editor Ver. 1.16.

- Added "CSelf Var Usage List" to Common Event Window operation.
- Added "\A+" "\A-" "\mx[?]" and "\my[?]" to Special Characters.
- Updated Frequently Asked Questions.

11/18/2009 Updated for Editor Ver. 1.15.

- Added "Off + 2x" to font anti-aliasing in Basic Game Settings.
- Added "Find" to Common Event Window operation.
- Added three new operators to "Set Variable."
- Added "String Done Displaying?" for Picture in "Set Variable+."

09/25/2009 Updated for Editor Ver. 1.14.

- Added "Undo" to Common Event Window operation.
- Fixed other small mistakes.

07/29/2009 Added an entry to Frequently Asked Questions.

07/22/2009 Updated for Editor Ver. 1.13.

- Updated "Map Select" in Main Window operation.
- Added explanation about MIDI resources to Resource Standards.
- Updated event commands "Set Variable+" and "DB Management."

06/02/2009 Updated for Editor Ver. 1.12.

- Updated "Map Select" in Main Window operation.
- Updated event commands "Set Variable+," "Transfer," and "Event Control."

04/25/2009 Updated for Editor Ver. 1.11.

- Replaced Map Settings window.
- Updated minor details about "Other 2" and "Set Variable+."

02/26/2009

- Added "File Select" explanation.

02/25/2009

- Added method for reading external font files to Basic Game Settings font explanations.
- Minor updates now that MP3s are usable for sounds.
- Noted method of using Ctrl+Left/Right Click to copy and paste in Main Window operation.

02/17/2009 Updated for Editor Ver. 1.10.

- Added explanation of "Voice Synthesis."
- Added explanation of calling by name to "DB Management."
- Updated explanation for receiving key input in "Other 1."
- Added explanation in "Set String" for "Replace Left Side" option.
- Added explanation of "Turn Party Following On" in "Party Graphics."
- Fixed other small mistakes.

01/13/2009 Updated for Editor Ver. 1.09.

- Added new "File Explanation" entry.
- Added new "Frequently Asked Questions" entry.
- Added "Remove Instances Of String..." function to "Set String."
- Added priority explanation to "Special Characters Reference."

01/04/2009

- Added explanation about String Variable conversion to "Set Variable."
- Added explanation of calling Common Events by name to "Other 2."
- Added explanation of specifying filename to "Sound."
- Added background/fog graphic filename, BGM/BGS filename, and choice contents to "System Variable/String List."
- Added explanations on getting state of Map Events to "Set Variable+."

12/06/2008 Updated for Editor Ver. 1.08.

- Added explanation of BGM/BGS/SE volume adjustment and X/Y scroll to "System Variable/String List."
- Added "Picture" explanations for [5] Simple Window (File Specified by String Variable), linking to scroll, and pictures -100000 or lower.
- Added explanation of specifying move target by variable to "Transfer."
- Added getting map size to "Set Variable+."
- Added thanks to "Auto-Generate Map Function."

11/17/2008

- Added explanation of getting data count/contents and initialization to "DB Management."
- Added chart mapping numbers to keyboard input to "Other 1." (Image Credit: Mutsukarasu)
- Provided missing explanations to "System Variable/String List."

10/27/2008

- Added explanation of "On Screen? (1=YES 0=NO)" to "Set Variable+" (for Editor Ver. 1.07).

10/25/2008

- Added explanations to "Chip Management" and "Effect" (for Editor Ver. 1.07).
- Added function key explanations to "Basic Controls."

10/24/2008

- Added character move speed adjustment to "Basic Game Settings."
- Added codes for using database entry names to "Special Character Reference."
- Updated other parts that work differently now.
- "Basic Controls" "Main Window" "Common Event Settings" "Picture" "System Variable/String List" "Type Definition" (Changes by: Rew)

09/20/2008

- Added "Auto-Generate Map Function."
- Added "Graphic Synthesizer" to list of tools.
- Added System String explanations to "System Variable/String List" (formerly System Variable List).

07/27/2008

- In accordance with version update, removed "Uses MIDI" from "Basic Game Settings."
- Fixed missing parts of "System Variable List."

07/12/2008

- Added new functions, and updated insufficient explanations: "Resource Standards," "Special Character Reference."

07/01/2008

- Revised command processes I forgot to edit: Event commands "Picture," "Sound," "Set String," and "Resource Standards."

06/04/2008 Changes made for Ver. 1.02.

[Frequently Asked Questions]

This section addresses common situations and questions encountered by those using WOLF RPG Editor for the first time.

<List> (Click to jump to that answer)

- [1] [Only half of the character displays](#)
- [2] [I set "Starting Hero Graphic" in Basic Game Settings, but it is not updating \[*Base System \(Sample Game\)\]](#)
- [3] [How do I display face graphics?](#)
- [4] [There is an event I cannot talk to in the Sample Game](#)
- [5] [I set up multiple pages in an event, but only the last page runs](#)
- [6] [\(Just after starting the game\) The position of displayed text is off](#)
- [7] [It is tedious not being able to copy Common Events](#)
- [8] [I want to erase an event after it activates once](#)
- [9] [When editing maps, transparent parts turn a strange color](#)
- [10] [The Variable DB looks fine, but I get "reading a non-existent entry" or "attempted to store a number in a place for a string" errors, and the data I changed is not showing up](#)
- [11] [When using "linked to scroll" background, the background shifts when you loop](#)
- [12] [My event will not start](#)
- [13] [When using an encrypted folder with Set String "Get File List of Folder...", I get <<ERROR>>](#)
- [14] [When using Move Route, the first action keeps repeating / the first action executes 2 or more times.](#)
- [15] [I want to display 2x-size pixels like with 320x240, but at higher resolutions.](#)

For anything not listed here, check the [Question Summary Thread \[Japanese\]](#) on the Wiki (It will require an internet connection).

[1] Only half of the character displays...



<== Looks like this.

From the top menu of the Editor, go to "Game Settings," then "Open Game Settings."
Change "Character Sprite Direction Type" to "4-Directional," and it should be fixed.
By default, it is set to use graphics with 8 directions, so if you use RPG Maker-standard resources or other 4-directional sprite sheets, you will only be able to see the left or right half of the character.

[2] I set "Starting Hero Graphic" in Basic Game Settings, but it is not updating.
[*When using the Base System (Sample Game Engine)]

When you use the Base System, even if you set "**Starting Hero Graphic**" in **Basic Game Settings**, it will not be reflected in-game. This is because after the game starts, the Base System runs a process to **automatically change character graphics to the initial party defined in Variable DB Type 6**.
Thus, the graphic ends up being from this "initial party" rather than the "Starting Hero Graphic."

Depending on what you want, follow one of the following methods.

"I want to display the hero with specific graphics at game start."

1. In Variable DB Type 0 "Hero Status," create the character you want to appear in any data slot you like.
Set the "**Walk Graphics Image**" field to their walking graphics.
2. In Variable DB Type 6 "Party Info," for Data #0 ("Member 1"; 2, 3, etc. are also fine),
select the character you defined in step 1.
3. Start the game.
If the Base System works as intended, the character should have the graphics you specified.

"I want the hero to be invisible at game start."

If you just want the heroes to be invisible, then right after the game starts (such as in an auto-starting event), simply use the Special function of event command "Party Graphics," "Make Party Transparent."

[3] How do I display face graphics?

There is a built-in face graphics system as of Base System (Sample Game Engine) Ver. 1.26. It is used as follows.

1. Open System Database Type 24: Face Graphics, and in Data #2 and on set up the face graphics you want with "Face Graphic File."
2. In the first line of a "Show Message," use the notation "@2" to change face graphics. (An @ followed by the number of the face to use)

*** Face Graphic #0 displays text without a message window, and #1 displays text with only a message window and no face.**

Example: Displaying Text With Face Graphic 2 in "Show Message"

@2
Hello, I am Wolfarl.

[4] There is an event I cannot talk to in the Sample Game.

Have you changed "Default Character Move" in "Basic Game Settings" from "0.5 Tile" to "1 Tile" When you move 1 tile at a time, it may make some event characters impossible to talk to. In this case, try changing the step amount to "0.5 Tile."

[5] I set up multiple pages in an event, but only the last page runs.

Map Events execute giving priority to **higher-number pages**. As a result, if the conditions are met for the last page, that one will be executed rather than the smaller-number pages.

[6] (Just after starting the game) The position of displayed text is off.

Try adding a 1-or-more-frame wait right before the text. Fixing it may be as simple as that.

The display position of text is set on frame 1 of the parallel Common Event 63, so text displayed before this runs will not match the window position.

[7] It is tedious not being able to copy Common Events.

After selecting a Common Event from the list on the left side, it can be copied by pressing C, cut by pressing X, or pasted by pressing V. C, X, and V can be used to Copy, Cut, and Paste in other places as well.

[8] I want to erase an event after it activates once.

You can erase events using "**Erase Event**" in event command "**Event Control**".
Another method is using variables to make "events that only happen once."

<Wolf Editor Perfect Guide> Making a Chest That Stays Open When Opened
http://www.silversecond.com/WolfRPGEditor/Guide/EV_006.html (Japanese)

[9] When editing maps, transparent parts turn a strange color.

As an intended function of the Editor, if a transparent color is not black, those parts are displayed as if with a translucent film over them.

Do not worry, they will display as normal in the game itself.



[10] The Variable DB looks fine, but I get "reading a non-existent entry" or "attempted to store a number in a place for a string" errors, and the data I changed is not showing up.

Try starting from a New Game, and it may fix the issues.

The Type info and Data of the Variable DB is transferred into the Save Data at game start.

So, if you make changes to the Variable DB's Type Settings or Data in the Editor, and then continue from old Save Data, it will try to load from a DB unaffected by these changes, which can result in data counts not matching up and data being out of date.

Note that the User DB and System DB do not depend on Save Data, so problems of this sort will not happen with them.

[11] When using "linked to scroll" background, the background shifts when you loop.

This is intended behavior. Supposing a 320x240 background, it should work if you make the map's width a multiple of 20 and the height a multiple of 15.

The rule is: **"When a "not moving background" is "linked to scroll", the image displayed when the hero is at coordinate 0,0 should always be the same."**

Because of this rule, if the map size(in pixels) is not a size evenly divisible by the width and height of the background image, you will see the background jump into place when you loop around.

[12] My event will not start.

If the activation condition is **"Confirm Key," "Event Touch,"** or **"Player Touch,"** first check if **"Place Half-Step Up"** is turned on for the event. If it is on, try turning it off.
If the **"Default Character Move"** in **Basic Game Settings** is set to **1 Tile (1x1 Hitbox)**, you are unable to activate events a half-step above.

Otherwise, it is very likely some **activation condition** or **variable condition** is not met, or you have made a mistake in the ordering of the pages.

Higher number pages have priority for activation. If Page 1 and Page 3 both meet their conditions, Page 3 will be the one that executes.

[13] When using an encrypted folder with Set String "Get File List of Folder..", I get <<ERROR>>.

This is intentional (as you would be able to see all the contents of encrypted folders).

[14] When using Move Route, the first action keeps repeating / the first action executes 2 or more times.

When using Move Routes in auto-starting or parallel events, make sure you are making proper use of "Wait" and "Wait For Move Route."

If the same "Move Route" executes before it finishes the first time, it will try to start from the beginning again.

As such, in auto-starting or parallel events, if you are repeatedly calling a "Move Route," it may be stuck doing only the first action of it forever, depending on how it is constructed.

In general, you should be able to resolve this issue by putting a **"Wait For Move Route"** after the Move Route command.

[15] I want to display 2x-size pixels like with 320x240, but at higher resolutions.

Currently, there is no functionality to display sizes other than 320x240 at a zoomed size.

If you change the screen size of the Sample Game to something other than 320x240, you must take the following steps if you want it to still display with double-size pixels.

1. Change the Screen Dimensions in Basic Game Settings. For instance, change to 848x480.
2. Change the Tile Size in Basic Game Settings from 16x16 to 32x32.
3. Change the Picture Scale Method from "Smooth and Fuzzy" to "Distinct and Rough [Standard]."
4. Enlarge all images to double their original size, saving them with the same filenames.
 - Characters (in CharaChip folder)
 - Map chips (in MapChip folder)
 - All other images (in BasicData, BattleEffect, EnemyGraphic, Fog_BackGround, Picture, SystemFile folders)

After performing these steps, run the game, and it should display with double-size pixel graphics at sizes other than 320x240.

Just remember: **Higher resolution means bigger graphics.**

■ Opening

- WOLF RPG Editor is a game construction tool that can create complex role playing games.
 - The games you make with it can be freely distributed, submitted to contests, and sold.
 - It is completely free, so all its functions are available, including encryption functionality.
 - With the proper mastery, you can make any kind of game, not just RPGs.
-

■ Supported Environments

■ Supported OS: **Windows 2000, XP, Vista, Windows 7, 8, 8.1, 10~**

- * These are the supported OSES for both playing and developing games.
- * Running through virtual systems like Wine may not work as-is, but I am told deleting the included GuruguruSMF.dll may allow it to run without MIDI playback.

■ System Requirements

For Editor: 1 GHz CPU, 128 MB RAM

For Games: Varies greatly depending on game's processing load and resolution.

If you have a graphics board that can do 3D drawing, this will greatly speed up processing.

- * With just a **1 GHz CPU, 128 MB RAM**, and a laptop graphics chip, the Sample Game (320x240 resolution) runs at a maximum of about 70% speed.

■ Required Runtime: **DirectX 9.0, 11, or greater (Included as standard in Windows XP SP2 on)**

■ System-Dependent Issues in Gameplay

- ◆ During gameplay on Windows 2000, if "**User.dll**" is out of date, it may display "**GetRawInputDeviceList could not be found in USER32.dll**" and fail to start. If this happens, look for a user32.dll Version 5.1.26 or higher, and replace it in the Windows\System folder. However, be cautious of old user32.dlls circulating online. (Win98: 68KB - Ver. 4.10 / Win2000: 393KB - Ver. 5.0.02 / WinXP: 561KB - Ver. 5.1.26)
- ◆ As a result of security software such as anti-virus, some have reported an issue where **direction keys do not register** (but Confirm/Cancel/etc. work). If this happens, you should be able to play as normal if you add WOLF RPG Editor's Game.exe to the anti-virus's "exception" settings. (07/2009)

■ System-Dependent Issues in Editor

- ◆ **If you change the window size, the buttons at the bottom of the window may become hidden.** On Windows 7 (and Vista?), it appears if [Control Panel] -> [Make text or other items larger or smaller] is set to "**Medium: 125%**", then changing maximization or size will hide the bottom buttons. Setting the size to "**Smaller: 100%**" or "**Larger: 150%**" should avoid the issue.
 - ◆ **If the Editor's font seems to be too big for the layouts:** WOLF RPG Editor displays all text with MS UI Gothic, but if this font does not display properly, layouts may become broken, and some text may be hard to read. If this happens, run the following registry file, which will reset MS UI Gothic. -> [<Internet Connection Required> MS UI Gothic Reset Data \(0.2KB\)](#)
-

■ Usage Agreement

■ Usage of WOLF RPG Editor■

Games developed in WOLF RPG Editor can be distributed for free, distributed for money, sold, used commercially (at any scope), submitted to contests - in general, they can be freely used for any purpose. Also, no matter how it is used, reporting to the creators is unnecessary.

■How to Install■

Simply run the self-extracting package.

Start "Editor.exe" in the folder to begin editing, and start "Game.exe" to play the Sample Game.

It does not alter the registry in any way, so if you copy the whole WOLF RPG Editor folder to a USB drive, you can bring your development environment anywhere.

■How to Uninstall■

Deleting the WOLF RPG Editor folder will fully uninstall it.

The registry is not altered in any way.

■Support■

Bug reports are accepted in the "Bug Report Thread" on the WOLF RPG Editor official site forums (**Japanese**).

■Reviews of/Tutorials on WOLF RPG Editor■

I have received messages from excited people wanting to write reviews and explanations (that is, tutorials on usage) of WOLF RPG Editor, but there is no need to get permission before or after the fact.

Feel free to just write them as you please.

If you do, you can get a few more eyes on it by posting your website to somewhere like the links page on the WOLF RPG Editor wiki (**Japanese**), which others should appreciate.

For tutorials, screenshots of the Sample Game and its resources, and screenshots of the Editor, can be taken and used without permission.

■About Resources■

Please always follow the terms of use for any resources you use.

In particular, RPG Maker RTP resources, and resources modified from them, are only allowed to be used with RPG Maker.

* RTP (Runtime Package)

A basic collection of data required to play RPG Maker games.

According to its terms, usage of it outside of the RPG Maker series is forbidden.

■Replacing the Program Icon■

Using external tools to edit the program icon of WOLF RPG Editor's "Game.exe" and "Config.exe" is allowed.

However, I cannot respond to any issues that may arise, so you must take responsibility if you do this.

■Warranty and Disclaimer■

This software is not guaranteed to run in all system environments.

This site will take no responsibility for any trouble that results from the use of this software.

■Copyright Info for Libraries Used

- DX Library

DX Library Copyright (C) 2001-2005 Takumi Yamada.

- JPEG Loading Functionality

libjpeg Copyright (C) 1991-1998, Thomas G. Lane.

- PNG Loading Functionality

libpng Copyright (C) 1998-2004 Glenn Randers-Pehrson.

zlib Copyright (C) 1995-2004 Jean-loup Gailly and Mark Adler.

- OggVorbis Loading Functionality

ogg_static, vorbis_static, vorbisfile_static

Copyright (C) 1994-2002 XIPHOPHORUS Company.

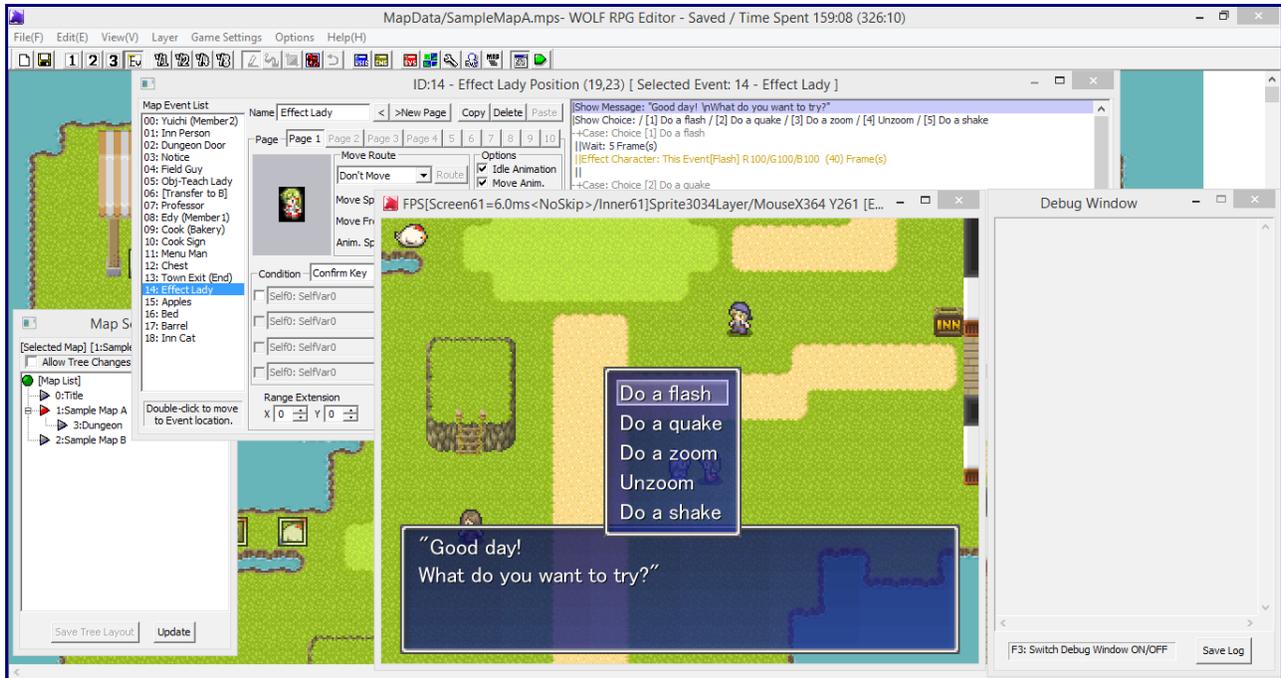
[Features and Functions of This Tool]

WOLF RPG Editor is a game construction tool that can create complex role playing games. It is often regarded as being aimed at advanced users, so keep this in mind if you intend to use it.

[Main Features and Functions]

■ Create An RPG Just by Assembling Event Commands!

Just by creating maps and putting together the provided event commands, you can make a 2D RPG.



A Practical Example of Event Commands

■ Configure Detailed Game Settings!

You can choose from a resolution of 320x240 or 640x480, and drawing at 30FPS (updating 30 times per second) or 60FPS! You can also change settings like whether to move by full or half-tiles, and the fonts used.

Basic Game Settings ✕

Game Name (If Main is changed, you cannot use old Save files.)
 Main Post

Tile Size: Screen Dimensions: Processing FPS: 60 FPS 30 FPS

MIDI Playback: Software (Recommended) Hardware

Character Sprite Direction Type: 4-Directional 8-Directional
 Character Animation Patterns: 3-Pattern 5-Pattern

Character Movement: 4 Directions 8 Directions
 Character Shadows: Use Do NOT Use

Default Character Move: 0.5 Tile 1 Tile (1x1 Hitbox)
 Default Hitbox: 1x0.5 Tile 1x1 Tile

Starting Hero Graphic:
*(Refers to the Sample Game Engine)
 *Usually does nothing when using Base System 2.
 Create parties with VariableDB Type 6.

Picture Scale Method: Distinct and Rough [Standard] Smooth and Fuzzy [3D Mode Only]

Inactive Window Behavior: Keep Running Pause Processing
*Pausing is recommended for editing while testing.

Move Speed Adjustment	
Hero/Allies	Event
Auto > <input type="text" value="1x"/>	<input type="text" value="1x"/>
Speed 0: <input type="text" value="1"/>	<input type="text" value="1"/>
Speed 1: <input type="text" value="2"/>	<input type="text" value="2"/>
Speed 2: <input type="text" value="4"/>	<input type="text" value="4"/>
Speed 3: <input type="text" value="8"/>	<input type="text" value="8"/>
Speed 4: <input type="text" value="16"/>	<input type="text" value="16"/>
Speed 5: <input type="text" value="32"/>	<input type="text" value="32"/>
Speed 6: <input type="text" value="64"/>	<input type="text" value="64"/>

Current speed limit is [32]
Units are [0.25 pixels / 1 frame]

Fonts Used (Blank = MS Gothic[Japanese])

Base:
 Sub 1:
 Sub 2:
 Sub 3:

Font Antialiasing: On Off Off + 2x

Adjust Message Display (Picture String)

Horizontal Spacing (Pixels):
 Line Spacing (Pixels):
*Horizontal 0, Line 6 (Recommended)
 Choice Separation (Pixels):

Adjust Game.exe Version:
Use only when you want Game.exe to behave like the old version.

2-Byte Character Language [Experimental]:
 System Language:

Basic Game Settings Screen

■ An All-Purpose Selection of Event Commands

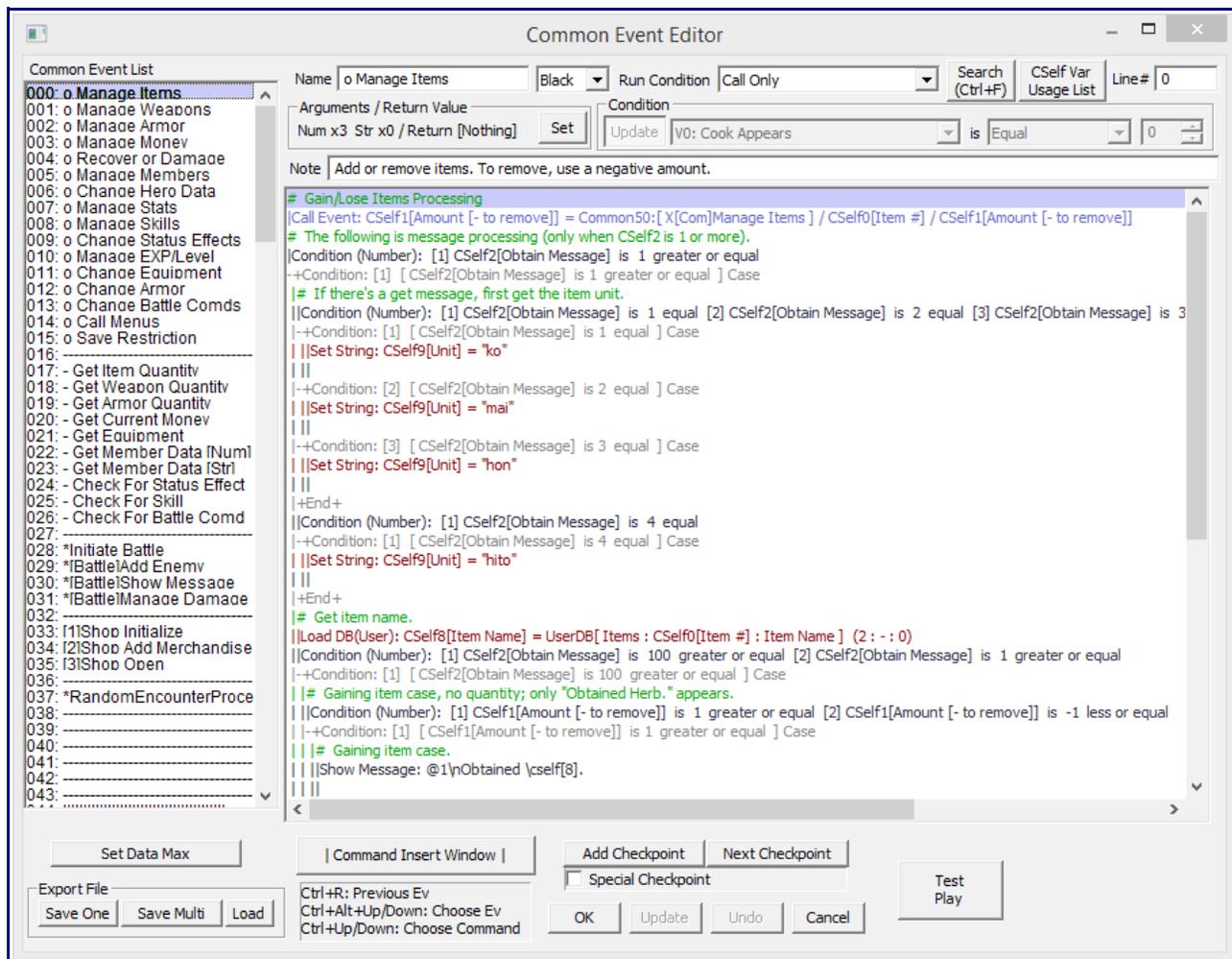
This tool features event commands that can be used for rather diverse purposes. In its barest state, it does not include a battle system or menus, yet many systems like these can be made using the event commands. Also, almost any numeric field can reference a variable, so you can build systems with minimal processing.

The screenshot displays the 'Event Creation Screen' for a game engine. The main window is titled 'ID:7 - Professor'. On the left, a 'Map Event List' shows various events, with '07: Professor' selected. The central area is divided into several sections: 'Name' (Professor), 'Move Route' (Don't Move, Route), 'Move Speed' (Standard), 'Move Freq.' (Medium), and 'Anim. Spd.' (Medium). Below these are 'Options' including 'Idle Animation', 'Move Anim.', 'Fixed Direction', 'Slip-Through', 'Above Hero', 'Square Hitbox', 'Place', and 'Half-Step Up'. There are also 'Condition' fields (Self0: SelfVar0 is Equal 0) and 'Range Extension' (X: 0, Y: 0). The right side shows a script editor with commands like 'Show Message', 'Wait', 'Choice', and 'Picture Show'. A 'Command Insert Window' is open, listing various commands like 'Show Message', 'Set Variable', 'Condition', etc. The background shows a 2D map with a building and a river.

Event Creation Screen

■ Exporting/Importing Common Event Files

You can export Common Events as files to post and exchange them online! With this function, you can easily put systems other people have made into your own game.



Common Event Creation Screen

* The lower-left buttons are used to export files.

You can also export multiple sequential Common Events in one file!
There is also an event command that allows "calling the Common Event 3 down from this one," so it is possible to distribute and import large-scale, Multi-Common-Event systems!

■ Databases, Key to Developing Your Own Systems

One unique feature of WOLF RPG Editor is the existence of databases. Databases can be set up however you want, and making use of them in events can make data management for your systems much easier.

The tool features three types of databases.

- **System Database** ... A database for setting up core elements of the game. Generally, you will not add your own fields to this database.

- **User Database** ... A database used to hold unchanging game information, such as weapon parameters or enemy battle stats.

- **Variable Database** ... A database of information that can change during the game, like the heroes' status and enemy/ally HP in battle. This is the only database whose contents can be rewritten by event commands.

The screenshot shows two overlapping windows from the WOLF RPG Editor. The 'Variable Database Editor' window is in the foreground, displaying a list of types on the left, a 'Data' list in the middle, and a detailed 'Type Definition' for 'Hero Status' on the right. The 'Type Definition' window is also visible, showing a grid of fields for defining character statistics.

Variable Database Editor - Type Definition for 'Hero Status':

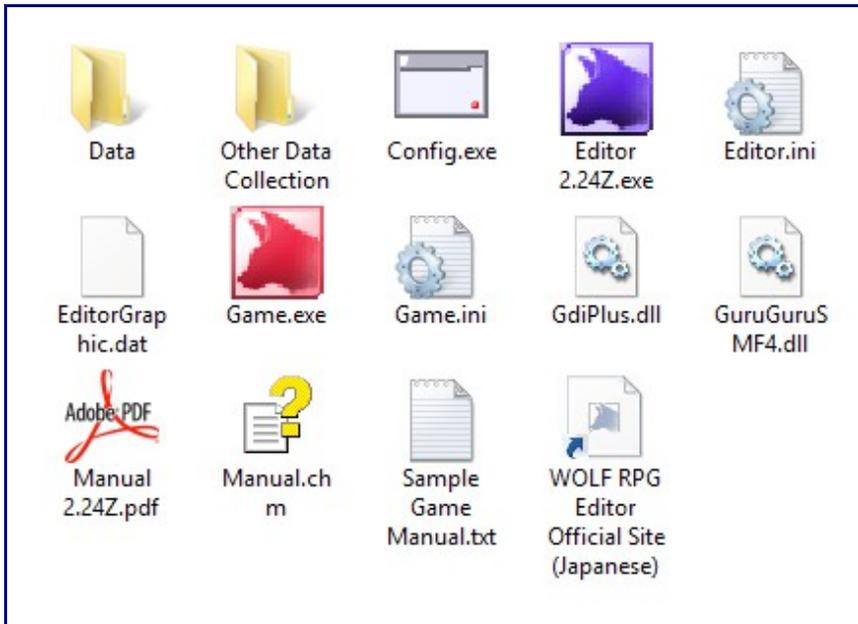
Field #	Field Name	Default
Field 0	Character Name	0
Field 1	Title	0
Field 2	Walk Graphics Image	0
Field 3	Face Image[Battle/Status]	0
Field 4	Level	1
Field 5	Max Level	99
Field 6	HP	50
Field 7	Max HP	50
Field 8	SP	20
Field 9	Max SP	20
Field 10	Base Attack	10
Field 11	Base Defense	10
Field 12	Base Special Attack	10
Field 13	Base Special Defense	10
Field 14	Base Agility	10
Field 15	Base Accuracy [%]	90
Field 16	Base Evasion [%]	0
Field 17	Critical Rate [%]	3
Field 18	[Growth]Max HP Add/1Lv	8
Field 19	[Growth]Max SP Add/1Lv	8

Variable Database Settings Screen

* You can define your own entries for HP, character names, etc. However, making your own entries has no inherent purpose unless you make events that refer to that data.

[File Explanations]

This explains the file contents of the "Full Package" version.



[Minimum Required Files to Run Games]

* Without the following data, a game cannot properly start.

- **Data Folder** ... Put all the files used by the game in this folder. Folders named anything other than "Data" will not function. If you copy in the Data folders from the "Other Data Collection" (explained further down), you can start developing from a blank slate. Also, important basic data like database info and Common Events is stored in the "BasicData" folder in Data.
- **Game.exe** ... The executable file. Run this to start the game.
- **Game.dat** ... You will find this file in the "BasicData" folder within the "Data" folder. It stores the information in "Basic Game Settings."
- **GuruGuruSMF4.dll** ... An important DLL for MIDI playback.

[Minimum Required Files to Run Editor]

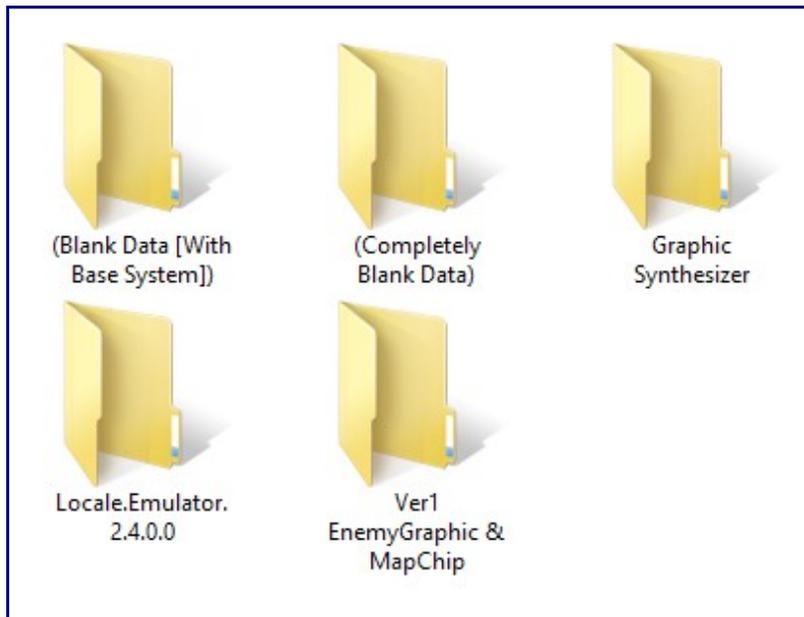
* In addition to the 4 files above, the below files are also required.

- **Editor.exe, Editor.ini** ... The Editor executable, and its settings file. The settings file stores things like what the last-loaded map file was.
- **EditorGraphic.dat** ... An image file of graphics used in the Editor. Actually a PNG file.
- **GdiPlus.dll** ... A DLL used to display images in the Editor.

[Other Files]

- **Config.exe** ... An executable that allows one to change the game's drawing mode, sound settings, etc. It is not required to distribute the game with it, but it is good to include.
- **Game.ini** ... A settings file that stores the drawing mode of the game executable and other things. Game.exe can still run without this file.
- **Manual 2.24Z.pdf** ... The English translation of Manual.chm
- **Manual.chm** ... The original **Japanese** manual file.
- **Shortcut to WOLF RPG Editor Official Site (Japanese)** ... A link to open the official **Japanese** website.
- **Sample Game Manual.txt** ... An instruction manual for the Sample Game. Refer to it when making your own game manual.

[Other Data Collection File Explanations]



(Completely Blank Data):

A folder with no Base System, only the minimal amount of initial data.

By copying the "Data" folder contained within, you can start from a complete blank slate.

(Blank Data [With Base System]):

A folder with the Base System (Sample Game Engine) and the minimal amount of initial data.

It does not contain any data from the Sample Game.

Graphics Synthesizer:

Software to create character walk graphics by combining parts.

Locale.Emulator.2.4.0.0:

You can use Locale Emulator to run WOLF RPG Editor as if it is on a **Japanese** locale computer.

You do not need to change your System Locale.

Since WOLF RPG Editor English 2.24Z is 99% translated (due to still possible remnants of **Japanese**), there should not be really any need to use Locale Emulator.

Ver1 EnemyGraphic & MapChip:

The files in this folder are from an earlier version of WOLF RPG Editor.

They are provided from the developer, but they are not necessary for game creation.

Unless you plan to use or learn from these files, we highly recommend creating and using your own images.

We hope you learn from what is provided in the "Full Package" and "Start Package".

Please give credit to the artists should you use their images in their original form.

[Monster Graphics]

Isooki (<http://www.geocities.jp/iso015/>)[**Japanese** but the website is nonexistent]

Suu (<http://suulabo.dw.land.to/>)[**Japanese** but the website is nonexistent]

[Map Graphics]

Nori (<http://nega.client.jp/>)[**Japanese**]

* Parts by the following used from the WOLF RPG Editor Wiki and Resource Request Thread (**Japanese**).

Yuuto, RETAW, Suu, Karekusa, Sumika, Urocchi, Rikugame

[Basic Controls]

■Starting and Closing Editor

- To start the Editor, run Editor.exe inside the extracted WOLF RPG Editor folder.
- To close the Editor, you can click the [X] button in the top-right of the window. You can also close the Editor by clicking on "File" and then clicking "Exit Editor".

■Starting and Closing Game

- To start the game, run Game.exe inside the extracted WOLF RPG Editor folder.
- To close the game, at any time, you can hold Alt on the keyboard and press F4. In windowed mode, you can also click the [X] button in the window's title bar.

■Game Controls

- There are four basic controls for games: direction buttons, the Confirm button, the Cancel button, and the Sub-Button. These inputs are done by pressing the following keys.

Button	Corresponding Keys
Direction Up:	[↑] [Numpad 8]
Direction Down:	[↓] [Numpad 2]
Direction Left:	[←] [Numpad 4]
Direction Right:	[→] [Numpad 6]
Confirm Button:	[Space] [Z] [Enter]
Cancel Button:	[Esc] [X] [Numpad 0] [Backspace] [Insert]
Sub-Button:	[Shift] [C]

The main function of each button is as follows, but these may not always fit depending on the systems implemented in each game.

Direction Buttons: Move the character.

Confirm Button: Confirms choices. Checks (talks to/inspects) events underneath or in front of the hero.

Cancel Button: Cancels choices. In the Base System (Sample Game Engine), it opens the menu.

Sub-Button: Used for functions other than the above (like dashing). In the Base System (Sample Game Engine), you can hold it in battle to speed the battle up.

- Some function keys have special functions.

F4: Changes window size (windowed mode only). You can switch between the following three sizes in order.

- 320x240 (400x300) (Only if graphics mode is "3D Mode")

- 640x480 (800x600) (Default size)

- Fake Full Screen (Maximum size that fits display; increases processing load a fair bit)

F5: Changes the frame-skip level. If you use a slow draw rate, processing will speed up (but movement will not be smooth).

F11: Maximizes the game window. Press again to revert to normal.

F12: Returns to the title screen.

There are also additional functions during Test Play. For details, refer to "[Debug Command List](#)."

- Some other keys have special functions.

PrintScreen, ScrollLock, Pause: Takes a screenshot. PrintScreen may not function in some environments.

◆Direction Button Behavior

When directions 180 degrees opposite to each other are pressed, the more recently-pressed key is prioritized.

With 4-directional movement, if up/down and left/right are combined, the up/down direction is prioritized.

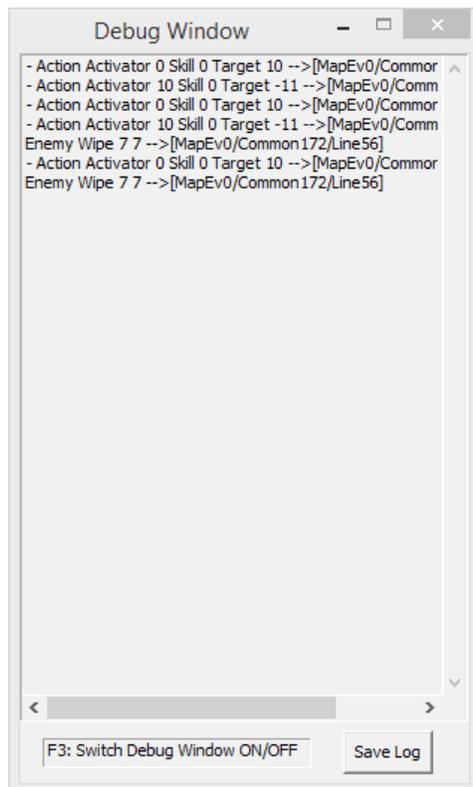
[Debug Command List]

During a Test Play, you can use the following commands with Game.exe.
Note that you cannot use any of these when starting up Game.exe directly.

- [Debug Window \(F3\)](#)
 - ["Save Log" Button](#)
 - [Detailed Picture List \(F7\)](#)
 - [Debug Data \(F8\)](#)
 - [Variable List \(F9\)](#)
 - [Load Data \(F11\)](#)
-

[Debug Window (F3)]

If the Editor has "Game Settings" -> "Use Debug Window" checked, a debug window like this will be shown.
Pressing F3 during Test Play will also show the window.



Even if this option is checked, the debug window will not appear when Game.exe is started directly.
The debug window only functions during Test Play.

The debug window will display text written to it as debug text using "[Show Message](#)," and error reports.
To aid in debug tracing, "MapEv#/CommonEv#/Line#" is automatically added at the end of each message.

["Save Log" Button]

Click the "Save Log" button on the Debug Window to save the text in the window to a text file. The file is named DebugLog.txt, in the same folder as Game.exe.

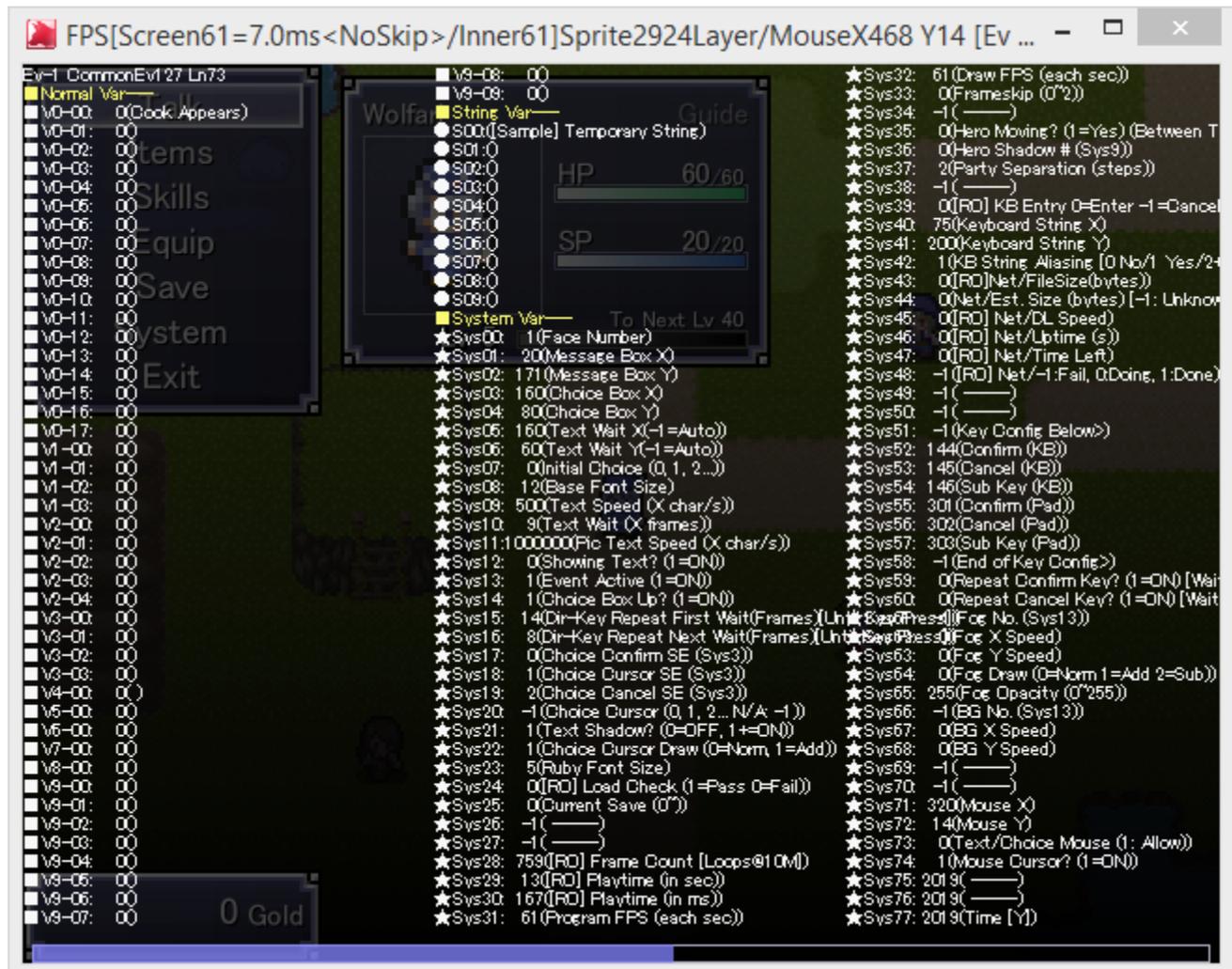
[Detailed Picture List (F7)]

Pressing F7 during Test Play shows a list of all pictures currently being displayed. It is useful for seeing picture positions, display states, and the events that display them.



[Variable List (F9)]

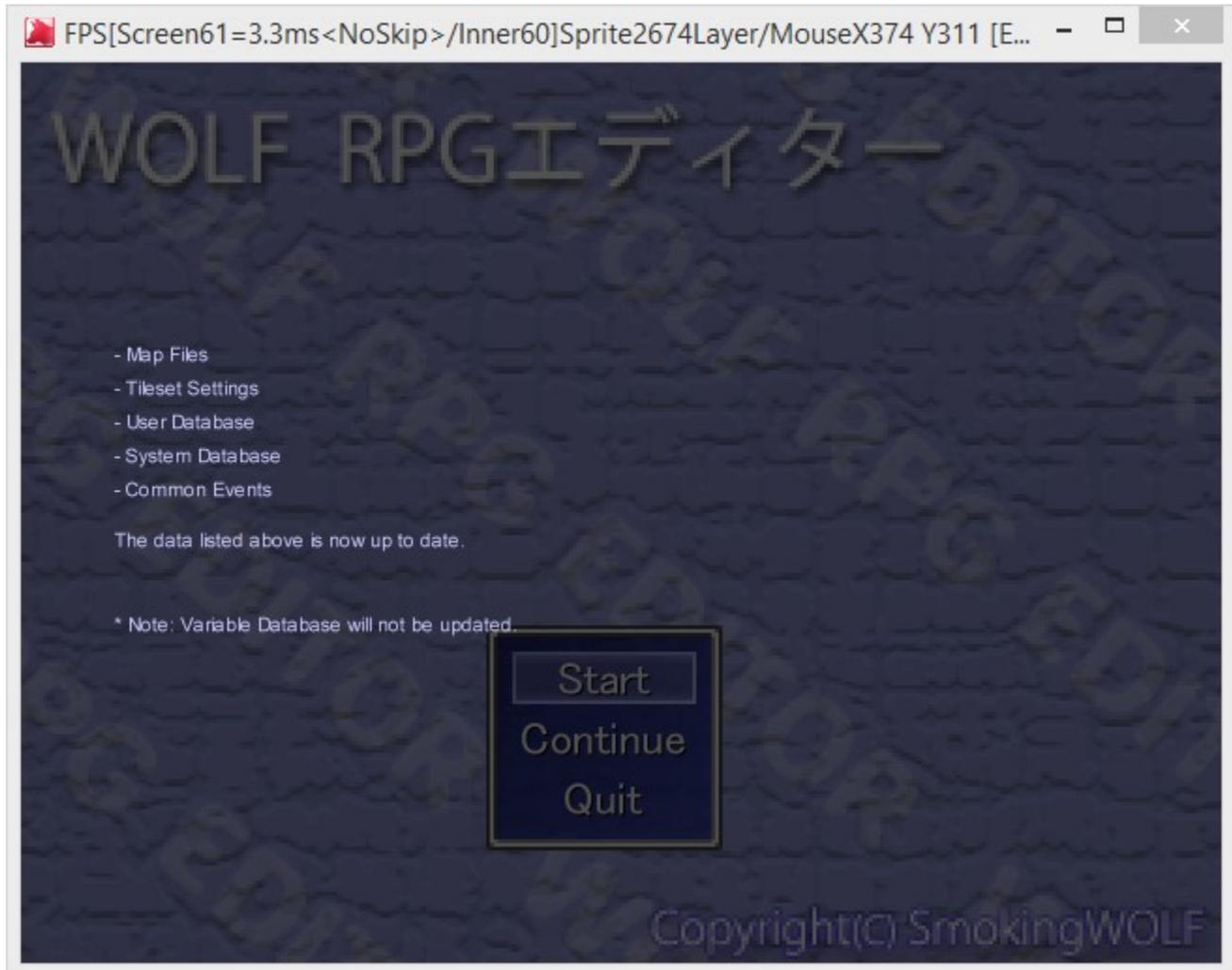
Pressing F9 during Test Play lets you see all Normal Variables, Spare Variables, String Variables, System Variables, and System Strings.



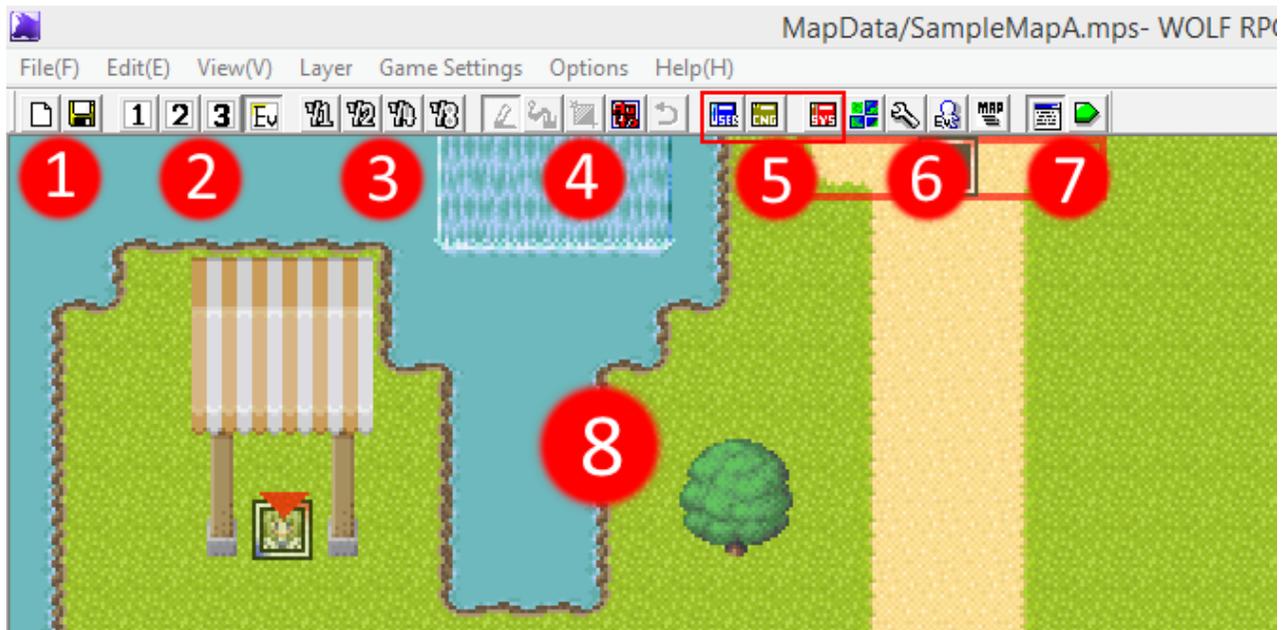
[Load Data (F11)]

Pressing F11 during Test Play will update Map files, Tileset Settings, the User Database, the System Database, and Common Events to their latest versions.

The Variable Database will not be updated. Also, if pressed during event execution, things will be reset.



[Main Window]



[Operation]

- With "Layer 1~3" selected from (2):

Left-Click ... Paint current chip onto a tile.

Right-Click ... Eyedropper (copy) function. By dragging, you can copy chips in an area.

Ctrl + Right-Click ... Copy-all function. By dragging, you can copy all chips in an area on Layers 1~3.

Ctrl + Left-Click ... Paste-all function. Pastes the Layer 1~3 chips gotten via Ctrl + Right-Click.

Shift + Left-Click ... Paint chips ignoring the auto-tile function (i.e. the process that auto-connects water tiles).

- With "Event Layer (Ev)" selected from (2):

Left-Click ... Select Event

Double Left-Click ... Create/Edit Event

Right-Click ... Display Sub-Menu

- Common Between "Layer 1~3" and "Event Layer (Ev)":

Drag With Middle Button, Space + Move Mouse ... Scroll Map

Ctrl + Shift + Arrow Keys (Up/Down/Left/Right) ... Shift Events/Chips on All Layers

- Shortcut Keys

1~4: Change Layer (1~3 = Layers 1~3, 4 = Event Layer)

Ctrl+S: Save Map

X: Cut Event

C: Copy Event

V: Paste Event

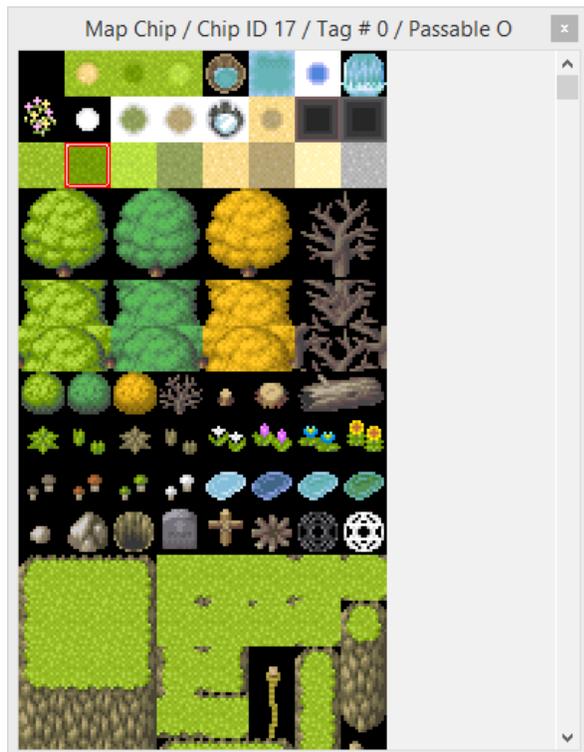
Delete/Backspace: Remove

[Individual Explanations]

- 1: Buttons to Create New Map and Save the current map.
- 2: Buttons to select and edit Map Layer (1~3) and Event Layer (Ev).
Use these to switch between what you want to edit.
Clicking a Map Layer button will show the map chip window.
- 3: Change the map scale, from 1/1~1/8. Useful to get a view of the whole map.
- 4: Drawing tools used for map creation. 
From left to right: Pencil Tool, Fill Tool, Square Fill Tool, Auto-Crete Map Tool, Undo (*Map Chips Only).
- 5: Open each database.
From left to right: Blue = User Database, Yellow = Variable Database, Red = System Database.
- 6: Special tools related to maps.
From left to right: Tileset Settings, Map Settings, Common Event Settings, Map Select.
- 7: Starts a Test Play.
During Test Play, you can press F8 to check loaded files and pictures in use, and F9 to view all variables.
- 8: The map editing area. Create your maps by placing chips here as if you are drawing a picture.

[Map Chip Window]

- Automatically appears when you click a Map Layer button (1~3).
Left-click to select a chip to use, then place it on the map in the Main Window. You can drag to select multiple tiles at once. When selecting single chips, the chip number, tag, and passable settings will be displayed.

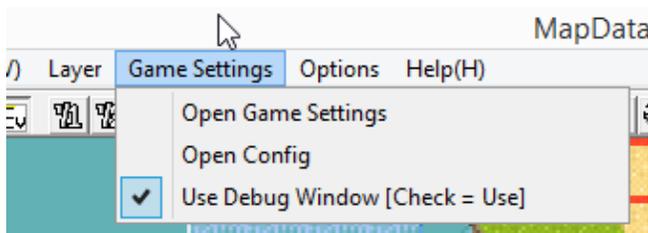


[Game Settings]

There are three kinds of game settings, plus a set of options:

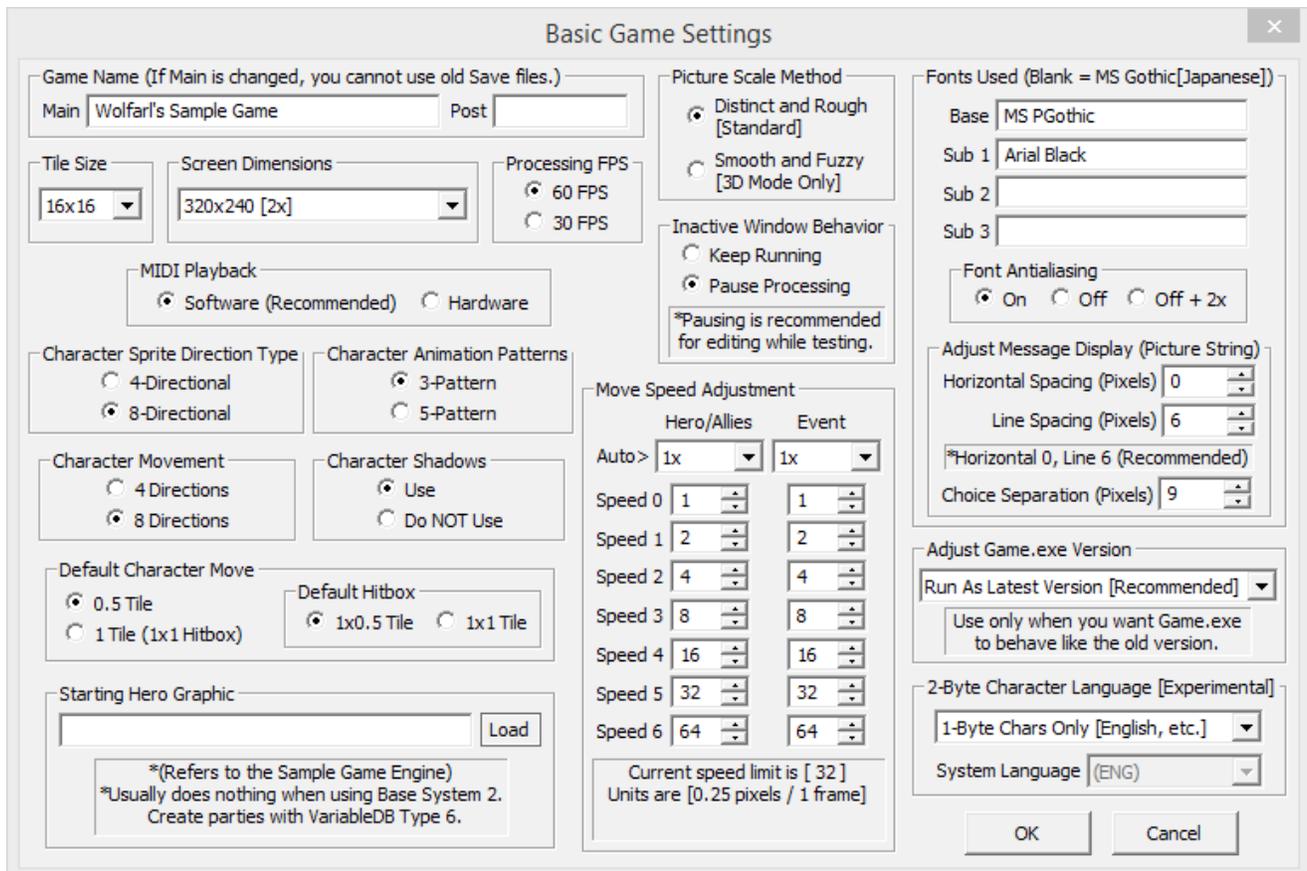
- [Basic Game Settings](#)
- [Config](#)
- [Use Debug Window](#)
- [Editor Options](#)

[Basic Game Settings]



You can edit Basic Game Settings from "Game Settings" -> "Open Game Settings."

The following window will be displayed.



You can configure all of the following settings.

Game Name (Main):

The title displayed in the window title bar when playing in windowed mode. It is also used as the default folder name when creating game data.

If this main name is changed, old save data will become unusable.

Game Name (Post):

This subtitle is appended to the main title and can be used for version numbers or the like.

Even if this postscript part is changed, past save data will remain usable.

Tile Size:

You can select a tile size of 16x16, 32x32, or 40x40.

Screen Dimensions:

You can choose from the following game screen sizes.

320x240 [2x], 640x480, 800x600, 848x480 (16:9), 854x480 (16:9), 960x720, 960x540 (16:9),
1024x768, 1024x576 (16:9), 1280x960, 1280x720 (16:9)

In the case of 320x240 only, the window is displayed at 2x size.

Screen size greatly impacts drawing load, so please be cautious.

Processing FPS:

You can choose the game's process FPS (how many times to draw in one second), between 30 FPS or 60 FPS. 60 FPS is smoother, but requires more processing. At 60 FPS, slowdown occurs if event execution time exceeds 16 milliseconds, whereas at 30 FPS, 33 milliseconds is the effective limit.

MIDI Playback: Choose whether to play MIDI with hardware or software.

- Software plays MIDIs using DirectShow, giving everyone the same sound. This is strongly recommended.
- Hardware has a pause on changing, but since it uses the internal sound player, it slightly reduces processing load. However, due to issues such as Windows Vista on not playing some audio tracks, this option is not recommended.

Character Sprite Direction Type:

Choose whether character graphics used in the game have 4 directions or 8 directions. This affects the format of the character graphics you must use.

Character Animation Patterns:

Choose whether character animations have 3 or 5 frames. This affects the format of the character graphics you must use.

Character Movement:

Whether characters can move in 4 directions or 8 directions. This does not affect the character graphics format; you can have 8-directional movement with only 4-directional character graphics.

Character Shadows:

Turn the function that puts shadows on characters on or off. One advantage of shadows is making characters jump and having their shadow stay on the ground while they float - it helps make it look more like the character is in the air.

Default Character Move:

Set the standard step amount for characters. With 0.5 Tile, you can get more fine movement.

Default Hitbox:

Choose whether characters have a hitbox of 1 tile wide and 0.5 tile high, or 1 tile wide and 1 tile high.

The hero's hitbox is locked into being what it is set to here. Event characters also use this setting by default, but it can be changed with the "Square Hitbox" option in the Event Editor.

Starting Hero Graphic:

Set the hero's character graphic at game start. Click the Load button to select an image file.

*** If using the Base System (Sample Game Engine), this will not function; instead, you need to configure the image in "Hero Status" and setup your character in "Party Info" of the Variable Database.**

Move Speed Adjustment:

Prior to Ver. 1.05, hero and character move speed differed, so this was added so you could replicate that. Setting the player to 1x and events to 2x will give the same behavior as old versions. If you are making a game for the first time, leaving both the player and events at 1x is recommended.

Also, you can finely configure the 7 stages of move speed for the heroes and events.

*** In some cases, the speed will appear to be the same even with different values.** (Added 12/30/2016)

Since characters always briefly stop in line with the tile grid, there are cases where the speed will appear to be no different. For instance, in "half-tile movement," with a 16-pixel tile size, any speed value from 16~31 will appear the same speed if you move continuously. (In the case of 16-pixel tiles: speeds 11~15 take 3 frames to move 1 tile, 16~31 takes 2 frames, 32 takes 1 frame.)

Picture Scale Method:

Choose the method with which to resize pictures: "**Distinct and Rough**" or "**Smooth and Fuzzy**."
"Smooth and Fuzzy" will only work in 3D Mode.

* As of Ver. 2.22, drawing differs between full screen and windowed mode.

With "Distinct and Rough" on, full-screening and scaling will both enlarge "roughly";
with "Smooth and Fuzzy" on, full-screening and scaling will both enlarge "smoothly."

→ Essentially, **pixel-art-focused games will look best with "Distinct and Rough,"**
but otherwise, it often looks better to switch to "**Smooth and Fuzzy.**"

Inactive Window Behavior:

Set the behavior when the window is inactive (when you click on a window other than the game).

You can have it "**Keep Running**" or "**Pause Processing**."

Fonts Used:

You can define a main font and three sub fonts; you do not have to specify anything. However, make sure you input the font names exactly. The sub-fonts defined here can be temporarily switched to in messages using the special code \font[1] (change to sub-font 1).

Also, you cannot use fonts with a .fon extension (Adobe Type fonts like "WST_Engl" or "Small Fonts").
(The moment you try to display text with these fonts, the game apparently force-closes.)

* About Loading External Font Files

If a font is not installed on the PC being played on, "MS Gothic"(Japanese) is used by default, **but you can include the font file and it will be automatically loaded in.**

To do this, include the font file (**ttf and ttc formats only**) in the **Data folder (* only if not encrypting the entire Data folder)**, or in the **same place as Game.exe**.

* Please enter the font name exactly. On Windows, you can find font names by right-clicking a TTF file, clicking Properties, and in the Details tab, it is listed as "Title." (It is possible more exactness is required as of Ver. 2.20.) Also, if you are releasing to English audiences, you will need to input the English font name.

Font Antialiasing:

Settings for antialiasing on fonts (smooths the corners of text). For games that display a lot of small text, "Off" is recommended.

- **On** ... Makes text have smooth edges, generally making it easier to read, but small text becomes less distinct.
- **Off** ... Makes text have sharp edges. Even at small font sizes, it should be relatively readable.
- **Off + 2x** ... Displays font with sharp edges and at 2x normal size. Fitting for retro-style games.

Adjust Message Display (Picture String):

Make minor adjustments to the display of messages and picture strings. **Horizontal Spacing** is the empty space between each character, and **Line Spacing** is the vertical gap between lines when displaying messages. **Choice Separation** is the vertical gap between lines when choices are displayed.

Adjust Game.exe Version:

If necessary, you can make Game.exe function like an older version. Use this if things display all wrong after updating to a new version.

"Run As Latest Version" ... Runs as the latest.

"Run As Latest Version 2.20" ... Use if you want it to always function as the current version 2.20 into the future.

* Characteristics of version 2.20 will be listed here when a version with newer behavior is released.

"Run As Old Version 2.10" ... Runs using the following characteristics.

- [Ver. 2.10 Text Display] When using icons larger than the font size, even inputting "\i[??]\ay[0]~~" would not set the "~~" text's Y position to 0. In latest version, the Y position of the "~~" part will become 0.
- [Ver. 2.10 Text Display] Max font size is around 100 pixels. In latest version, you can go up to 400 pixels.
- [Ver. 2.10 Volume] Lowering Background Music (BGM) Volume Adjustment does not do much to quiet volume; even near 0%, it is rather loud. In the latest version, the volume is lowered in a way that fits the values.
- [Ver. 2.10 Move Event] "Move Away From Hero" only moves in 4 directions.
In the latest version, it can move away from hero in 8 directions.

"Run As Old Version 2.02" ... Run using the following characteristics.

[Ver. 2.02 Special Characters] Special code \m[X] causes vertical centering in 320x240 only (in latest, it is bottom-aligned).

[Ver. 2.02 Set Variable+] When getting tile tags, internal Y coordinates are 1 tile up from what they are in the latest version.

[Ver. 2.02 Messages/String Pictures] Newlines at the start of text without a character have no effect (in latest version, such newlines are reflected).

2-Byte Character Language:

Use when making games in languages other than Japanese. You can choose "**Japanese**," "**Hangul**," "**Chinese Traditional/Simplified**," or "**1-Byte Chars Only [English, etc.]**."

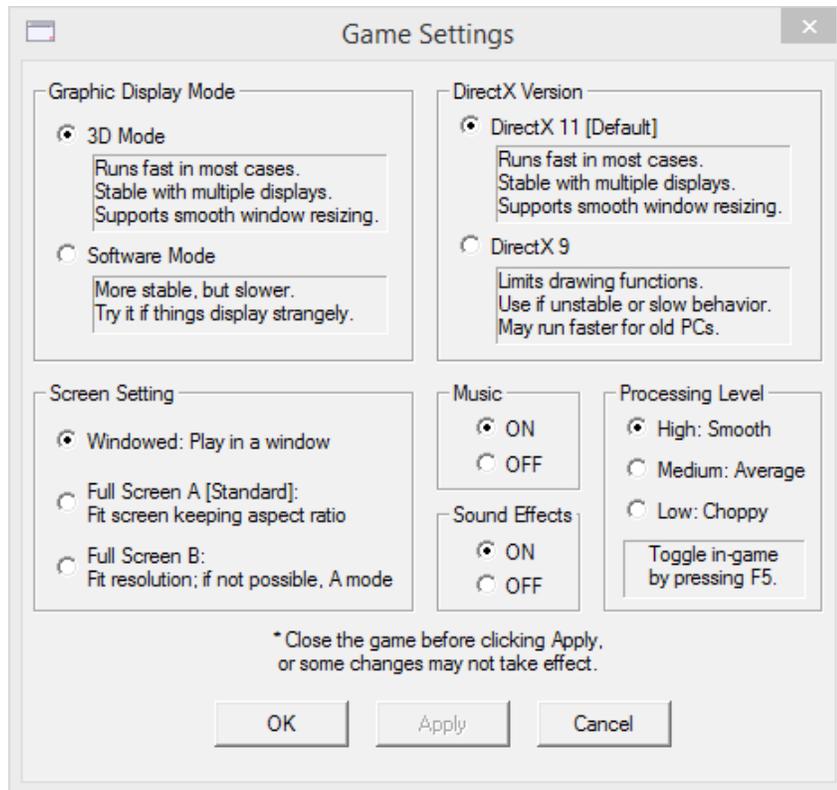
Please choose the proper language. For instance, if you set it to "**Chinese Traditional/Simplified**" and there is Japanese in the game, it will become gibberish.

- **System Language** ... The language just for the title bar and error (debug) messages.

You can choose "**Japanese (JPN)**" or "**English (ENG)**." Also, if the player's system locale (Windows language setting) is anything but Japanese, the game will automatically switch the system language to English.

[Config]

Selecting "Game Settings" -> "Open Config" runs Config.exe, which displays the following window.



You can configure settings for playing games.

Graphic Display Mode:

Choose whether to draw graphics with 3D Mode or Software Mode.

The default is 3D Mode, but if things behave oddly, changing to Software Mode may stabilize it.

Screen Setting:

Choose whether to start the game in "Windowed," "Full Screen A," or "Full Screen B" mode.

The default is Windowed.

DirectX Version:

Choose the DirectX version used for drawing in-game, 11 or 9.

The default is DirectX 11, which supports smooth drawing for enlarged windows.

(Only in 3D Mode, and if Basic Game Settings has Picture Scale Method set to Smooth and Fuzzy.)

Processing Level:

Adjust the balance of screen drawing and processing.

Music:

Turn playback of BGM and BGS on or off.

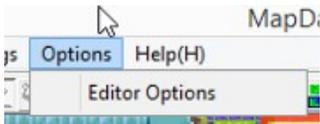
Sound Effects:

Turn playback of Sound Effects on or off.

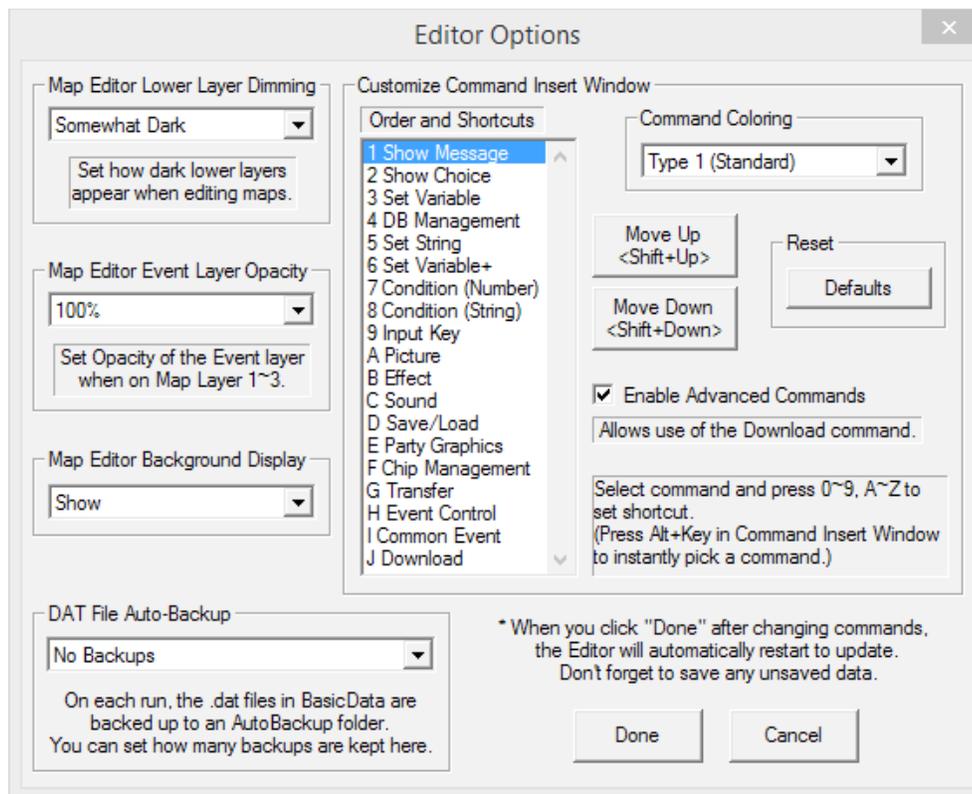
[Use Debug Window]

Set whether to display the debug window during Test Play.
For details, consult "[Debug Command List](#)."

[Editor Options]



You can change Editor Options from "Options" -> "Editor Options." It will display the following window.



You can change the following options.

- Map Editor Lower Layer Dimming

Set the visibility of the lower layer when editing maps.
You can choose from 5 stages of darkness.

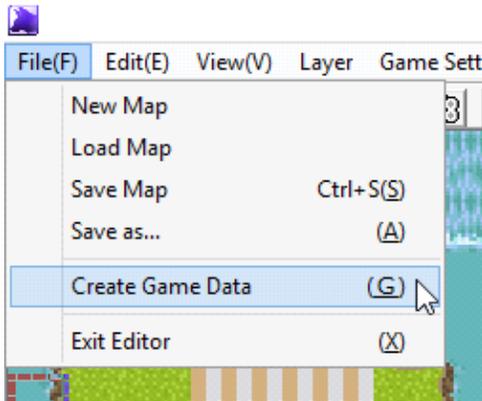
- Map Editor Background Display

Set whether to display backgrounds when editing maps.

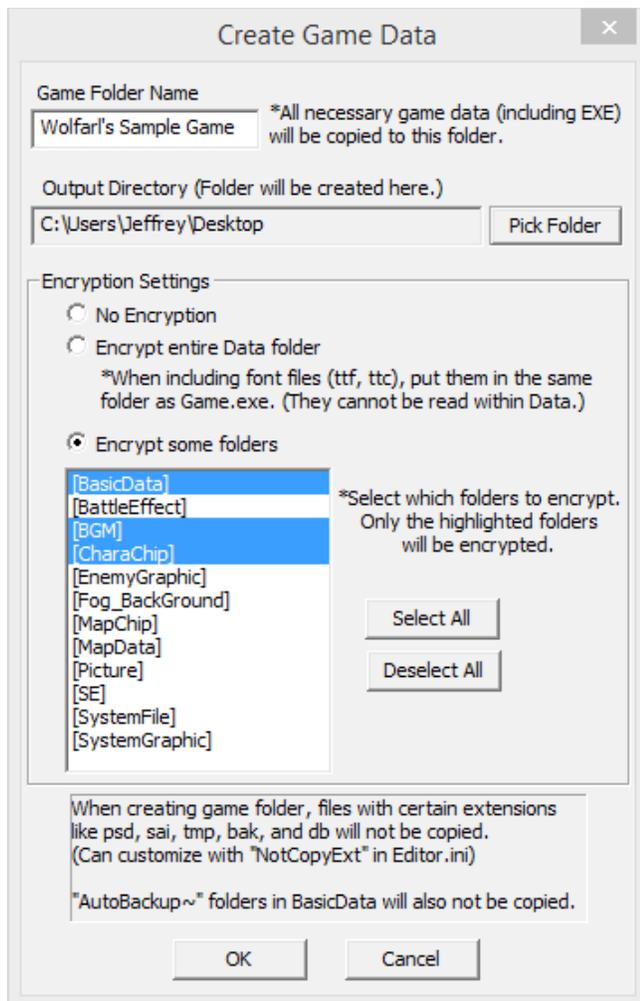
- Customize Command Insert Window

Set the order of the command list and shortcut keys.
Choose a command and press 0~9 or A~Z, and you will be able to use Alt plus that key as a shortcut for that event command.

[Create Game Data]



Accessed from "File" -> "Create Game Data." It will display the following window.



"Game Folder Name" and "Output Directory" are self-explanatory.

Here are some details about the encryption settings.

[Encryption Settings]

- No Encryption

Using no encryption, just copies all important files for the game to run (Game.exe, Game.dat, contents of the Data folder).

- Encrypt entire Data folder

Encrypts all the data in the Data folder into one file, copying it and other necessary files.

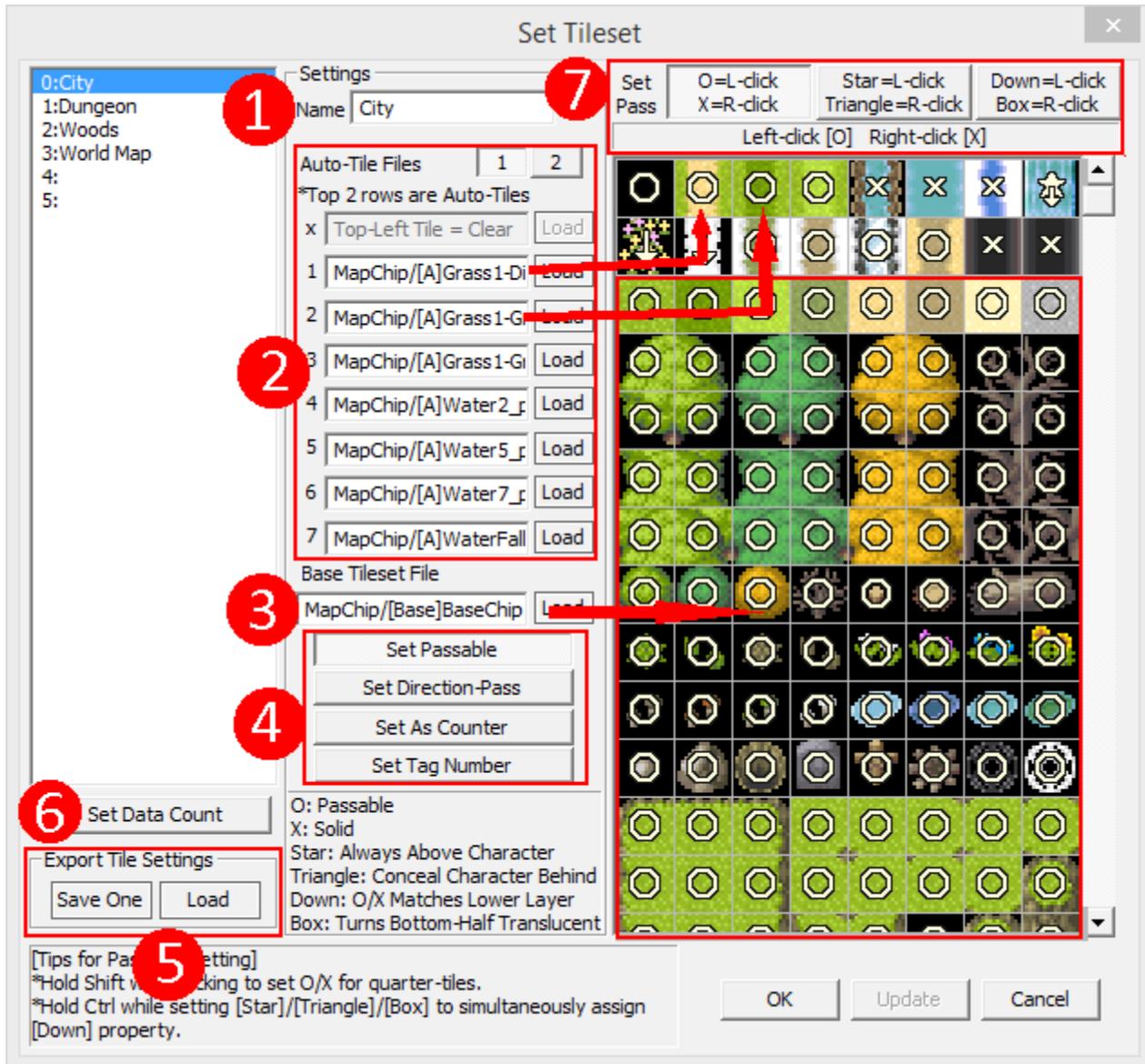
Also, if you are including font files, please move them to the same folder as Game.exe first.

- Encrypt some folders

Encrypts and copies the folders within the Data folder selected below. This can be rather convenient if there is a chance for future updates.

[Tileset Settings]

Tileset Settings can be opened by clicking this icon  in the Main Window. It is in the top-right of the window, divided up by a white grid.



1: Set the name of the tileset.

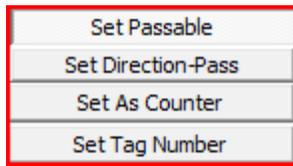
2: **Auto-Tile Files:** Specify filenames for each individual auto-tile, placed in the first and second rows of the grid on the right. **Auto-tiles are tiles that automatically create their boundaries.** The 1/2 buttons at the top switch between the first and second rows of auto-tile files.

3: **Base Tileset File:** Specify an image file for the third row onward in the grid on the right. You can click the Load button to view and select graphics.

4: Buttons to switch between four ways of editing chip settings.

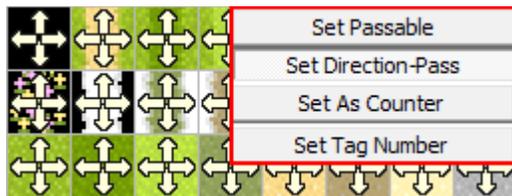
- Set Passable

Set whether chips can be passed through. For details, consult "[7: Passable Settings.](#)"



- Set Direction-Pass

Use when you want to set passability of chips from only certain directions.

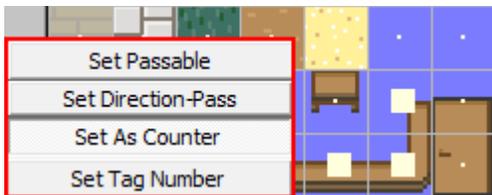


Symbol meanings:

- Can pass through in that direction
- Cannot pass through in that direction

- Set As Counter

Set whether chips have the "counter" property. If there is a counter between the hero and an event, you will be able to trigger the event across the counter.



Symbol meanings:

- Has the counter property
- Does not have the counter property

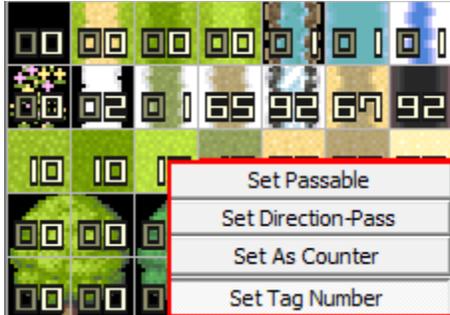


Example of starting an event across a chip with the counter property.

- Set Tag Number

Set tag numbers for chips. The tag numbers set here can be used via the event command "Set Variable+", which can get the tag number of a chip at a specific coordinate.

It is useful for assigning various properties to chips - for instance, changing the background depending on whether you encounter the enemy in a forest or in the desert plains.



Left-Click to add 1 to the number, Right-Click to subtract 1. Tag numbers can range from 0 to 99.

Also, tag 0 is used as a "transparent" tag.

When getting "**Uppermost Tile Tag**" in "[Set Variable+](#)", chips with tag 0 are ignored.

5: Export Tile Settings

You can export the selected tileset settings as a file.

6: Set Data Count

Change the total count of tilesets.

7: Set Pass

Set whether each chip can be passed through. There are six possible states, and you can change a tile's state to one of those currently selected with Left and Right-Click. (*) The symbols have the following meanings:

○ = Passable

× = Solid

▲ = Passable, and hides character if you go behind it (characters with "Slip-Through" ON will always be hidden by this chip). **To be more precise, "characters with the same Y position or a smaller Y position as a ▲ chip are hidden behind it."** (This behavior is not affected by the "height" set by Move Route.)

★ = Passable, and always draws above characters.

↓ = Passability is based on lower layer. If there is no lower layer, it is passable.

◻ (Half-Filled) = Characters on this chip will have their lower half (20/16/8 pixels) turn translucent. **However, if "height" (set by Move Route) is 1 or more, they will not turn translucent.**

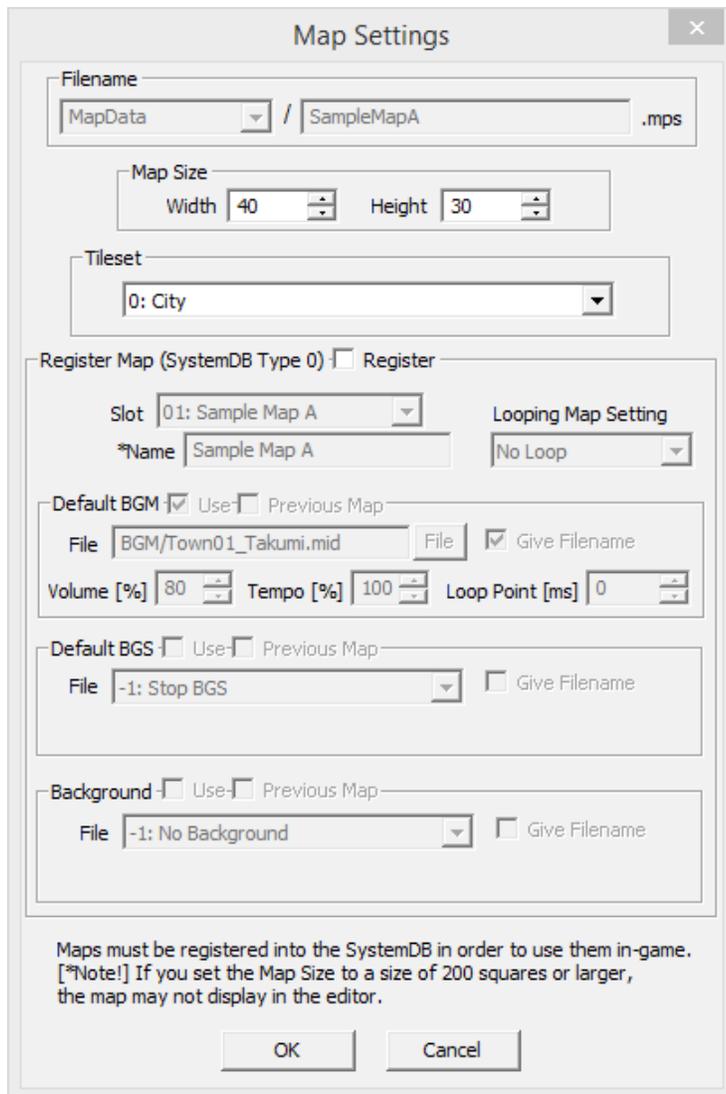
* Clicking a chip while holding Shift lets you set ○ and × for each quarter-tile.

Setting ○, ×, ★, ▲, ◻ while holding Ctrl lets you also assign the ↓ property.

* Chips placed on a higher layer than ▲ or ★ chips are automatically assigned the ▲ or ★ property (★ has priority over ▲). This is because of the fundamental nature of drawing, that you can only draw on top of something else.

[Map Settings]

Map Settings can be opened by clicking this icon  in the Main Window.



Map Settings

Filename
MapData / SampleMapA .mps

Map Size
Width 40 Height 30

Tileset
0: City

Register Map (SystemDB Type 0) Register

Slot 01: Sample Map A Looping Map Setting
*Name Sample Map A No Loop

Default BGM Use Previous Map
File BGM/Town01_Takumi.mid File Give Filename
Volume [%] 80 Tempo [%] 100 Loop Point [ms] 0

Default BGS Use Previous Map
File -1: Stop BGS Give Filename

Background Use Previous Map
File -1: No Background Give Filename

Maps must be registered into the SystemDB in order to use them in-game.
[*Note!] If you set the Map Size to a size of 200 squares or larger,
the map may not display in the editor.

OK Cancel

[Filename]

The filename of this map.
It is just displayed here for reference.

[Map Size]

Set the size of the map.
The units are number of chips.

[Tileset]

Select the tileset to use to draw the map.

[Register Map]

Map files (.mps extension) normally cannot be used in-game unless their filenames are put into System Database Type 0.

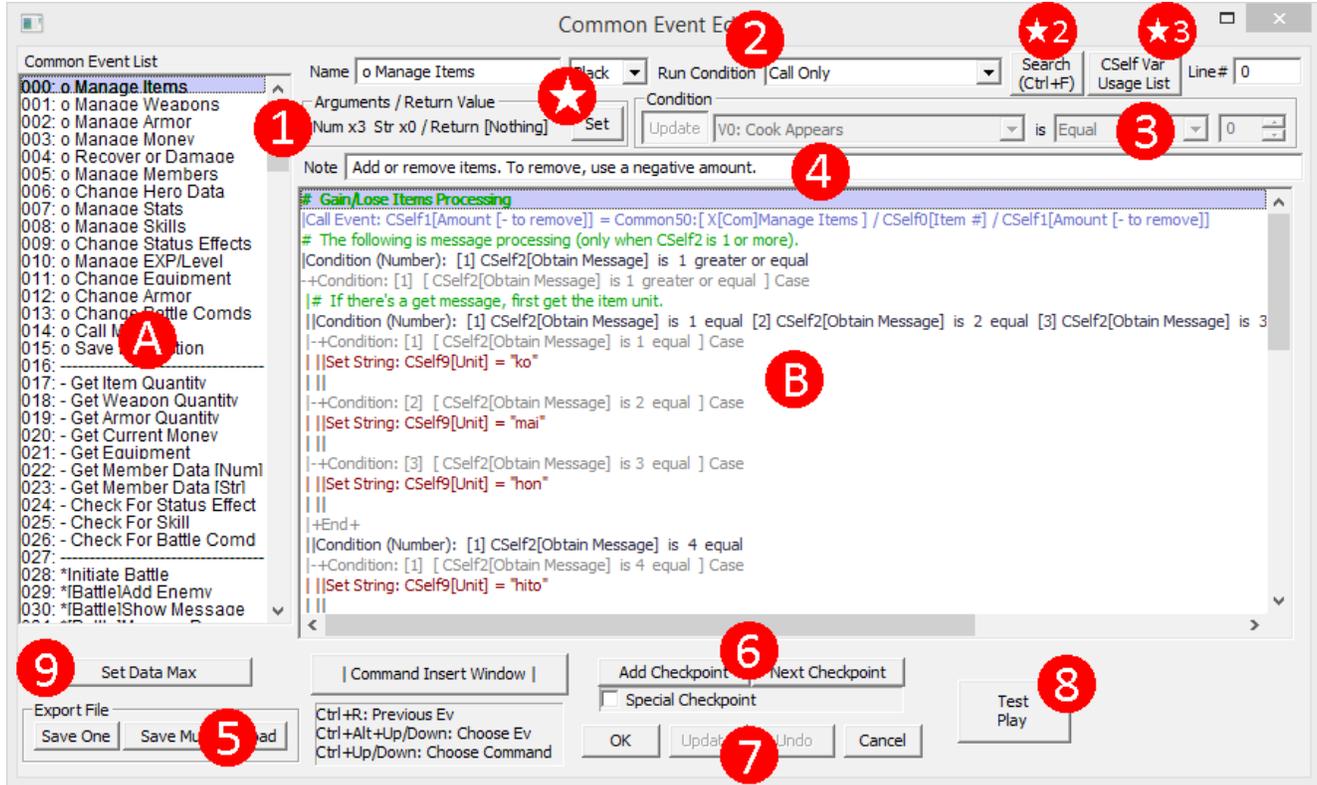
This section streamlines the process of adding maps to SystemDB Type 0.

To use it, check "Register," then specify the Map name, BGM, BGS, Loop settings, and Background settings.

[Common Event Settings]

* Common Events: Events that can be called and used from anywhere in the game.

You can open the Common Event Editor by clicking this icon  in the Main Window.



The general usage is identical to the Event Editor, so refer there.
Here, we will explain the features unique to the Common Event Editor.

A. Common Event List

Generally the same as the Map Event Window in (1), but with Common Events, you can do multi-copy.

B. Event Command Section

The same as (11) in the Map Event Window.
Also, dragging and dropping a .common file here has the same effect as loading the file.

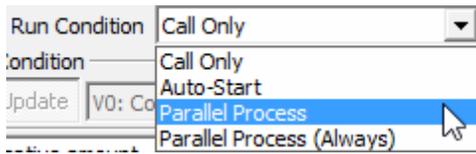
1. Arguments / Return Value

"Call Only" Common Events can define numbers or strings to take as "input." When called from another event, they will accept the specified number of "inputs." The number values provided for Arguments 1~4 are stored in Common Self0~3, and the strings provided are stored in Common Self5~8, allowing the Common Event to use these values via the CSelfs. In programming terms, these "inputs" are better known as "arguments."

The number of arguments and settings for return result can be set from the ★-marked "Set" button. The details are explained further down.

2. Run Condition

Set the condition which triggers the Common Event to run. There are four condition types.



Call Only ... Only runs when called by another event.

Auto-Start ... Runs when the Condition (see below) is met. Other events will not start when it is running.

Parallel Process ... Runs when the Condition (see below) is met. Other events can start even while it is running.

Parallel Process (Always) ... Always runs regardless of any condition. Other events can start even while it is running.

* When you call another event from a Parallel Process event, the called event is run as if it is a parallel event.

* Parallel events, as soon as their condition is no longer met, will have processing interrupted even if it is in the middle of the event. (However, as a side-effect of high-speed processing, it will continue through "commands that absolutely cannot wait" like setting variables.) They will also be interrupted if the map changes mid-event, and if the condition is still met, will resume from the start.

3. Condition

If the Run Condition is "Auto-Start" or "Parallel Process," you can edit this section.

The event will run only if the condition specified here is met.

4. Note

This message is displayed when calling the Common Event.

Use it to make notes about arguments and what the event does.

5. Export File

Export and import Common Events as .common format files.

- **Save One** ... Saves the currently-selected Common Event.
- **Save Multi** ... Saves multiple Common Events as one file.
- **Load** ... Loads Common Event(s) from a .common format file. Beware, the events previously there will be overwritten.

* If you click "Save One" while holding Shift, you can export in a text format.

Make use of this for making tutorials.

6. Checkpoints

You can place checkpoints in events. Use "**Add Checkpoint**" to add one, and when you click "**Next Checkpoint**," you can jump to the next one. They should prove useful to insert in events you are making, or for easily seeing the individual processes of a large event.

Also, if you check "Special Checkpoint," you can place and search for a different set of checkpoints from usual. Insert them in places like the latest parts of what you are working on.

7. Undo Button

Undoes one action done with C, V, X, and Delete. Any other actions (like adding event commands) are currently not undo-able. Use it if you accidentally press one of the keys and it makes a mess. You can undo up to 10 times.

8. Test Play

Starts a Test Play; identical in function to the Test Play button  in the Main Window.

9. Set Data Max

Change the number of Common Events. The limit is 10,000.

★. Argument Settings With "Set"

Press the "Set" button to display the following window. Here you can set the number of arguments, their contents, and return settings.

- Argument Names

If you set names for the arguments here, the names will be displayed when calling this [Common Event](#), making them easier to use.

- Special Specification

Set special ways of providing numeric arguments. You can choose from three types.

- No Special Specification

This is used by default. With this, you input numbers directly.

- Database Reference (Number)

Makes it so you pick a number while referencing the specified database type.

- Create Options (Number)

Create your own options.

- Return Result?

If you choose to "Return," then when the Common Event finishes, the value of the specified CSelf will be returned to the caller.

★ 2. Command Search

Click the "Search" button, and the following window will be displayed.

You can search for the next command in Common Events that contains the desired string/number.

[Search Term]

Input the string you want to search for in the box.

With "Search [variables/values]" selected, a number entry box will be displayed.

Clicking the Search button (or pressing Shift + Enter) while holding Shift will search in the reverse direction.

[Search Target]

- Search [command text]

Searches through the command text as displayed on the command list.

- Search [variables/values] in commands

Searches through variables/values specified in commands.

When you want to search for a variable, use the corresponding [variable reference value](#).

- Search [strings] used in commands

Searches through strings specified in commands.

Some things, like text used in string pictures, is cut off in the command text, but by using this search method, you can properly search the full internal string.

- Search [Common Event names]

Searches through Common Event names only.

[Search Domain]

- **Current Event Only** ... Only searches inside the currently-open event.

- **All Events** ... If not found in this event, it will continue to search through events one after another.

★ 3. CSelf Variable Usage List

Click the "CSelf Var Usage List" button to display the following window.

You can view which of the current Common Event's Common Self Variables are used and which ones are not. At the same time, it lets you give names to each Common Self Variable.

However, **this only lets you check those variables directly referenced using values 1600000~1600099, or chosen from the drop-down list.**

Variable calls such as \cself[?] in text will not be detected. Also, variables will be detected as used even if the data is never referred to, only set.

The screenshot shows a window titled "View Info" with a close button in the top right. Below the title bar is a subtitle "[CSelf Usage Overview] - = Unused | = In Use". The main area is a grid of 100 rows and 5 columns, each representing a CSelf variable from 00 to 99. Each row has a label on the left (e.g., "CSelf 00=") and an input field. Some input fields contain text: CSelf 00 (Item #), CSelf 01 (Amount [- to remo), CSelf 02 (Obtain Message), CSelf 08 (Item Name), and CSelf 09 (Unit). At the bottom of the window, there are six buttons: "Copy All", "Paste All", "Paste Into Blanks Only", "Reset Unused Variables Only", "Done", and "Cancel".

Also, with the **Copy All** and **Paste All** functions, you can copy all the Common Self names from one common to another.

Paste Into Blanks Only will use the copied names only for the fields that are currently blank in the target event. **Reset Unused Variables Only** will blank the names of Common Selves that are not used.

[Hidden Commands]

While on the Common Event List, the following inputs can be used to insert/remove Common Events.

- **Ctrl+Shift+F6**: Inserts a new Common Event.
- **Ctrl+Shift+F7**: Removes the selected Common Event.

However, using these functions will shift all Common Event numbers afterward.

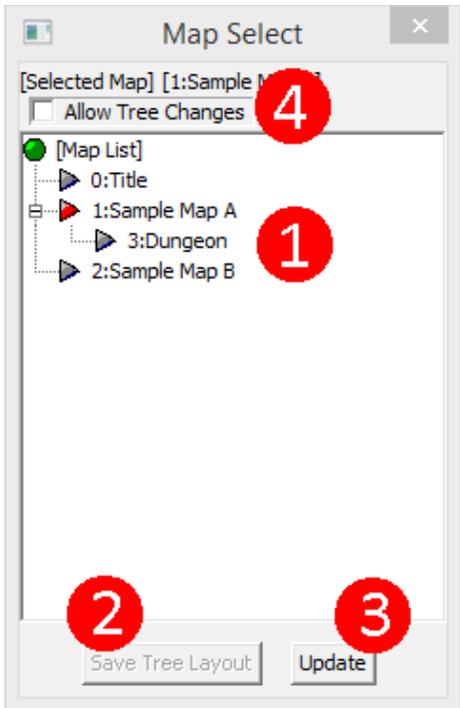
Using these has a very high chance of making your entire game break, so please be very careful with them.

There is one other secret shortcut for experts.

- **Ctrl+R**: Return to the previously-selected Common Event. Press repeatedly to come and go.
→ This shortcut is handy when copying parts from another event.

[Map Select]

You can open the Map Select Window by clicking this icon  in the Main Window.
(This window has completely changed as of Ver. 1.12.)



1. Map List

The list of maps is displayed in a tree layout.
You can do the following two commands.

- **Click Map Name:** Switches the map being edited in the Main Window to this.
- **Drag and Drop Map A onto Map B:** Makes Map B a child of Map A.

2. Save Tree Layout

Saves the tree layout after it has been changed.
If you close the Editor without clicking this, the tree layout will have reverted next time you open the Editor.

3. Update

When map names are changed or new maps are added, the list displayed in (1) will not update automatically.
Click this button to update the display data.

4. Allow Tree Changes

Allows modifications to be made to the tree layout.
If this is not checked, you cannot edit the tree.

If you Right-Click a menu item, you can use the following commands.

Create New Map	(N)
Copy Map	(C)
Paste Map	(V)
Delete Map	(D)
[←] Move to Parent Level	Shift+[←]
[→] Move to Child Level	Shift+[→]
[↑] Move Up	Shift+[↑]
[↓] Move Down	Shift+[↓]

- **Create New Map:** If the right-clicked map is not blank (no file), it creates a new map as that map's "child."
- **Copy Map:** Copies the right-clicked map.
- **Paste Map:** Pastes copied map as a "child" of the right-clicked map. The map file is copied to a separate filename. (If you copy Test.mps, it becomes Test_2.mps. This continues up to ~_9.mps.)
- **Delete Map:** Asks if you want to delete the right-clicked map's map file (.mps) as well. Deleted data cannot be brought back.
- [←] Move to Parent Level
- [→] Move to Child Level
- [↑] Move Up
- [↓] Move Down

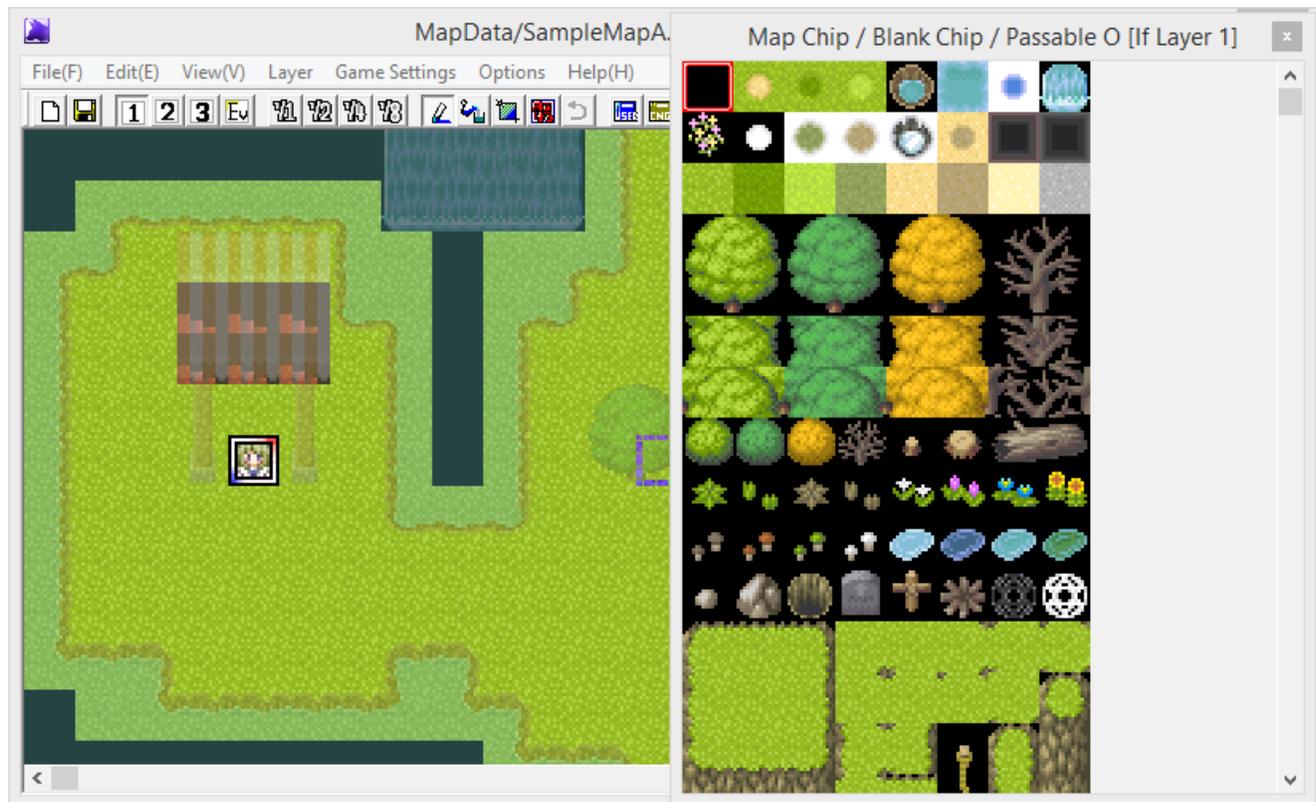
You can also use the following shortcuts.

- **N** ... Create New Map
- **C** ... Copy Map
- **V** ... Paste Map
- **D** ... Delete Map
- **Shift + ↑** or **↓** ... Moves the chosen map up and down (within the same depth).
- **Shift + ←** or **→** ... Moves the chosen map in the parent/child direction.

These functions are the same as those in the Right-Click menu.

[Map Creation Flow]

This explains three main elements of creating and using maps: creating a new map and registering it in the System Database, setting the game start position, and registering positions.



But first ...

[What is a Map?]

Maps in WOLF RPG Editor are made out of small square images referred to as "map chips." In-game, characters move on these maps.

[What are Layers?]

1 **2** **3** **Ev** Clicking these 1~3 buttons in the Main Window lets you select a layer.

The three layers can be considered three sheets of transparent film: you place map chips on these individual layers, and in the game, the three layers are stacked on top of each other to display a combined image.

The ordering of the layers is **1 (lowest) => 2 => 3 (highest)**.

The Event Layer, switched to using the Ev button, is a layer exclusively for placing events on.

Now, let us move to the next step.

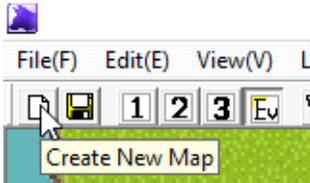
[\[1. Creating a New Map and Registering It\]](#)

[1. Creating a New Map and Registering It]

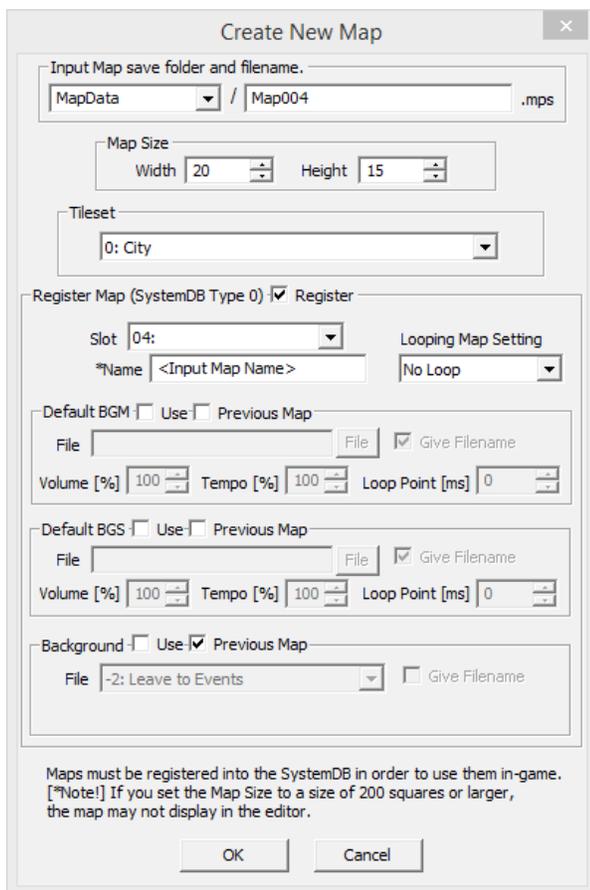
This explains how to create a new map and register it to the System Database.

In WOLF RPG Editor, maps must be registered in the System Database to be used in-game.

First, let us make a new map.



Clicking this icon on the left side opens the "Create New Map" window.

A screenshot of the 'Create New Map' dialog box. The 'Input Map save folder and filename.' field shows 'MapData' in a dropdown and 'Map004' in a text box with a '.mps' extension. The 'Map Size' section has 'Width' set to 20 and 'Height' to 15. The 'Tileset' dropdown is set to '0: City'. The 'Register Map (SystemDB Type 0)' checkbox is checked. The 'Slot' dropdown is set to '04:'. The '*Name' field contains '<Input Map Name>'. The 'Looping Map Setting' dropdown is set to 'No Loop'. There are three sections for BGM, BGS, and Background, each with 'Use' checkboxes and 'Previous Map' options. The 'Background' section has 'Use' checked and 'Previous Map' checked. The 'File' field for Background is set to '-2: Leave to Events'. At the bottom, there are 'OK' and 'Cancel' buttons. A note at the bottom states: 'Maps must be registered into the SystemDB in order to use them in-game. [*Note!] If you set the Map Size to a size of 200 squares or larger, the map may not display in the editor.'

[Map Save Folder and Filename]

Choose the folder you want to save in from the pull-down menu on the left, and type the filename you want to use in the text box on the right.

[Map Size]

Set the size of the map. The units are number of chips.

[Tileset]

Select the tileset to use to draw the map.

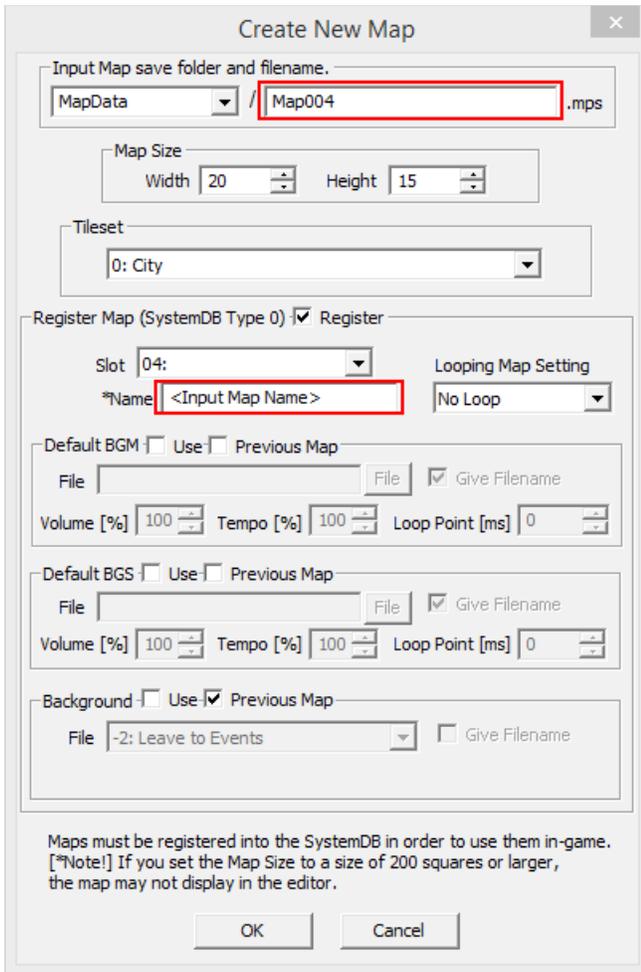
[Register Map]

Map files (.mps extension) normally cannot be used in-game unless their filenames are put into System Database Type 0. However, by specifying a slot to register to here, you can easily add them to System DB Type 0.

"Register" is checked by default, so just select a "Slot," a map name, looping settings, BGM, BGS, and background settings.

Note that the "map name" here has no usage in-game; its purpose is to help you distinguish maps when you are editing. Set it to whatever you want.

↓ Input Example, next page.



Fill in at least the parts in red boxes and press OK, and you will be all done with creating a new map and registering it to the System Database.

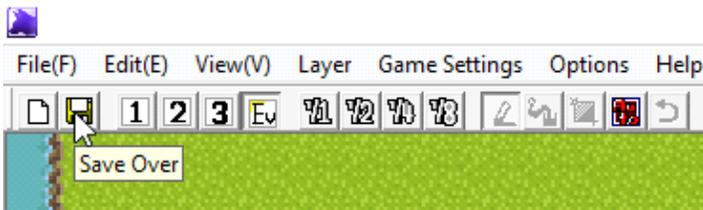
Even if you do not properly set up everything at first, you can change the settings later from



["Map Settings"](#) in the Main Window.

That is all for creating a new map and registering it, but this new map has nothing drawn on it, so let us make something.

Clicking the Layer Select buttons will display the Map Chip Window. Select map chips from there, and place them on the map like so.



Once you have drawn your map, click the Save icon to save it. Once you save, the "*" in the title bar will disappear.

This "*" indicates whether the map has been changed since it was last saved, so you will see it appear when you draw anything or create events.

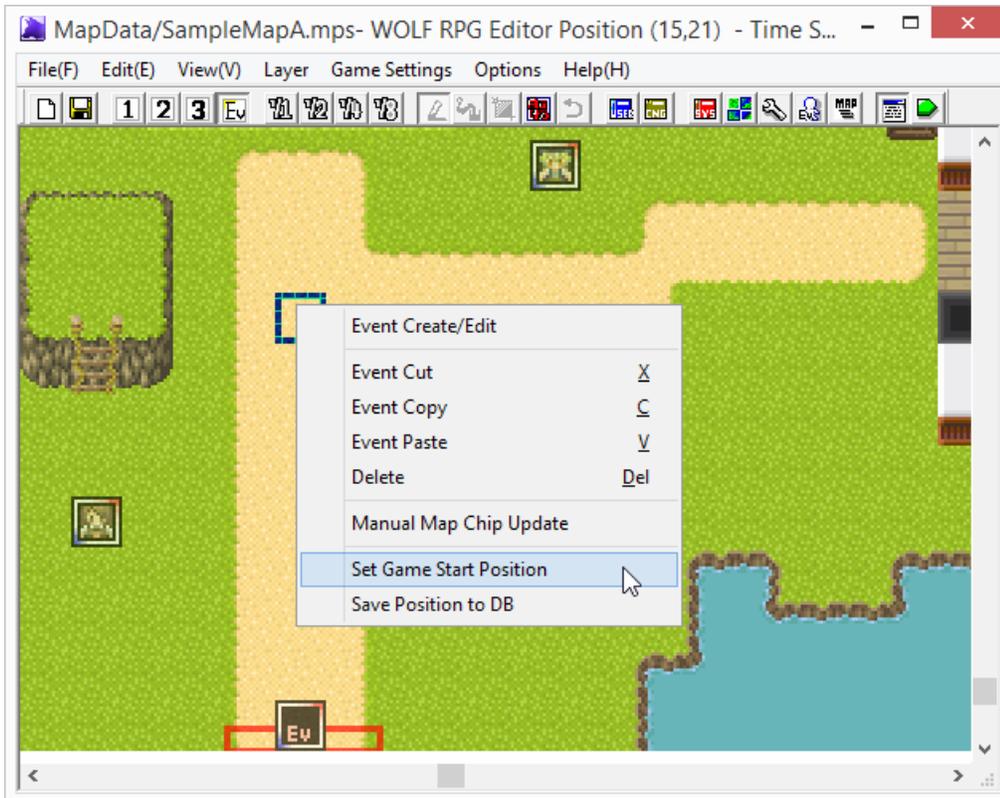
Through the process above, you can create a new map, register it to the System Database, and draw chips on it. Now, we will explain how to set the game start position.

[\[2. Set Start Position\]](#)

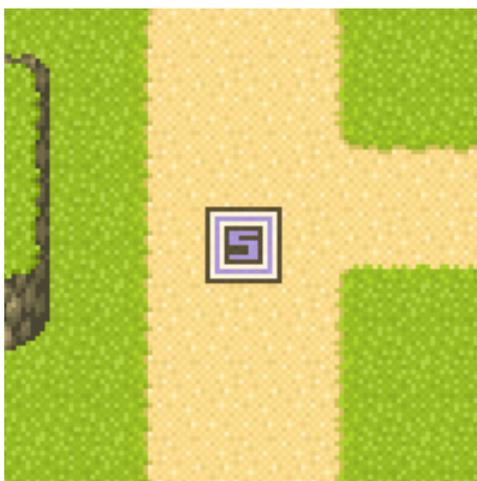
[2. Set Start Position]

This explains how to set the position you start from when you start a new game.

First, with the "Event Layer" selected in the Map Editor, Right-Click the spot you want to set as the start position. The following menu will appear.



From this menu, select "Set Game Start Position." The following image will then appear.



The square with the letter "S" represents the position where your hero (player-controlled character) will start in the game.

Your game's start position is automatically stored in **System DB, Type 7: Position List, Data 0: Starting Position.**

It will start from this position when you run **Game.exe**.

Do not delete Data 0: Starting Position!
It is critical for starting your game!

↓ After placing the start position, try clicking the "Test Play" button  to start the game.



And just like that, you will now start from the specified position.

Through the process above, you can set the game start position on any map, anywhere in the game(just make sure about the attributes of your map chips). You can also use this function to test different parts of your game without playing it from the “beginning.”

This is essential for making your games, but as a bonus, here are explanations of "Registering Positions" and the "Auto-Crete Map Function."

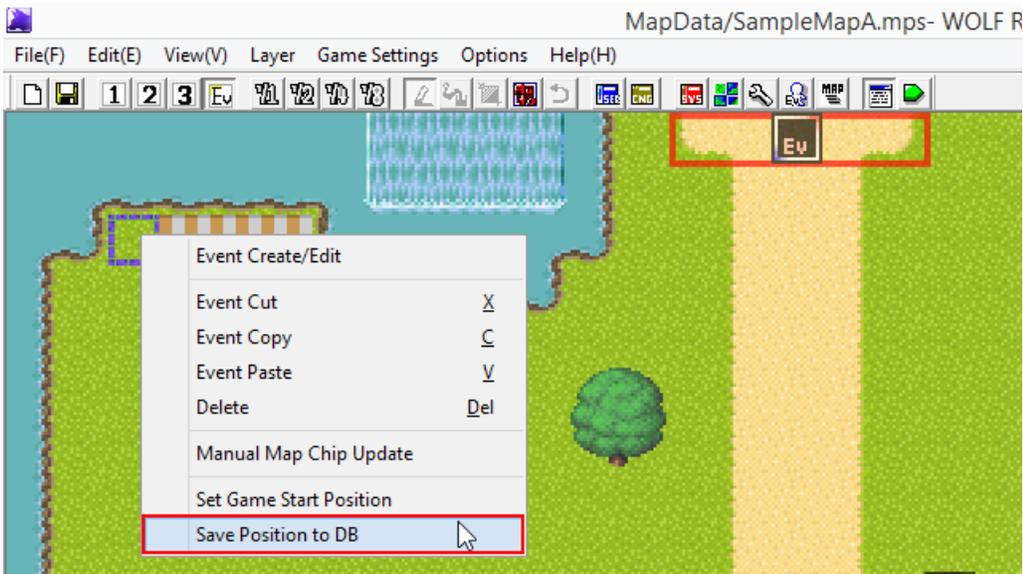
[\[Registering Positions\]](#)

[\[Auto-Crete Map Function\]](#)

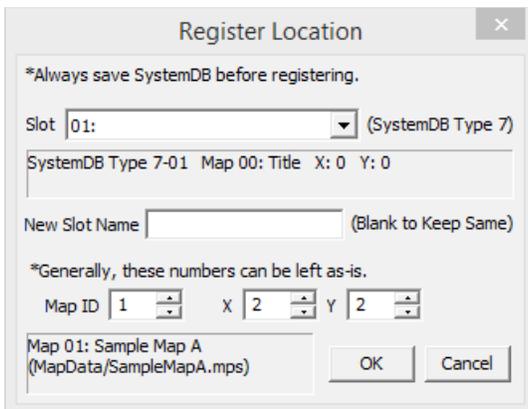
[A. Registering Positions]

WOLF RPG Editor lets you store not only a start position, but other positions which can be transferred to. This is a similar process to "2. Set Start Position."

With the Event Layer selected, Right-Click the position you want to register. The following menu will appear.



From this menu, select "Save Position to DB." The following window will be displayed.



[Slot]

Select where to save this position in System Database Type 7. The contents of this slot are displayed below (the part that says "SystemDB Type 7-01 Map 00: Title").

[New Slot Name]

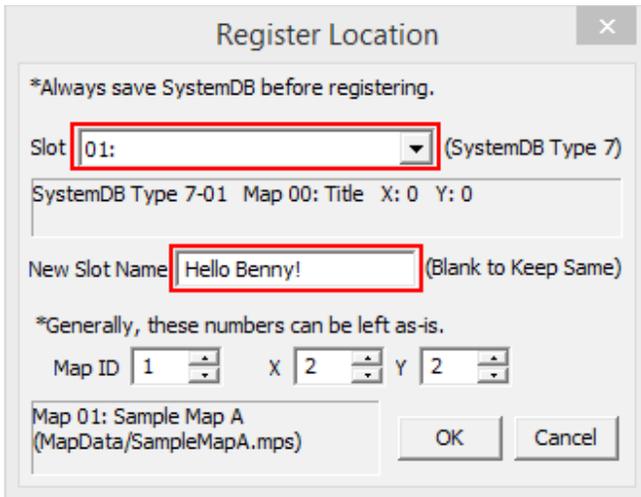
The name given for the position. It has no usage in-game; its purpose is to help you distinguish positions when you are editing. Set it to whatever you want.

[Map ID, X, Y]

No real need to mess with these.

* If a map is not registered in the System Database, **its Map ID cannot be automatically entered**, so registering it to the System Database from  "[Map Settings](#)" before you register positions in it is recommended.

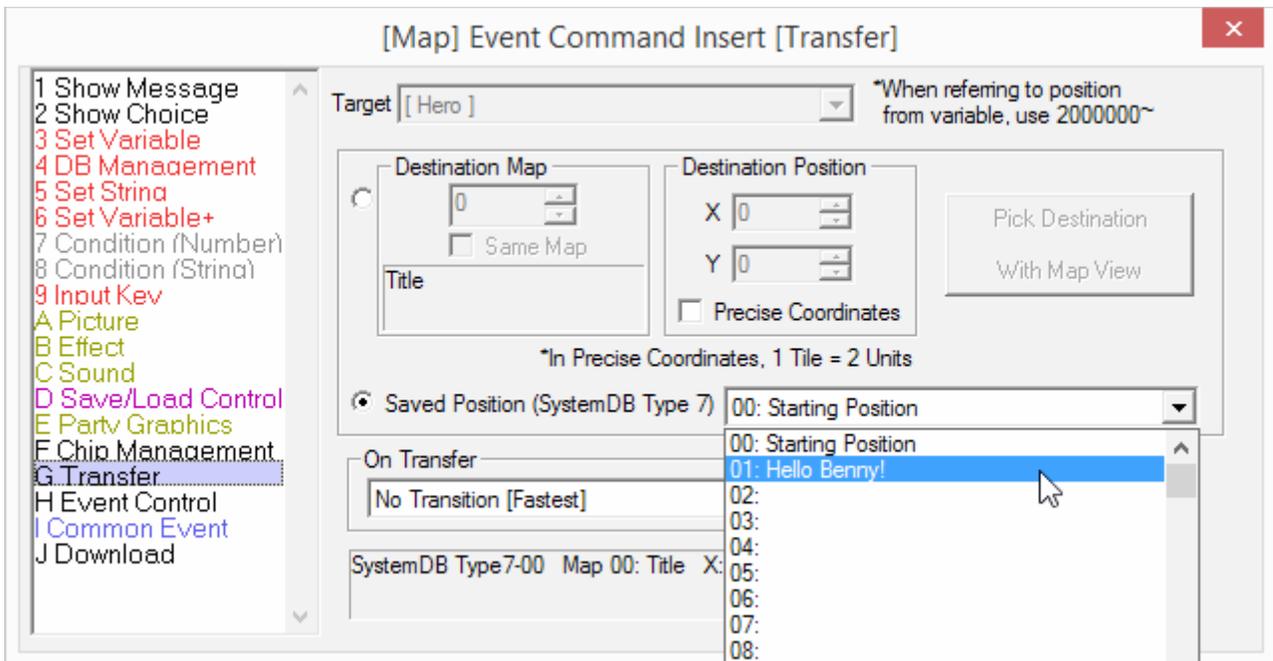
↓ Input Example



The destination slot was set to "01" (which is empty), and "Hello Benny!" was inputted as a new name for the position. Nothing else was changed. Now just click **OK**.

With this, you have registered a transfer position.

Using the following process, you can use registered positions as destinations in the event command "Transfer."



The newly-registered position "01: Hello Benny!" can now be selected. If it is not showing up after registering, you should check **SystemDB Type 7** and make sure it is there, and then click the "Update" button at the bottom of the System Database.

That is all for the explanations regarding maps.

[B. Auto-Create Map Function]

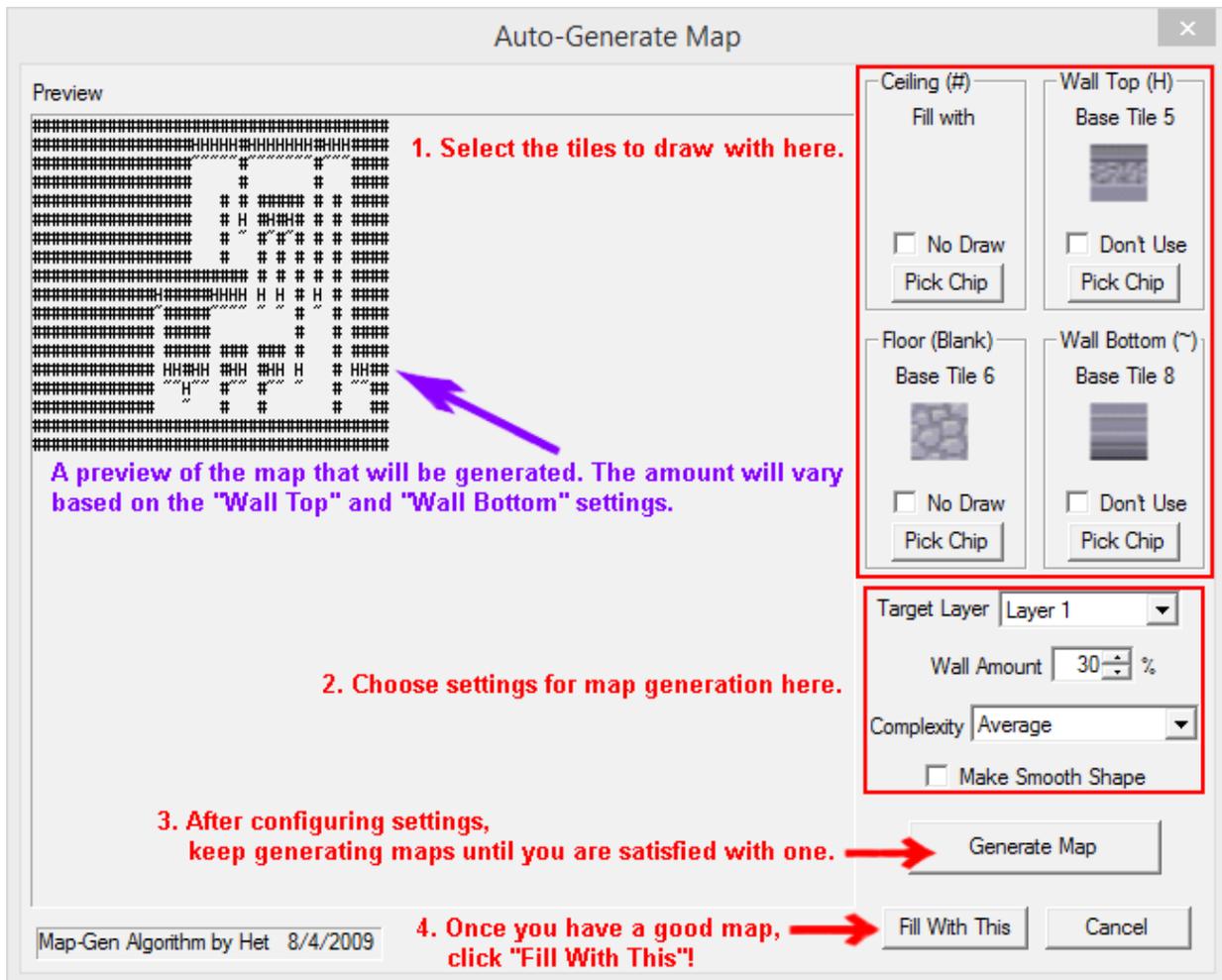
WOLF RPG Editor includes an automatic map creation function. When you are tired of making maps or lacking inspiration, it can be a useful aid.

To auto-generate a map:



Click this icon, and the below "Auto-Generate Map" window will appear.

For general usage, follow steps 1 to 4 below.

A screenshot of the 'Auto-Generate Map' dialog box. The dialog has a title bar with a close button. It contains a 'Preview' section on the left showing a map generated with '#' characters. To the right of the preview are four sections for selecting tiles: 'Ceiling (#)', 'Wall Top (H)', 'Floor (Blank)', and 'Wall Bottom (~)'. Each section has a 'Base Tile' selection, a 'No Draw' checkbox, and a 'Pick Chip' button. Below these are settings for 'Target Layer' (Layer 1), 'Wall Amount' (30%), 'Complexity' (Average), and a 'Make Smooth Shape' checkbox. At the bottom are 'Generate Map', 'Fill With This', and 'Cancel' buttons. Red arrows point to the 'Generate Map' and 'Fill With This' buttons. A purple arrow points to the preview area. Red text instructions are overlaid on the dialog.

1. Select the tiles to draw with here.

2. Choose settings for map generation here.

3. After configuring settings, keep generating maps until you are satisfied with one.

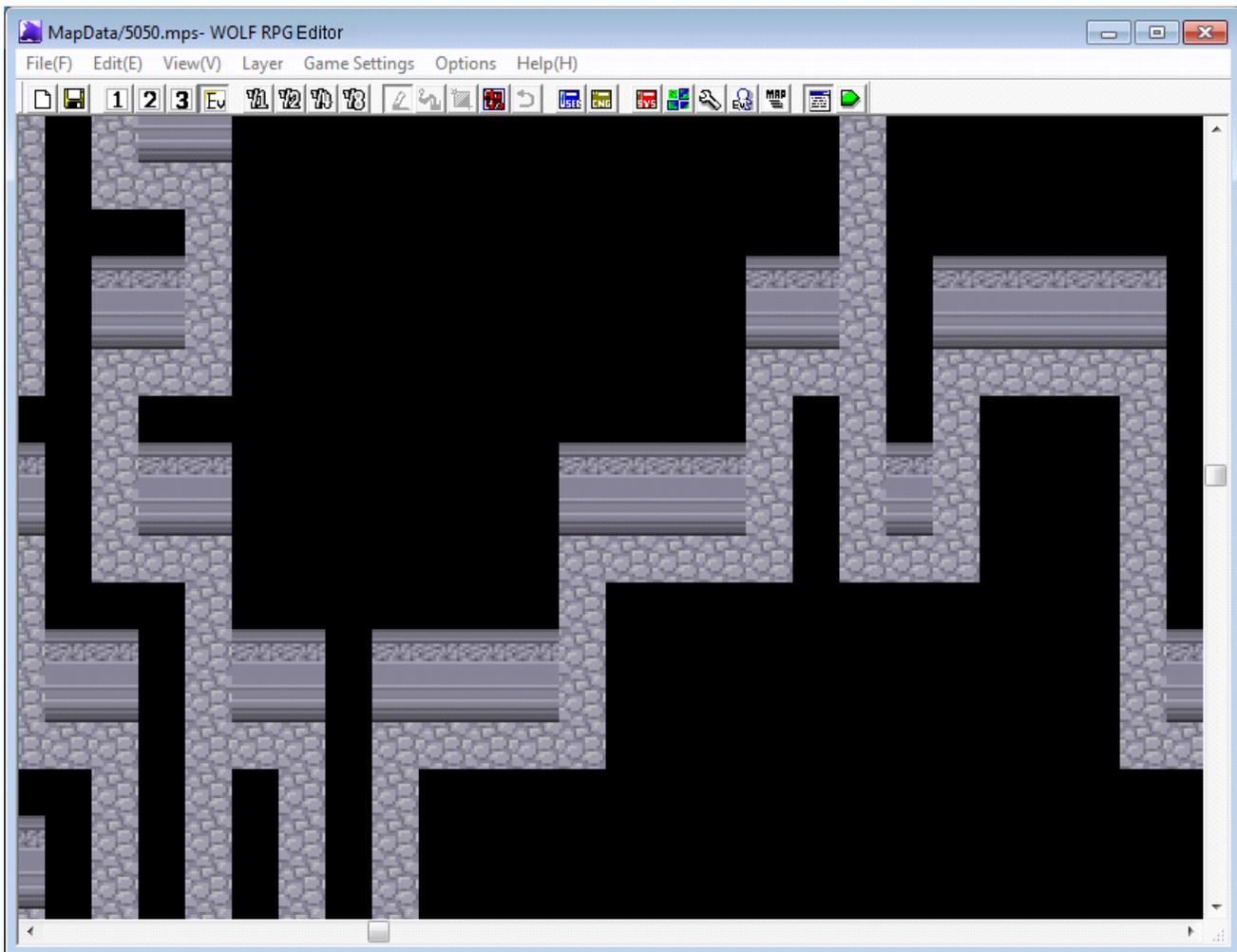
4. Once you have a good map, click "Fill With This"!

A preview of the map that will be generated. The amount will vary based on the "Wall Top" and "Wall Bottom" settings.

Map-Gen Algorithm by Het 8/4/2009

↓ Upon clicking "Fill With This"...

The map will be generated like this, as in the preview, on the specified layer.
However, this is still a rough map; you will probably want to edit it and add decorations to make it look nicer.



[Thanks] This map-generation algorithm is thanks to Het. Thank you for your assistance.
Het's Website: <http://d.hatena.ne.jp/Het/> (Japanese)

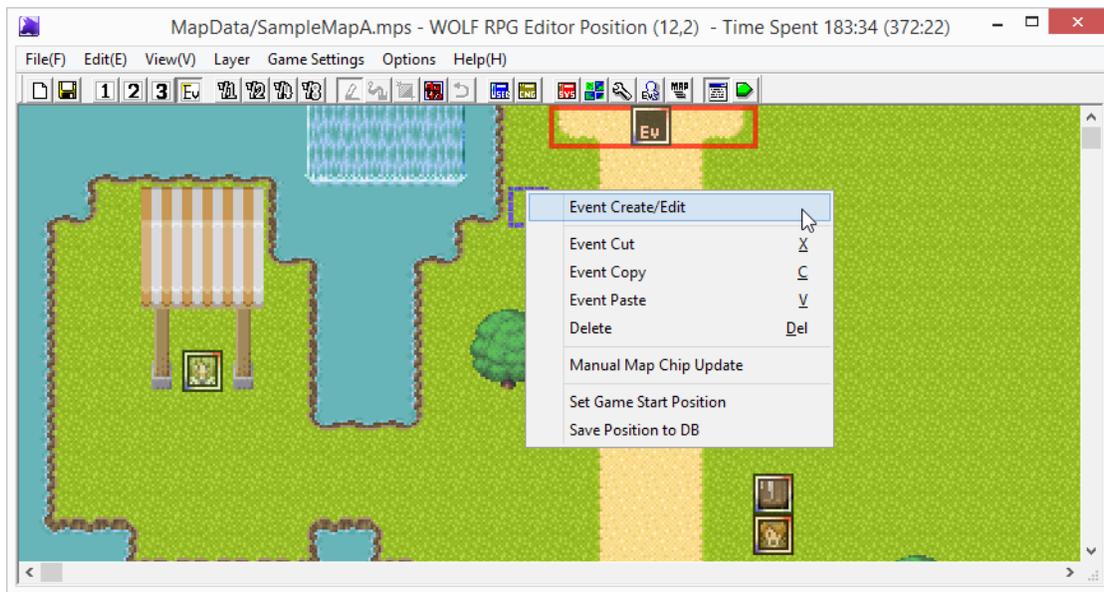
[Creating Events]

- [Basic Explanation](#)

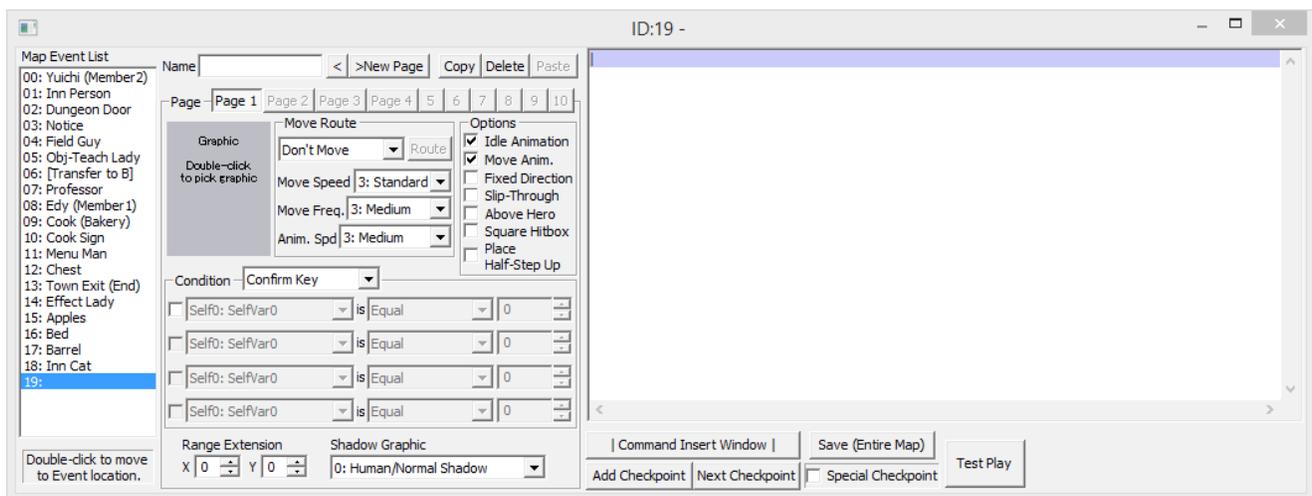
- [Tricks for Creating Events](#)

An "event" in an RPG can be a townspeople, a sign, a door, a chest, or anything else that causes some reaction when checked. This section describes the process of creating events.

To create an event, with the "Event Layer" selected on a map, Right-Click the place you want to place the event (you can also double Left-Click to immediately start editing an event).



The following menu will appear. Select "Event Create/Edit." This will display the following Event Edit Window.



Read this for detailed explanations of everything in this window.

[\[About the Event Edit Window\]](#)

[Tricks for Creating Events]

<Overlapping Events>

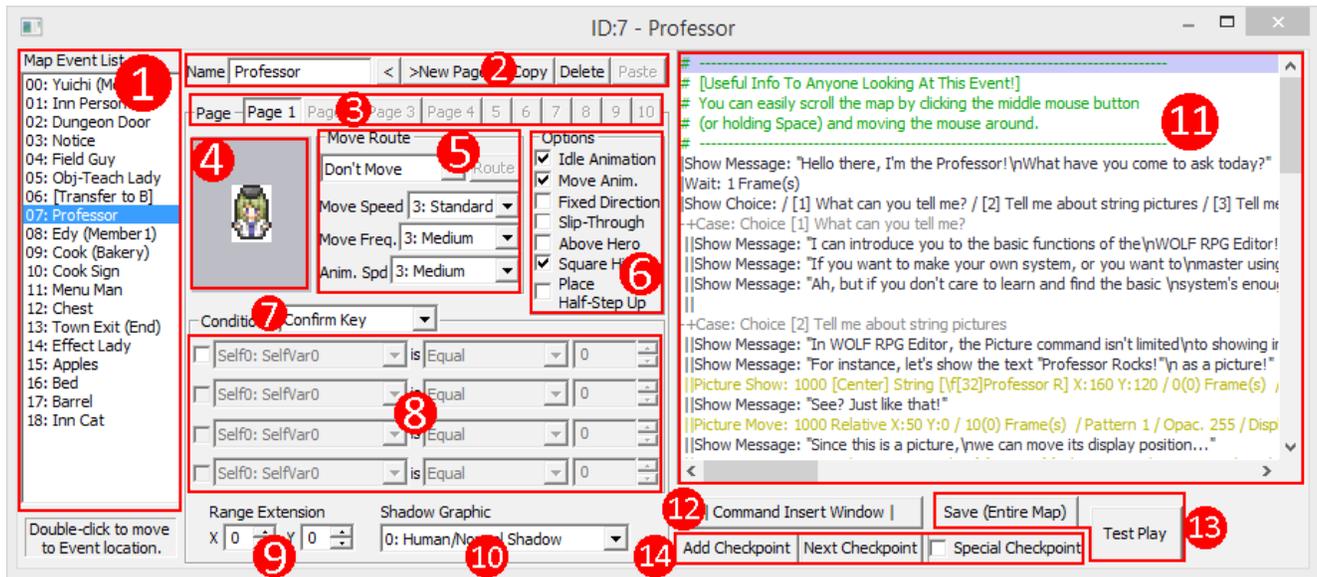
You can actually **place two or more events in the same coordinate** by dragging an event over another event on the map. To select and delete events in this case, use the [Event Edit Window](#).

When multiple events occupy the same position, their display order is as follows.

1. If "Above Hero" is on, it has priority (displayed highest). This is unchanged by "Slip-Through" being on or off.
2. If either "Above Hero" or "Slip-Through" is off, it is displayed with this priority.
3. If not "Above Hero" and "Slip-Through" is on, it is displayed lowest.

* When events are in the same priority, the one with the bigger Event ID is drawn on top.

[Event Window]



[Operation Inside the Event Window (11)]

- **Edit Event Commands ...** Double-click the event command you want to edit.

- **Controls for manipulating event commands:**

C: Copy

X: Cut

V: Paste

Delete: Delete

Space: Edit Event Command

Enter: Display Event Command Insert Window

T: Jump to Top of Selection Range

B: Jump to Bottom of Selection Range

For other shortcut keys shared with the Event Command Insert Window, consult "Event Command List."

- **Right-Click Menu**

Open Command Insert Window	Enter
Edit Command	Space
Cut	X
Copy	C
Paste	V
Delete	Del
Select All	A
Jump to Start of Selection	T
Jump to End of Selection	B
Copy Command String to Clipboard	Q
Copy Event Code to Clipboard	W
Paste Event Code from Clipboard	E

Open Command Insert Window: Same as (12).

Edit Command: Same as double-click.

Cut/Copy/Paste/Delete: Same as X/C/V/Delete.

Jump to Start of Selection: Shows start of current selection. Useful when condition cases get long.

Jump to End of Selection: Shows end of current selection. Useful when condition cases get long.

Copy Command String to Clipboard: Copies the text of the selected commands to the Windows clipboard.

Copy Event Code to Clipboard: Copies selected event commands to the Windows clipboard as text data that WOLF RPG Editor can process.

Paste Event Code from Clipboard: Pastes text data on the Windows clipboard as WOLF event commands.

As long as the format is correct, you can paste code from a browser or Notepad.

However, if you edit the contents to have unexpected parameters, it will not function properly.

* Code Example (If you copy the below text, from "WoditorEvCOMMAND_START" to "END", you can use "Paste Event Code" to paste it in as an event command. There cannot be any blank space at the top.)

```
WoditorEvCOMMAND_START
```

```
[103][0,1]<0>()("<dqrt>WOLF RPG Editor will continue to evolve!<dqrt>")
```

```
WoditorEvCOMMAND_END
```

[Section Explanations]

1. Map Event List

A list of all the events that exist in the current map. Clicking on an event name will open that event.

Double-clicking will move the map to the location of that event.

Also, you can cut/copy/paste/delete events from the Right-Click menu.

2. Name / New Page, Copy, Delete, Paste

Input a name for the event in the name field. If you do not put anything in, it will automatically use the graphic filename. It has no in-game meaning, so set it to whatever you like.

The "New Page/Copy/Delete/Paste" buttons are used to add and delete event pages. You can have from 1 to 10 pages, and events will give execution priority to the higher-number pages. Also, the "<" to the left of New Page adds a new page to the left of the current one, while ">New Page" adds a page to the right.

3. Page Buttons

Select the event page to edit.

4. Set Event Graphic

Double-click this area to select a character graphic.

Note that if you select a map chip to use as an event graphic here, the Shadow Graphic (10) will not be displayed. Shadow graphics will only be displayed for **character chips**.

5. Move Route

Select a movement route for the event. There are four options: "Don't Move," "Custom," "Random," and "Toward Hero."

- **Don't Move** ... Will not move at all.

- **Custom** ... Set up your own route. Click the "Route" button to display the "[Move Route Window](#)."

- **Random** ... Moves in random directions.

- **Toward Hero** ... Approaches the player character. **However, if too far away, it will move randomly instead.**

You can also set Move Speed, Move Frequency, and Animation Speed. Their effects are as follows.

- **Move Speed** ... As the name implies, the speed at which the event moves.
- **Move Frequency** ... Generally, events pause after moving 1 tile. Increasing Move Frequency lets you reduce the length of that pause. Setting it to "Every Frame" results in moving without stopping.
- **Animation Speed** ... The speed at which the event's animations advance. Setting it to "Every Frame" will make it wriggle at incredible speeds.

6. Options

Set event behavior and features. Checking these options will turn them on.

- **Idle Animation** ... Event will animate while not moving.
- **Move Animation** ... Event will animate while moving.
- **Fixed Direction** ... Event's facing direction will remain fixed.
- **Slip-Through** ... Makes it possible to pass through event.
- **Above Hero** ... Event will display above hero.
- **Square Hitbox** ... Makes the size of the event (hitbox) 1 tile by 1 tile. If off, it will be 1 tile wide by 0.5 tile high.
- **Place Half-Step Up**... If this option is on, the event will be placed 0.5 tile above where it is placed on the map.

7. Trigger Condition

Select the condition by which the event is triggered. There are five types.

- **Confirm Key** ... Starts the event if you press Confirm while on top of the event (if the event is not solid) or with the event directly in front of the hero. If the event is not solid, both halves of the hero must be overlapping it to trigger.
- **Autorun** ... Automatically executes when the conditions are met. While running, other events cannot run (except parallel events).
- **Parallel Process** ... Executes repeatedly as long as the conditions are met. Other events can start while it is running. Note that if a parallel process calls an event, that event is run as a parallel event.
- **Player Touch** ... Starts only when the player touches the event. For Player Touch, **the Hero's X position must be exactly the same as the event's.**
- **Event Touch** ... Starts when the player touches the event, or if **the event touches the player. The event will trigger repeatedly as long as the event and player are overlapping.** For Event Touch, as little as a quarter-tile of the Hero's hitbox must overlap the event.

8. Event Start Conditions

Input the conditions necessary to run the current event page. These conditions are things like "when a variable is a certain value" or "when it is above a certain value." You can set up to 4 variable conditions at once, and all of them must be met to trigger that event page.

9. Range Extension

For the case of the "**Confirm Key**," "**Player Touch**," or "**Event Touch**" Trigger Conditions, extends the trigger area for the event. Use this to make a single event that spans more than one tile.

10. Shadow Graphic

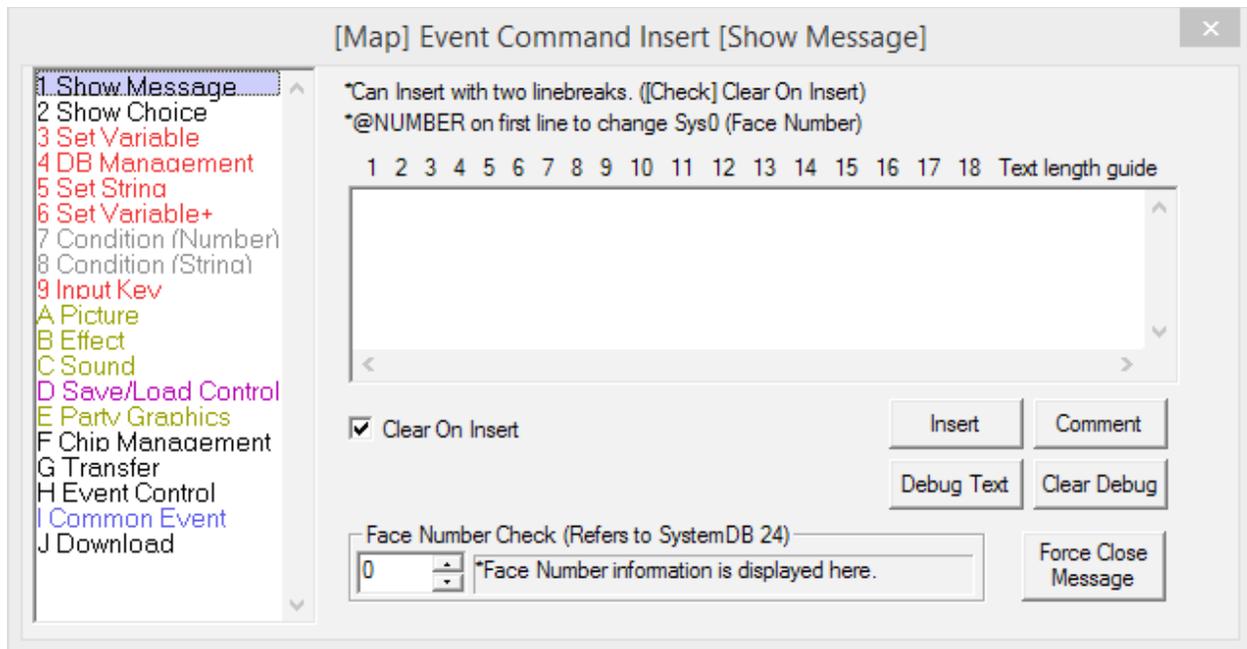
Select a shadow graphic (read from System DB Type 9). Has no real purpose if not using shadow graphics.
Note that if you selected a map chip as the event graphic, shadow graphics will not be displayed.

11. Event Command Display Box

All event commands for the current event and page are shown here.

12. Show Command Insert Window

Click the "**Command Insert Window**" button to show the following window for adding event commands.
 This is essential for making events.



13. Save (Entire Map) / Test Play

The "Save (Entire Map)" button saves the map you are working on.

The "Test Play" button starts a Test Play, just like the "Test Play" button  in the Main Window. When you change an event and want to test it, it is handy to click "Save" and then "Test Play."

14. Add Checkpoint / Next Checkpoint

Adds checkpoints among the event commands.

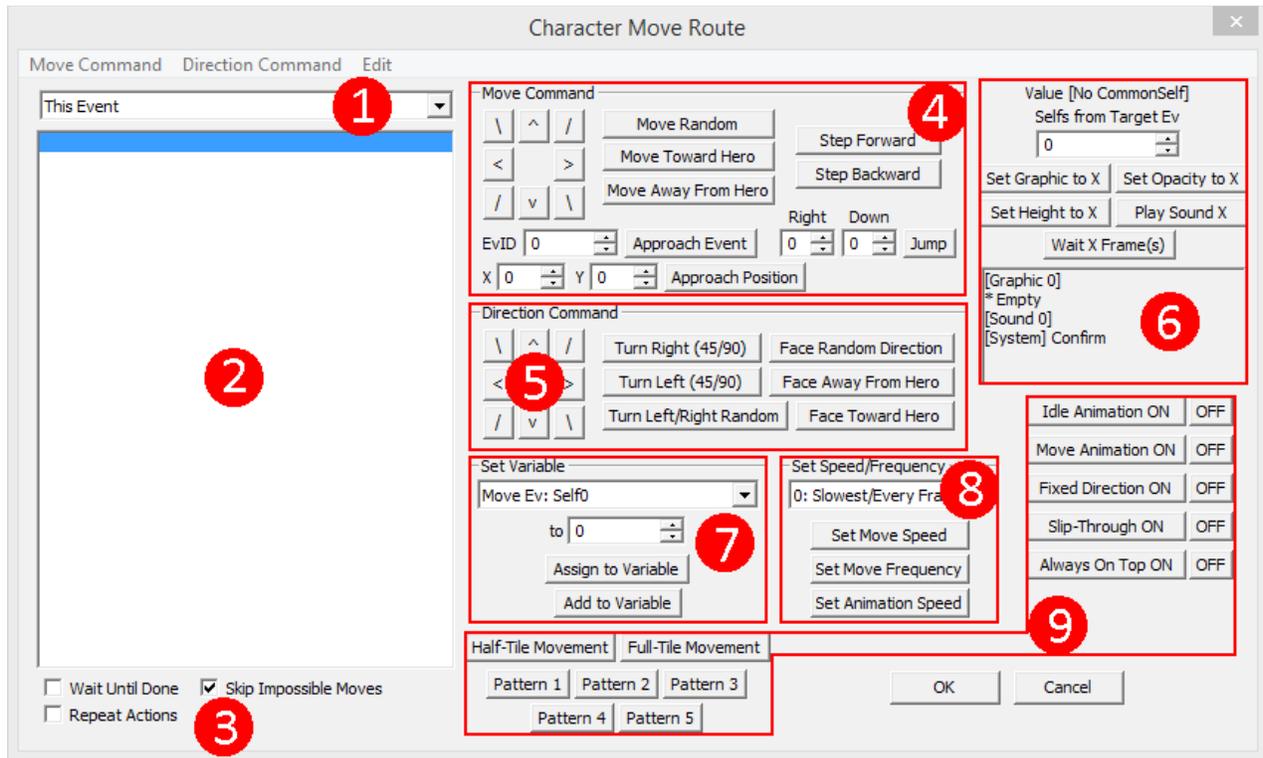
Clicking the "Add Checkpoint" button will add a checkpoint like so.



Click the "Next Checkpoint" button to move to the location of the next checkpoint. When making large events, placing checkpoints in important or work-in-progress sections can help development.

Also, if you check "Special Checkpoint," you can place and search for a different set of checkpoints from usual. Insert them in places like the latest parts of what you are working on.

[Move Route]



This explains the character Move Route Window. This window appears in two different places:

- In the Event Editor, setting Move Route to "Custom" and clicking the "Route" button
- In event command "Event Control," clicking the "|Move Route" button

[Move Route Window Controls]

- C**: Copy
- X**: Cut
- V**: Paste
- Delete**: Delete

[Section Explanations]

1. Target to Move

Select which event (or player character) to move. You can also specify the character to move via Self Variables or Normal Variables; variables are included at the bottom of the pull-down list.

When using Self Variables and Normal Variables to specify a target, the values match up to events like this:

- 0 or more** = The event with that ID
(Example: If the variable contains 2, Event ID 2 is moved)
- 1** = This Event
- 2** = Hero (Front of the line)
- 3** = Member 1 (Second in line)
- 4** = Member 2 (Third in line)
- 5** = Member 3 (Fourth in line)
- 6** = Member 4 (Fifth in line)
- 7** = Member 5 (Sixth in line)

2. Move Actions

A box showing the actions to perform.

3. Repeat Actions / Skip Impossible Moves

If "**Repeat Actions**" is on, the specified actions will be repeated over and over.

If "**Skip Impossible Moves**" is on, actions that cannot be completed are skipped. Be warned: If this is not checked, and a situation arises where the character cannot move, it will not move to the next action and the game may be stuck.

4. Move Command

A group of commands to move around the map. The eight direction buttons make the character take 1 step in that direction, and the rest are as written. Explanations for some special ones:

- **Jump** ... Makes the character jump the distance specified in the "Right" and "Down" boxes. A distance of 0 right, 0 down means jumping in place. To move left or up, put negative values in the Right/Down boxes.
- **Approach Event EVID[]** ... Take one step toward the specified event. If 8 directions are allowed, it can move diagonally, but if only 4 are allowed, it will follow with 4 directions only. Variables can be used in this field (example: input 2000000 to approach the event given by the value of Normal Variable #0).
- **Approach Position X[] Y[]** ... Take one step toward the specified position. Otherwise, works just like "Approach Event."

5. Direction Command

A group of commands used to change the event's facing direction. Here are the ones that warrant explanation.

- **Turn Right/Left (45/90)** ... Turns one "angle unit" to the right or left. For example, if using 8-directional graphics, it turns 45 degrees right, but with 4-directional graphics, it turns 90 degrees right.
- **Turn Left/Right Random** ... Like the above, this turns one "angle unit" right or left.

6. Set Graphic / Set Opacity / Play Sound / Wait

Performs these actions making use of the value in the box above. To change graphics, you must first set them up in System Database Type 8, and to play sounds, they must be set up in System Database Type 3. You can also specify variables in this field (example: input 2000000 to use the value of Normal Variable #0).

7. Set Variable

Use to change the value of a variable mid-Move Route. It is possible to use this for such things as triggering an auto-starting event once the route finishes.

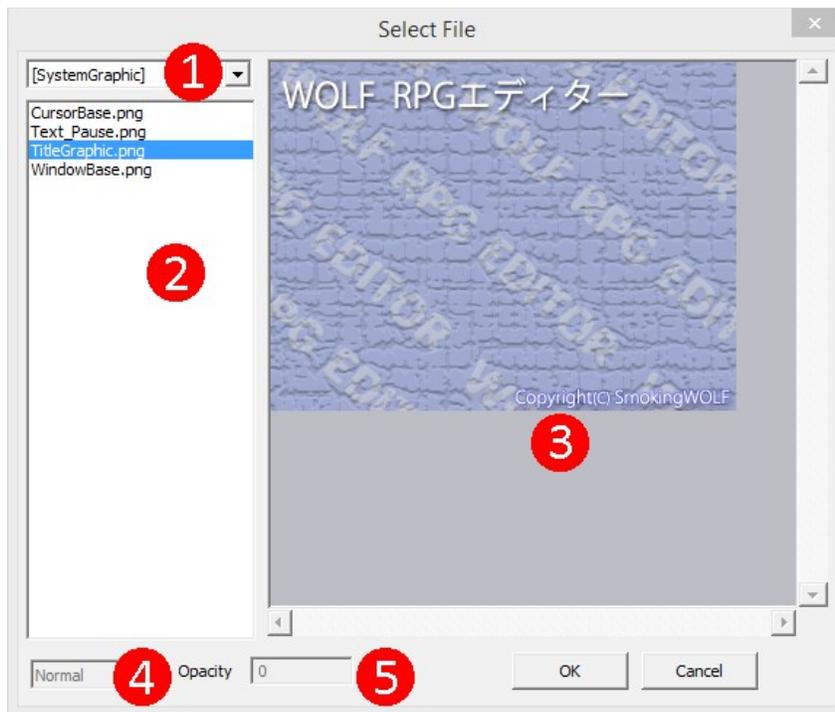
8. Set Speed/Frequency

Set Move Speed, Move Frequency, and Animation Speed. Refer to "[Event Window](#)" for their meanings.

9. Other Settings

A group of commands for changing event options, switching between half-tile/full-tile movement, and changing the event's animation pattern.

[File Select]



This explains the file selection window. This window is displayed in the following places:

- In the Event Editor, double-clicking the character graphic box
- In Basic Game Settings, clicking "Load" for Starting Hero Graphic
- In event commands, clicking "Load" or "File" buttons to specify files
- In Databases, clicking the "File" button for Fields defined in the Type settings as files to load

[Section Explanations]

1. Folder Select

Select the folder where the file to load is stored. You can only choose folders inside the "Data" folder.

2. File List

Click here to select a file. Sometimes only files of a certain extension are shown (example: when selecting an image file, only png, jpg, jpeg, and bmp files are shown).

Double-clicking will open the file in the appropriate application defined by Windows. For instance, MP3 files may be played in Windows Media Player.

3. Preview Window

Shows a preview of image files. Images with transparent colors or alpha blending will not display properly, but they will not have a problem in-game.

Also, when selecting character graphics in the Event Editor or Basic Game Settings, an inverted mesh border is displayed. This lets you choose the character's direction and initial animation pattern.

4. Image Display Type

Used only when selecting character graphics in the Event Editor. Works the same as Display Type in "[Picture](#)."

5. Opacity

Used only when selecting character graphics in the Event Editor.

Set the initial opacity of event graphics (255 is fully opaque, 0 is fully transparent.)

[Formula Explanation / About Variable Shorthand]

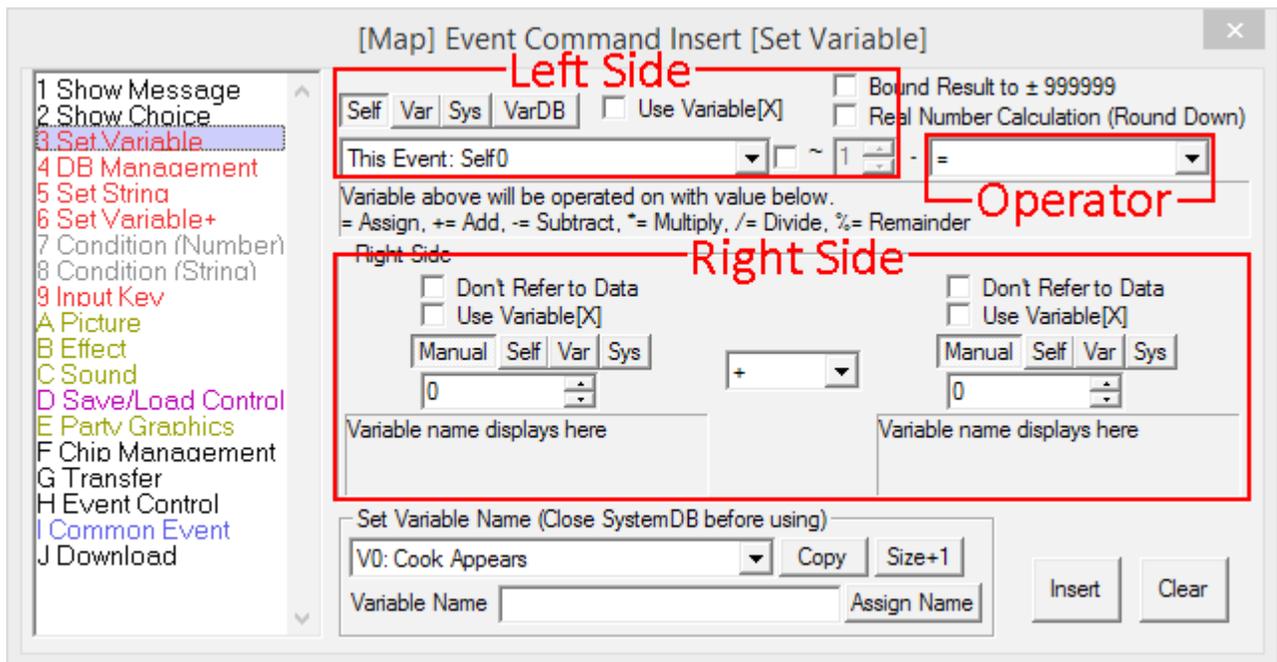
- [Formula Explanation](#)
- [About Variable Shorthand \(Such as V?, S?, Sys?, CSelf?\)](#)

[Formula Explanation]

This explains the assignment operators seen in some event commands.

First, some terminology: the destination of the assignment is called the "**Left Side**," the method of assignment is the "**Operator**," and the value being assigned is the "**Right Side**."

These can be seen in the event command "Set Variable" like so.



* Important note about calculations:

For the sake of optimization, WOLF RPG Editor does not perform a number-limiting process when using Set Variable in **Common Events**.

What does that mean, you ask? **If you add to a number causing it to go over the limit value of 2,147,483,648, the value will overflow, making it become -2,147,483,647.**

So please construct Common Events in such a way that values do not exceed ± 2 billion. This issue will not happen with Map Events.

→ Also, if any of the check-boxes in Set Variable are checked, or if you use variables other than Self Variables or Normal Variables, this "optimization" will not happen, so values will be limited to ± 2 billion.

(Assign Destination) (Operator) (Assign Value)
Left Side = Right Side

Operator	Name	Meaning
=	Assignment	Assigns the right side to the left side as-is.
+=	Addition	Adds the right side to the left side.
-=	Subtraction	Subtracts the right side from the left side.
*=	Multiplication	Multiplies the left side and the right side.
/=	Division	Divides the left side by the right side.
%=	Remainder	Assigns the remainder from division of the left side by the right side.
Lower Bound		If the right side is bigger than the left side, assigns the right side (the "lower limit") to the left side.
Upper Bound		If the right side is smaller than the left side, assigns the right side (the "upper limit") to the left side.
Absolute		If the right side is negative, changes it to positive and assigns to the left side. If the right side is positive, no change is made.

There are also these three operators exclusive to the "[Set Variable](#)" command.

Operator	Name	Meaning
Angle[x10] of Slope	arctan	The two values of the right side are the X vector and Y vector. Put the X and Y vectors of the slope into the right side, and the "slope of the angle (in degrees)" x10 will be assigned to the left side. (Note: For the X vector, right is positive, and for the Y vector, down is positive.)
sin[x1000] of Angle[x10]	sin	Put an angle in degrees x10 on the right side, and "the sine of that angle" x1000 will be assigned to the left side. (Example: If right side is 600 [60°], the left side will be set to 866 [sin60°=(√3)/2=0.866].)
cos[x1000] of Angle[x10]	cos	Put an angle in degrees x10 on the right side, and "the cosine of that angle" x1000 will be assigned to the left side. (Example: If right side is 600 [60°], the left side will be set to 500 [cos60°=0.5].)

Also, there are two special operators that are only used between the two values of the **right side**.

Operator	Name	Meaning
Bitwise &		Gets the result of "anding" the bits of the two right-side values. (Example: The bitwise & of 5 (101) and 3 (011) is 1 (001).)
~	Random	Gets a random value within the range of the two right-side values. "3~6" can return any of the values 3, 4, 5, or 6.

[About Variable Shorthand]

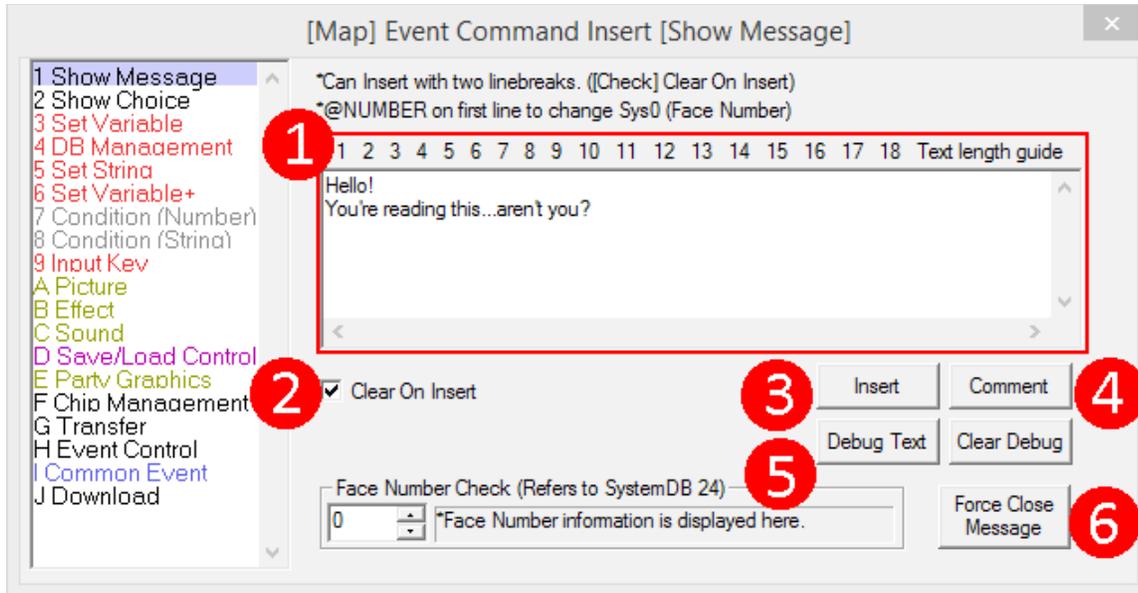
Parts of WOLF RPG Editor represent variables with abbreviations. (Example: **V1 = CSelf1 + 5**). This outlines what these variable abbreviations are. Note that the "?"s are places where a number goes.

Variable	Type	Meaning
V?	Number	Normal Variable #?
V?-?	Number	Spare Variable #?-?
S?	String	String Variable #?
Self?	Number	(Map Event's) Self Variable #?
CSelf?	Number/String	Common Self Variable #? #5~9 are strings, the rest are numbers.
Sys?	Number	System Variable #?
SysS?	String	System String #?

[Event Command List]

Event Command "Show Message"

An event command that displays messages.



[Section Explanations]

1. Message Input Box

Input the message you want to display here. You can also use [special characters](#) like `\v[?]` and `\s[?]` in the input.

2. "Clear On Insert" Checkbox

If this is checked, the message area will be cleared every time you click the Insert or Comment buttons.

3. "Insert" Button

Adds a message command using the text you entered. This message will be displayed in-game.

4. "Comment" Button

Writes the text you entered as a comment. Comments **are like notes** you can place among commands for the purpose of helping you in development, and have no effects when the game is running. It is recommended you add comments to complicated processes so you can more easily understand them when looking at them later.

Also, unlike normal messages, putting linebreaks in comments will make those linebreaks be reflected in the event command list.

5. "Debug Text," "Clear Debug" Buttons

Writes the text as a debug message and clears the debug window, respectively.

These work only when using the debug window.

To make it easy to trace debug messages, "MapEv#/CommonEv#/Line#" is automatically added to the end of each message.

* Using "Clear Debug -> Debug Text" repeatedly will essentially let you have information update in real time.

* Naturally, if you perform "Debug Text -> Clear Debug" in that order on the same frame, nothing will be displayed.

For details on the Debug Window, refer to "[Debug Command List](#)."

6. "Force Close Message" Button

A function useful only for parallel events. If this is called while a message is displayed, the current message is forcibly closed.

[Show Message Example]



* In the default state, no message box or face graphic is displayed; such systems must be made with Common Events. There is an example in the Sample Game, so you can refer to that.

[Special Functions]

- Face Graphic Number Input

If you put "@" followed by a number in the first line, then a linebreak, that value will be put in System Variable #0 "Face Number" right before the message is displayed. This allows you to both change the face graphic and display text in a single event command, simplifying your commands.

Usage Example:

```
@4  
Input it like this, and it will use Face Graphics #4.  
Though you need to have a Common Event to display faces!
```

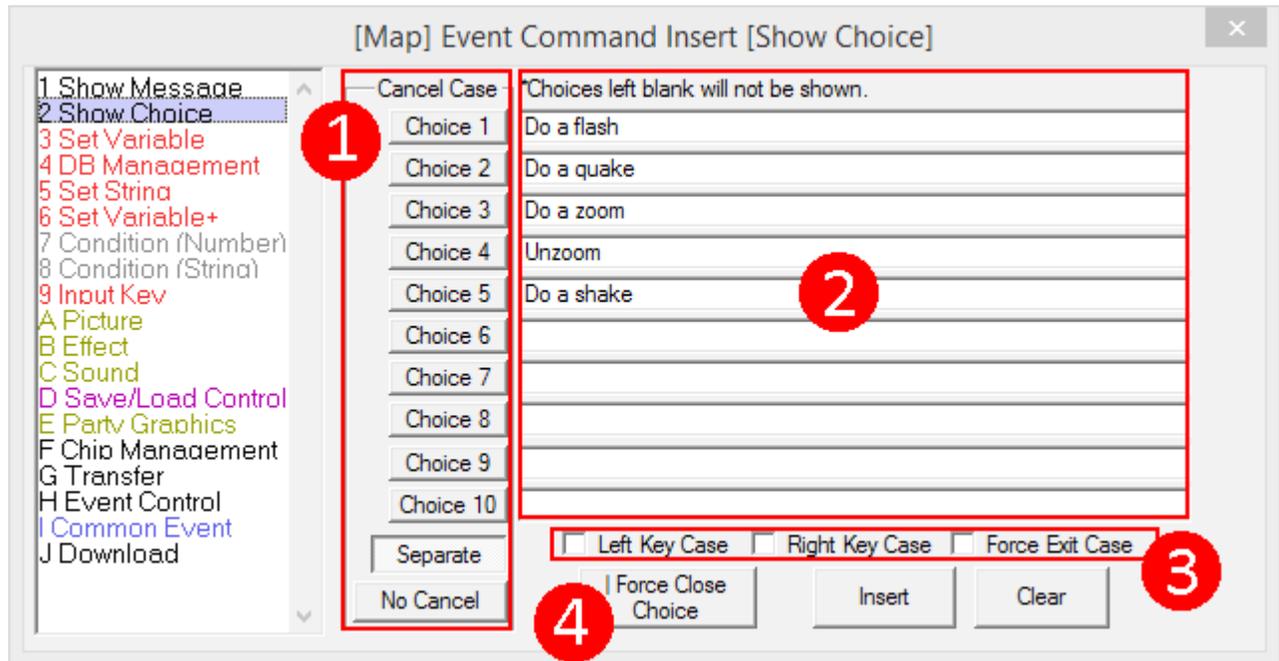
- Auto-Insert Via Double Linebreak

With "**Clear On Insert**" checked, if you **linebreak twice in a row**, the message will insert automatically. This can speed up entry of multiple messages, as you do not have to move your hands off the keyboard.

Also, if you **hold the Shift key while linebreaking twice**, the text will be inserted as a **comment**.

Event Command "Show Choice"

Displays up to 10 choices, from which the player can select.



[Section Explanations]

1. Cancel Case

Which choice case to go to when the Cancel button is pressed. "**Separate**" creates a separate case for pressing Cancel, and "**No Cancel**" prevents the choice from being canceled with Cancel.

2. Choice Input Boxes

Input the contents of each choice. If blank, that choice will not be selectable in-game. You can also use special characters like `\s[?]` and `\v[?]` (?=number) in these.

3. Left Key Case / Right Key Case / Force Exit Case

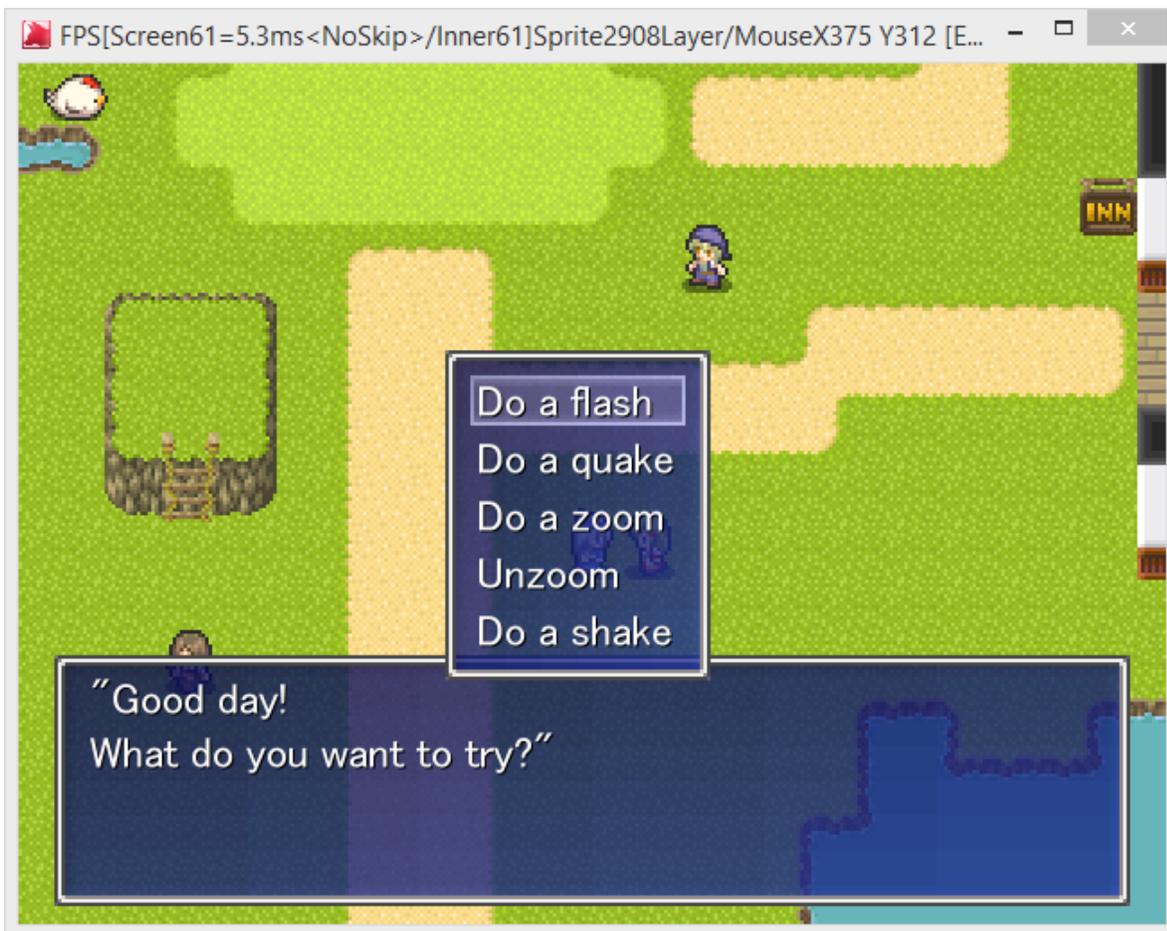
If "**Left/Right Key Case**" is on, a special case is used if that key is pressed while displaying choices. Think of it as the "**Left/Right**" version of the separate Cancel case.

"**Force Exit Case**" lets you define a case for when a "**| Force Close Choice**" occurs.

4. | Force Close Choice

Forcibly closes choices being displayed. **This command must be executed from a separate parallel event.** To name a use example, you could use it for a quiz with a time limit.

[Show Choice Example]



As long as choice graphics are set up, choices are automatically displayed in a window.

[Special Functions]

- **If the string is empty, that choice is removed.**

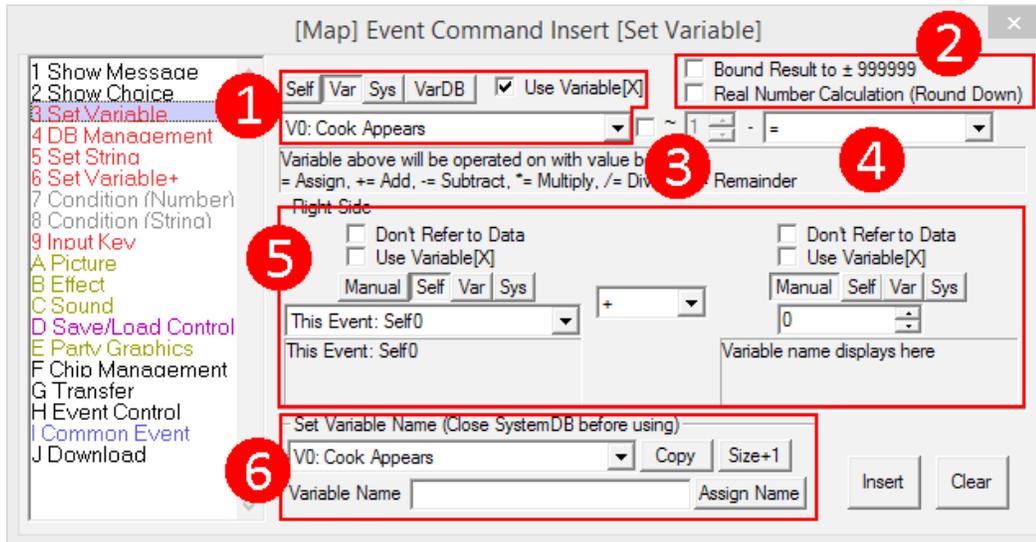
When nothing is written in the text box, that choice does not show at all in-game and cannot be selected.

This also holds true when special characters like `\s[?]` (displays String #?) are used and result in an empty string.

Thus, when used properly, it is possible to toggle whether certain choices are included.

Event Command "Set Variable"

Changes the value of a specified variable or does calculations.



* Important note about calculations:

For the sake of optimization, WOLF RPG Editor does not perform a number-limiting process when using Set Variable in **Common Events**.

What does that mean, you ask? If you add to a number causing it to go over the limit value of 2,147,483,648, the value will overflow, making it become -2,147,483,647.

So please construct Common Events in such a way that values do not exceed ± 2 billion. This issue will not happen with Map Events.

→ Also, if any of the check-boxes in Set Variable are checked, or if you use variables other than Self Variables or Normal Variables, this "optimization" will not happen, so values will be limited to ± 2 billion.

[Section Explanations]

1. Assigned-To Variable

Select the variable to assign to.

- **Self/Var/Sys/VarDB** ... Select the type of the assigned-to variable. From left to right: Self Variables, Normal Variables and Spare Variables, System Variables, and the Variable Database.

- **Use Variable[X]** ... Uses "variable #[value of specified variable]."

For instance, if you specify "Normal Variable #0," and the value stored in Normal Variable #0 is 2000005 (represents Normal Variable #5), the variable that is ultimately assigned to is "Normal Variable #5."

2. Set Variable Options

- **Bound Result to ± 999999** ... Makes it so that values will not cross over into the domain of variables (1,000,000 or more). Since the value of a variable crossing 1,000,000 would make it refer to the value of another variable, this check is meant to prevent that. If this is on, resulting values 1,000,000 or more change to 999,999, and values -1,000,000 or lower change to -999,999.

- **Real Number Calculation (Round Down)** ... If this is on, the internal calculation will temporarily use real numbers instead of integers, thus accounting for decimals. For example, when calculating "**100 * = 3/2**", turning real number calculation on will use "**100 * = 1.5**", resulting in 150. With real number calculation off, it will internally become "**100 * = 1 (with integer arithmetic, 3/2 is treated as 1)**", so the result is 100. This would mainly be key for things like battle formulas.

3. ~ (Range Option)

Turn this on to process multiple variables all at once. For instance, if you select "V0 (Normal Variable #0)" as the starting point and enter "4" for the range, V0 through V4 will all be processed at the same time.

4. Operator

Set the assignment method. You can select "=", "+=", "-=", "*=", "/=", "%=", "Lower Bound," "Upper Bound," "Absolute," "Angle[x10] of Slope," "sin[x1000] of Angle[x10]," or "cos[x1000] of Angle[x10]."

For detailed explanations, refer to "[Formula Explanation](#)."

5. Assignment Values

Set the value to assign here. You can use two variables or integer values at once.

If you specify a String Variable, that string is converted to a number. However, only strings that begin with a number can be properly loaded in like this.

- **Manual/Self/Var/Sys** ... Select the type of value to load. If you use Manual and just put in the reference codes for variables, you do not have to change modes.

- **Don't Refer to Data** ... Will not refer to variables when values are 1,000,000 or greater, instead treating them as normal integers.

- **Use Variable[X]** ... Same as "Use Variable[X]" in (1).

6. Set Variable Name

This section lets you quickly assign names to variables and increment the variable maximum. Select the variable you want to configure from the pull-down list, enter the name to assign, and click **Assign Name**.

The "**Copy**" button copies the name of the current variable in the pull-down list to the "**Variable Name**" box below.

The "**Size+1**" button adds one to the maximum of the current variable type. For instance, if **V3-1** (#1 in Spare Variables 3) is showing, Size+1 will add one to the size of Spare Variables 3.

When the name entry text box is selected, you can use the following shortcut keys.

- **Up/Down** ... Change variable you want to change the name of.

- **Enter Key** ... Set name; same as clicking the Assign Name button.

- **Shift + ↓** ... Copy target variable's name into name box; same as clicking the Copy button.

* **"Clear" Button** ... Resets the contents.

[Special Functions]

- You can convert strings to numbers.

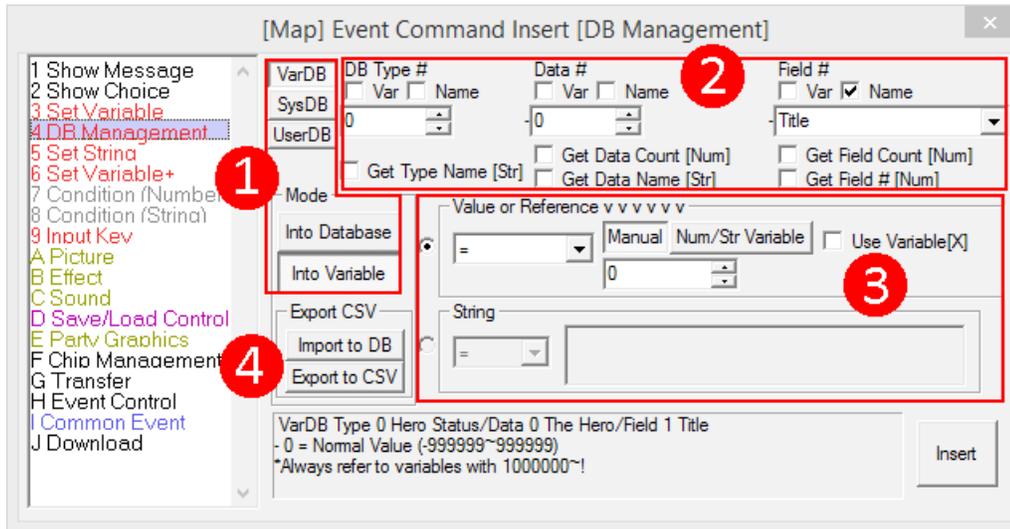
If a String Variable (example: 3000000/String Variable #0) is given as a Right Side value, that string can be converted to a number and used. However, these strings must begin with a number to be read properly.

(Examples: The string "234" becomes 234 as expected, but "82g1A" becomes 82, and "A42" becomes 0.)

* This feature should be useful when reading data from external text files,
or as a way of storing large amounts of numbers in a single String Variable.

Event Command "DB (Database) Management"

Can read values from each database, or change values in the Variable Database.



[Function Explanations]

1. "VarDB/SysDB/UserDB" Buttons and "Into Database/Into Variable" Buttons

Select which database to act on, and which process to use. "Into Database" only works on the Variable DB, so "Into Variable" is always selected when using the other databases.

2. Database Target

Specify the database **type number**, **data number**, and **field number** you want to process (or read from). You can also specify variables like 2000000 (Normal Variable #0) in these.

If you specify a type number and check "Get Data Count" or "Get Field Count," the variable will be assigned the number of data entries/fields in that type.

You can provide numbers and "Get Type Name," "Get Data Name," or "Get Field Name." You can also provide names and "Get Type #," "Get Data #," or "Get Field #."

If you specify a type number and choose "Reset All Data," that Variable DB type will be completely reinitialized. Providing a type number and a data number and choosing "Reset All Fields" lets you reset only a specific data entry. Note that the values used for initialization are those defined as default values in "[Type Definition](#)" (which may differ from the values at game start).

Finally, if you check "Name" above each input field, you can refer to database types/data/fields by name. If the Data ID Specification in the Variable DB is set to "Same As First String Data," use the initial name it has at game start; this will work regardless of changes made during the game. Be warned that you cannot refer to variables with special characters. (For example, if you input "\cself[5]", it will not search for the contents of Common Self 5, but rather the string "\cself[5].")

3. Variable to Process

Specify the variable to read from (or act on).

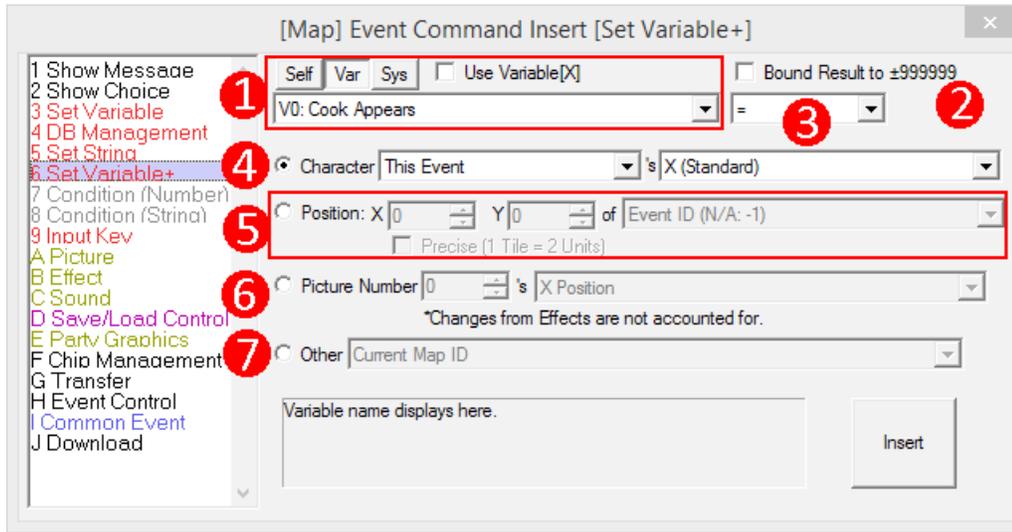
For the meaning of operators like "=", refer to "[Formula Explanation](#)."

4. Import CSV to DB / Export to CSV

Allows you to save the data of a database to a CSV file mid-game, and read data from a CSV file. (A CSV file is a comma-separated text data file.) You can even overwrite the User DB, but if you are calling data entries by name, things can behave strangely. This is because the connection between data names and data numbers is only made once at game start for optimization's sake, so changes are not reflected.

Event Command "Set Variable+"

Reads special values into the specified variable, and does calculations.



[Section Explanations]

1. Assigned-To Variable

Select the variable to assign to.

- **Self/Var/Sys** ... Select the type of the assigned-to variable. From left to right: Self Variables, Normal Variables and Spare Variables, and System Variables.

- **Use Variable[X]** ... Uses "variable #[value of specified variable]."

For instance, if you specify "Normal Variable #0," and the value stored in Normal Variable #0 is 2000005 (represents Normal Variable #5), the variable that is ultimately assigned to is "Normal Variable #5."

2. Set Variable Options

- **Bound Result to ±999999** ... Makes it so that values will not cross over into the domain of variables (1,000,000 or more). Since the value of a variable crossing 1,000,000 would make it refer to the value of another variable, this check is meant to prevent that. If this is on, resulting values 1,000,000 or more change to 999,999, and values -1,000,000 or lower change to -999,999.

3. Operator

Set the assignment method. You can select "=", "+=", "-=", "*=", "/=", "%=", "Lower Bound", "Upper Bound", or "Absolute". For detailed explanations, refer to "[Formula Explanation](#)."

4. Assignment Value: "Character"

Get info about heroes or other event characters, such as their position. You can retrieve the following data:

- **X/Y Pos (Standard)** ... The X/Y coordinate, in 1-tile units.
- **X/Y Pos (Precise)** ... The X/Y coordinate, counting 1 for every 0.5 tiles. The coordinate of the top-left 1x1 tile the character is standing on.
- **Height Off Ground (pixels)** ... The event's current height in pixels (can be set with character state variables starting from 9100000~).
- **Direction (1~9, Matches Numpad)** ... The character's facing direction. For mapping of directions to numbers, refer to Chart A below.

- **Screen X/Y** ... The current screen coordinate in pixels. For example, if the character is in the center at 320x240 resolution, X is 160 and Y is 120.
- **Shadow Graphic #** ... The number of the shadow graphic set for the event.
- **Current Tile Tag** ... The tag number of the tile at the event's location.
- **Event ID** ... The target event's ID. Only useful when target is "This Event."
- **On Screen? (1=YES 0=NO)** ... Returns whether the target is currently within ± 1 tiles of the screen (at 100% map zoom). If inside that area, the value is 1. Could be used for parallel events where you do not want to process off-screen events.
- **Active Page 0=None 1~=Page #** ... The target's currently-running event page. If there is not one running, returns 0.
- **Run Condition (0: Confirm - 4: Ev Touch)** ... The run condition for the target's active event page. 0 = Confirm Key, 1 = Autorun, 2 = Parallel Process, 3 = Player Touch, 4 = Event Touch.
- **Range Extend X/Y** ... The target's range extensions.
- **Animation Pattern [0-4]** ... The current pattern number of the current animation. Three-frame animations range from 0 to 2; five-frame animations range from 0 to 4.
- **Moving? [1=YES 0=NO]** ... Returns 1 if the character is mid-movement (between tiles).

7	8	9
4		6
1	2	3

Chart A

5. Assignment Value: "Position"

- **"Precise" Checkbox** ... If turned on, the coordinates you give are considered to be in half-tile units (2 units = 1 tile).

Get info about the specified X/Y position on the map. You can retrieve the following data:

- **Event ID (N/A: -1)** ... The ID of the event at this position. If no event exists there, -1 is returned. When using precise coordinates, returns event ID if the character's top-left position matches the event's precise location.
- **Passable (Tile Only)** ... The passability of the tile at this position.
- **Passable (Tile + Event)** ... The passability of the position, accounting for both events and tiles. In short, determines "whether the hero can pass through."
- **Uppermost Chip # (0: Clear, 1-15: Auto, 16-: etc)** ... The chip number at this position on the uppermost tile layer. Transparent chips on Layer 2/3 are ignored. 0 is a transparent chip, 1~15 are auto-tiles, and 16 onward are basic tiles.
- **Layer 1/2/3 Chip #** ... The chip number at this position on that layer.
- **Uppermost Tile Tag** ... The "tag number" of the tile at this position on the uppermost layer. Layers on which the tag is 0 are ignored.
- **Layer 1/2/3 Tile Tag** ... The "tag number" of the tile at this position on that layer.

6. Assignment Value: "Picture Number"

Get info about the specified picture number. You can retrieve the following data:

- **X Position/Y Position**
- **Image Size (Width/Height)** * This is the size at 100% zoom.
- **Pattern #**
- **Opacity**
- **Angle**
- **Zoom (Width/Height)**
- **Mouse Cursor Hover? (1=YES)**
Returns 1 when the mouse cursor is inside the picture's area.
- **Picture In Use? (1=YES)**
Returns 1 if the picture number is being displayed, and 0 if it is cleared.
It is useful for checking if that picture number is already being used.

- **String Done Displaying? (1=YES)**

If the picture is a string picture, returns 1 if the string is done fully displaying.

If it is still in the middle of displaying, returns 0; if it is not a string picture, returns -1.

- **[Free Mode] Top-/Bottom- Left/Right X/Y**

If the picture is being drawn in free transform mode, you can get its various position values. If the picture does not exist, -1 is returned; if it is not in free transform mode, 0 is returned.

* Be warned that the picture info you get here ignores adjustments made with "[Effect](#)."

[* Mouse Cursor Hover? (1=YES) Supplement 04/13/2010]

This check is dependent on when the **screen redraws**,

so when frameskip is used, the info may update a few frames late.

Also, due to the nature of the function, if you move a picture on the same frame using a Process Time of 0, this will not be reflected either. Keep these things in mind when using.

7. Assignment Value: "Other"

Get miscellaneous info. You can retrieve the following data:

- **Current Map ID**

- **Playing BGM/BGS #**

- **BGM/BGS Position (ms / For MIDI, Ticks)**

- **Current BGM/BGS Length (ms / For MIDI, Ticks)**

- **Mouse Left/Middle/Right Click State (Normal = 0, Hold = 1, Press = 2)**

The state of each mouse button. When the button is not held, it returns 0.

When clicked, the value returned is 2 for that single frame; if it continues to be held afterward, the value will be 1.

- **Mouse Wheel Delta (Change from Last Frame)**

How much the mouse wheel was spun since the last frame.

Be careful about checking for values 2 or more, as it may differ depending on the player's environment.

- **Mouse X/Y Delta (Change from Last Frame)**

How much the mouse X/Y position changed since the last frame.

- **Event ID at Mouse Position (Smaller ID Priority, -1 for None)**

The ID of the event the mouse cursor is currently over. If there is none, returns -1.

If an event's "Square Hitbox" flag is on, its ID may be returned if the cursor is over a 1 tile x 1 tile hitbox.

If the flag is off, the area of its hitbox is 0.5 tiles high x 1 tile wide.

If there are multiple events at the cursor position, the event with the lowest ID is chosen.

- **ID of This Map Event (In Common, Caller's ID)**

As it says, the ID of the Map Event that is processing this command.

When called in a Common Event, it returns the ID of the source Map Event that called it.

- **Map Size (Width/Height)**

The size of the current map.

- **ID of This Common Event (-1 Outside Common)**

As it says, the ID of the Common Event that is processing this command.

If it is not a Common Event, it returns -1.

- **Active Auto/Touch/Confirm Event ID [+500000 For Common]**

Call this in a parallel event to get the ID of the currently-running auto/touch/confirm-triggered event.

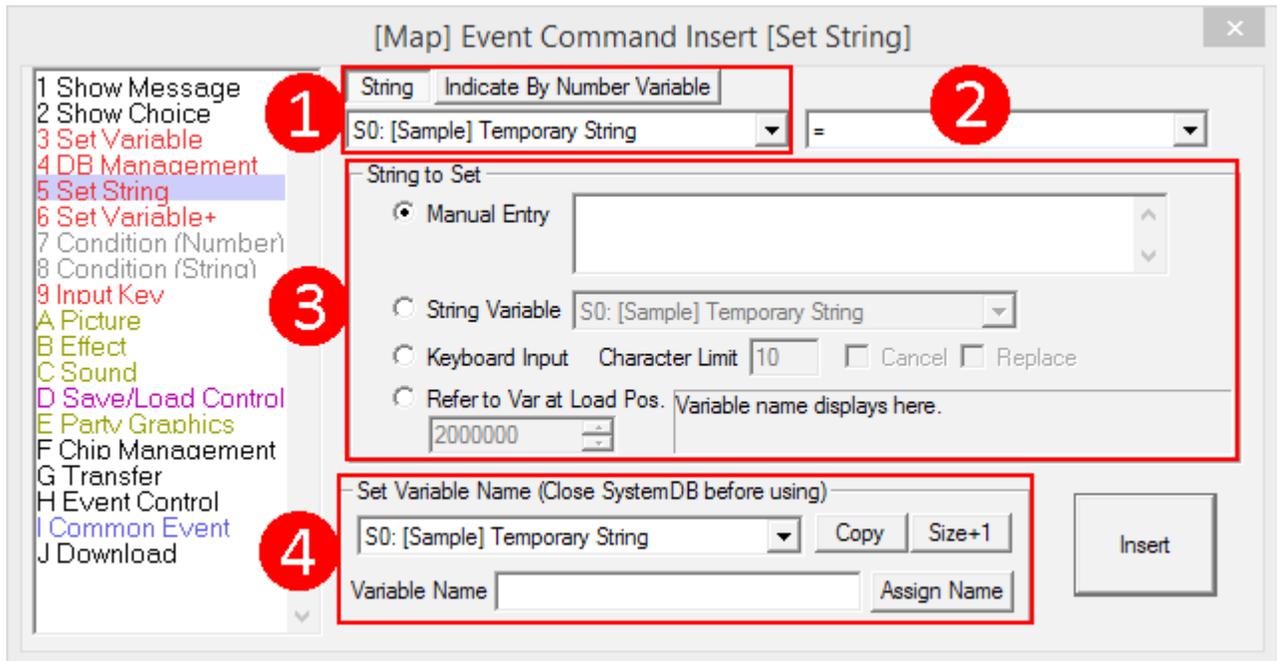
If said event is a Common Event, the value is 500000 + the ID.

- **Active Event Line [Auto/Touch/Confirm Events]**

Call this in a parallel event to get the number of the line being processed in the currently-running auto/touch/confirm-triggered event.

Event Command "Set String"

Manipulates String Variables, and can take typing input from the keyboard.



[Section Explanations]

1. Assigned-To Variable

Select the String Variable to assign to.

When using "Indicate By Number Variable," put a value 3000000 (String Variable #0) or more in a Normal/Spare Variable.

2. Assignment Method

The methods of assignment to the variable are as follows.

The assigned-to variable is referred to as the "left side," and the string being assigned is the "right side."

"=" ... Sets the left side to the right side as-is.

"+=" ... Appends the right side to the end of the left-side String Variable.

"Copy First Line Of..." ... Sets the left side to just the top line of the right side.

"Cut First Line Of..." ... Cuts the top line of the right side, and sets the left side to it. In other words, this removes the first line from the right side.

"Cut First Character Of..." ... Cuts the first character of the right side, and sets the left side to it.

Special codes are a special case; cutting once from \A+ will get "\A+", cutting once from \c[1] will get "\c[1]", etc. Codes that contain strings like \r[X,A] will cut only the "\r[" part.

"Load File Contents..." ... Loads the file with the filename given by the right side, and stores its contents in the left-side String Variable. Use the format "Data\test.txt" (using two "\s is safer) for the right side.

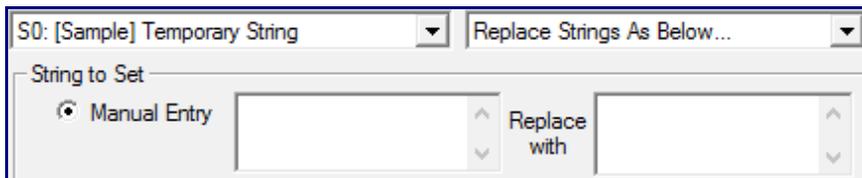
"Export Variable to File..." ... Saves the contents of the left-side String Variable to a file named by the right side.

"Get File List of Folder..." ... Gets a list of files in the folder given by the right side, and sets the left side to it. Each file has its own line in the stored string, so use "Cut First Line Of..." to process individual file names.

Note that encrypted folders are not supported (they will return <<ERROR>>).

"Remove Instances of String..." ... Removes all instances of the right-side string found in the left-side string. (Example: If left side is "aeiou", removing all instances of "i" makes the left side "aeou".)

"Replace Strings As Below..." ... Replaces all instances of the left input with the right input within the left-side string. (For example: If the left side is "aeiou", replacing "e" with "o" makes the left side "aoiou". When reading a comma-separated data file (CSV), you can use this to replace ","s with linebreaks, allowing you to then process line by line.)



"<Cut From Variable Up To String...>" ... Cuts out the contents of the left-side string, up to the string entered in the box. (Example: If left side is "aeiou," cutting up to "i" makes it "ae.") If the specified string is not found, "<<NotFound>>" is returned.

"<Cut From Variable After String...>" ... Cuts out the contents of the left-side string, starting after the string entered in the box. (Example: If left side is "aeiou," cutting after "i" makes it "ou.") If the specified string is not found, "<<NotFound>>" is returned.

* With skilled use of the file export and cut line/character functions, you can trade character data with others!

3. String to Set

The string to set can be specified in one of four ways.

- **Manual Entry** ... Manually enter text. You can use special codes like \v[?] or \s[?] here.
- **String Variable** ... Read another String Variable.
- **Keyboard Input** ... Asks the player to type in text with the keyboard. If you enable Cancel, you can cancel with the Esc key. Japanese can also be entered into this input field. If you check "Replace," the input field starts out having the string that was already in the left side.
- **Refer to Variable at Load Position** ... Goes to read the String Variable at the value of the given variable.

For example:

If the value stored in the variable is 3000002 (code for String Variable #2), it goes to read String Variable #2.

* In Test Play only, during a Keyboard Input, pressing F7~F9 to show a debug window or pressing F11 to reload data **may change the font size of the input text**. However, the font size is set "at the start of Keyboard Input," **so this is proper behavior**. (Added 3/7/2017)

4. Set Variable Name

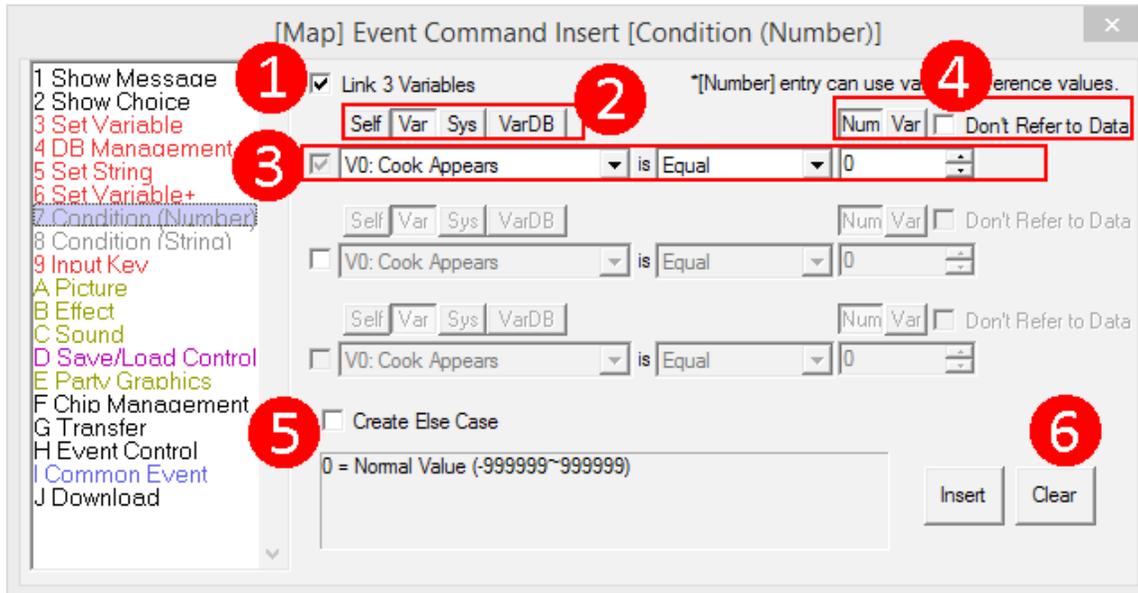
This section lets you quickly assign names to String Variables and increment the maximum. Select the variable you want to configure from the pull-down list, enter the name to assign, and click **Assign Name**.

The **"Copy"** button copies the name of the current String Variable in the pull-down list to the "Variable Name" box below.

The **"Size+1"** button adds one to the maximum of the String Variable type.

Event Command "Condition (Number)"

Splits event commands into a maximum of 3 cases based on the state of numeric variables.



[General Explanation]

You can set up to three conditions for cases. Note that condition checks are done from top to bottom. So for example, with these two conditions:

- **V0 is greater than or equal to 3**
- **V0 equals 3**

When V0 = 3, the first case (V0 = 3, so it satisfies "greater than or equal to 3") is chosen, and the second case is ignored. In other words, this is an example of what not to do.

[Section Explanations]

1. Link 3 Variables

If this is checked, all three of the variables in the pull-down for each condition will be made the same as the top variable.

2. Set Comparison Source: Self/Var/Sys/VarDB

Select the type of the comparison variable. From left to right: Self Variables, Normal Variables and Spare Variables, System Variables, and the Variable Database.

Note: When using the Variable Database, you cannot use variable-calling values (such as 1600000).

3. Define Conditions

Set the conditions for each case here. You can use variables in the comparison value boxes (example: 2000000 to compare with the value of Normal Variable #0). There are seven types of comparison.

- **Greater than ...** Take this case if variable is higher than comparison value.
- **Greater or equal ...** Take this case if variable is higher than or the same as comparison value.
- **Equal ...** Take this case if variable is the same as comparison value.
- **Less or equal ...** Take this case if variable is lower than or the same as comparison value.
- **Less than ...** Take this case if variable is lower than comparison value.
- **Not equal ...** Take this case if variable is not the same as comparison value.
- **Bitwise & ...** Take this case if result of bitwise "and" between variable and comparison value equals the comparison value.

* Explaining "**Bitwise &**" is difficult, so here is an actual example.

First of all, you need to have some knowledge of how numbers are represented in base-2 (binary).

When the variable's value is expressed in base-2, its 1 and 0 bits are compared with the comparison value. Here, Variable V0 contains 5 (101 in base-2), and the comparison value is set to 1 (001 in base-2).

Variable Value **101** (5)

Comparison Value **001** (1)

In this case, the "1" bit in the 1's place of the comparison value is also present in the variable, so the case succeeds. However, if the comparison value were 3 (011 in base-2)...

Variable Value **101** (5)

Comparison Value **011** (3)

Since the "1" in the second digit is not present in the variable, the condition is not met in this case, and it will move on to check the other conditions.

By the way, if you have a bitwise & condition using 0 (000 in base-2), naturally, it will always be met.

4. Set Comparison Value: Number/Variable

Select the type of the comparison value.

Also, the "Don't Refer to Data" checkbox makes it so the value will not refer to a variable when it is 1,000,000 or greater; instead, it is treated as a normal integer.

5. Create Else Case

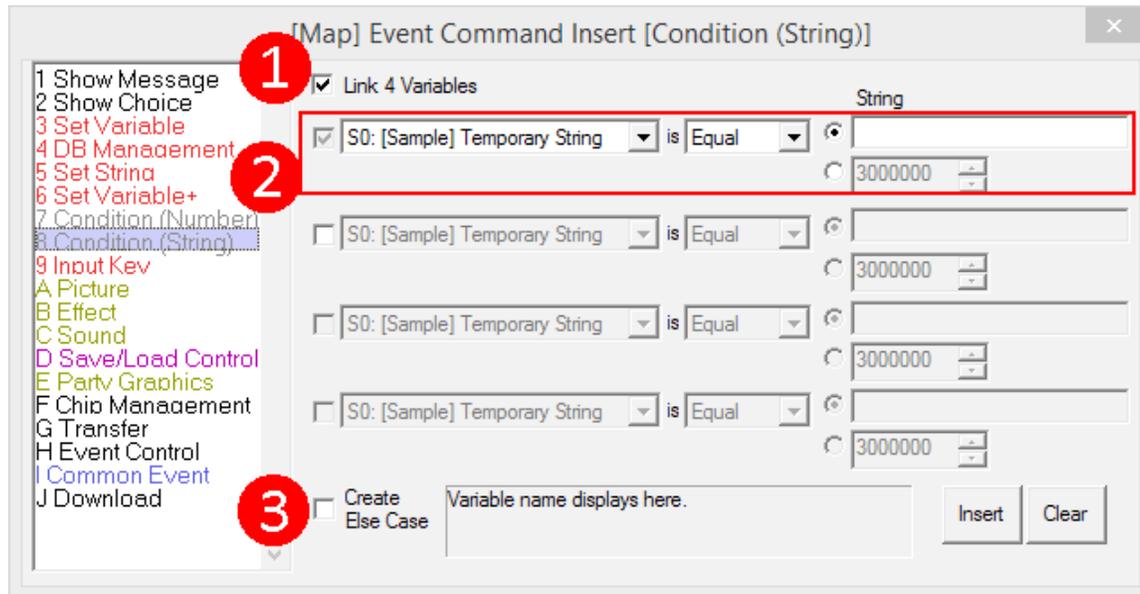
Includes a case for when none of the conditions are met.

6. "Clear" Button

Resets the contents.

Event Command "Condition (String)"

Splits event commands into a maximum of 4 cases based on the state of String Variables.



[General Explanation]

You can set up to four conditions for cases. Condition checks are done from top to bottom.

[Section Explanations]

1. Link 4 Variables

If this is checked, all four of the variables in the pull-down for each condition will be made the same as the top variable.

2. Define Conditions

Set how to compare the String Variable and the comparison string. You can input a comparison string manually, or select from String Variables.

There are four types of comparison.

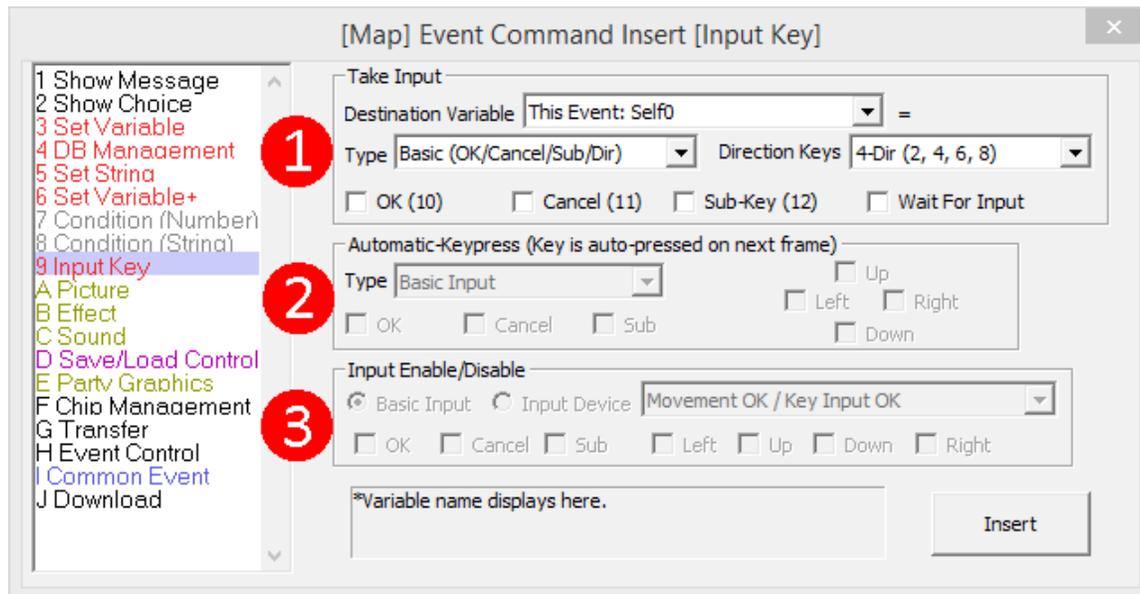
- **Equal** ... Take this case if the String Variable and the comparison string are a perfect (case-sensitive) match.
- **Not equal** ... Take this case if the String Variable and the comparison string are different in any way.
- **Includes** ... Take this case if the String Variable contains at least one instance of the comparison string.
- **Starts with** ... Take this case if the String Variable starts with the comparison string.

3. Create Else Case

Includes a case for when none of the conditions are met.

Event Command "Input Key"

Gets key/button input and such.



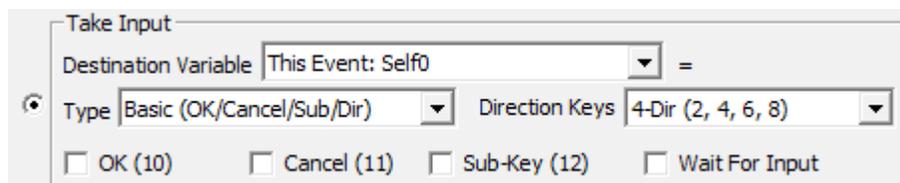
[Function Explanations]

1. [Take Input]

Checks if certain keys are being pressed at the moment this command runs, and sets the destination variable to the corresponding code. If nothing is found, it is set to 0. See explanations of each section below.

- **Type** ... Each type accepts different sets of keys.

- Basic (OK/Cancel/Sub/Dir)



Accepts the direction keys from the keyboard/gamepad, as well as Confirm (Enter/Space), Cancel (Esc/Backspace/Delete), and Sub-Key (Shift), using the key codes defined in System Variables 52 to 57 for Confirm/Cancel/Sub-Key.

- **Direction Keys** ... Set what directions are valid to accept. The values for each direction correspond to the numbers on the numpad like so.

7	8	9
4		6
1	2	3

- **OK/Cancel/Sub-Key** ... Set whether to accept each of the buttons.

- **Wait For Input** ... Pauses the event until any of the keys specified above are pressed.

- All Keyboard Keys (100~)

Take Input

Destination Variable: This Event: Self0 =

Type: All Keyboard Keys (100~) Direction Keys: 4-Dir (2, 4, 6, 8)

Specific Key Key Code: 100 Wait For Input

Accepts all symbol keys from the keyboard (basically, only the keys marked with a • in the image below). If you insert key codes into System Variables 52~54, you can use System Strings 1~3 to check the key names.

- **Specific Key** ... If you check this, you can accept only the key with the specified key code. You can get key codes not normally accepted by "All Keyboard Keys" (keys without a • in the image below) using this.

The mapping of key codes to the keyboard is as shown in the following chart.

* Both the main and numpad Enter keys are "128," but if you press both together and let go of the main Enter, it becomes "256."

The half/full-width key is "248," but after being pressed once, it and the kana key act as if they are always pressed.

PrtScr is "283," Home key is "299"

Esc 101	F1 • 159	F2 • 160	F3 • 161	F4 • 162	F5 • 163	F6 • 164	F7 • 165	F8 • 166	F9 • 167	F10 • 168	F11 • 187	F12 • 188	Pause 297	PrtSc 283	299 Del
半角/ 248	1 • 102	2 • 103	3 • 104	4 • 105	5 • 106	6 • 107	7 • 108	8 • 109	9 • 110	0 • 111	- • 112	^ 244	¥ 225	Back 114	
Tab 115	Q • 116	W • 117	E • 118	R • 119	T • 120	Y • 121	U • 122	I • 123	O • 124	P • 125	@ 245	[126	Enter 128		
CapsLock 158	A • 130	S • 131	D • 132	F • 133	G • 134	H • 135	J • 136	K • 137	L • 138	; 139	: 246	'] 127			
LeftShift 142	Z • 144	X • 145	C • 146	V • 147	B • 148	N • 149	M • 150	< • 151	> • 152	? • 153		↑ 143	8 8	Right 154	
Ctrl • 129	Win	Alt • 156	無変換 223	Space 157	変換 221	カタカナ 212	AltGr 284	Ctrl 257	← 4	↓ 2	→ 6				

Confirm: 10 Cancel: 11 Sub-Key: 12 8 Directions: ↖ 7 9 ↗
↙ 1 3 ↘

ScrL • 170	7 • 171	8 • 172	9 • 173	- • 174
Nur • 169	4 • 175	5 • 176	6 • 177	+ • 178
/ 281	1 • 179	2 • 180	3 • 181	Enter 256
* • 155	0 182	. 183		128 128

- Keys accepted by normal key input
- Keys to be careful about using
- Key whose value changes in certain conditions
- Key readable only when accepting specific key

o Testing and Image Thanks: Nanami

- Mouse Click/Position

The screenshot shows the 'Take Input' configuration panel. The 'Destination Variable' is set to 'This Event: Self0'. The 'Type' is 'Mouse Click/Position' and the 'Target' is 'Click State'. There are four checkboxes: 'L. Click (20)', 'R. Click (21)', 'M. Click (22)', and 'Wait For Input', all of which are currently unchecked.

Gets mouse clicks or mouse position.

- **Click State:** Gets left/right/middle clicks.
- **Mouse X Pos:** Gets the mouse's X position in the window.
- **Mouse Y Pos:** Gets the mouse's Y position in the window.
- **Wheel Delta:** Gets the amount of mouse wheel rotation in 1 frame.
Values are generally -1 or -2, 0, and +1 or +2.

However, on some PCs, there are cases of constantly getting values below -100 or above +100.

- Pad Buttons (300~)

The screenshot shows the 'Take Input' configuration panel for 'Pad Buttons (300~)'. The 'Destination Variable' is 'This Event: Self0' and the 'Target' is 'Click State'. There is a 'Specific Key' checkbox which is unchecked, and a 'Key Code' field set to '100'. The 'Wait For Input' checkbox is also unchecked.

Accepts all buttons from gamepads. If you insert key codes into System Variables 55~57, you can use System Strings 4~6 to check the button names.

2. [Automatic-Keypress]

Automatically causes a specific key or mouse input to happen on the next frame.

Uses for this function include making a replay function, adding a gameplay demonstration, or letting you use alternate mouse controls for an RPG (* though input will be 1 frame = about 0.017 seconds (depends on FPS) late).

If you choose Keyboard, the below-mentioned "disabled" keys will still be able to take input, so be careful.

The idea is to limit player input and have input be entirely handled by WOLF.

By the way, if "Basic Input" is disabled, "Basic Input"s made by Automatic-Keypress will not work either.

- **Type ...** Each type can press different sets of keys.

- Basic Input

The screenshot shows the 'Automatic-Keypress (Key is auto-pressed on next frame)' configuration panel. The 'Type' is 'Basic Input'. There are checkboxes for 'Up', 'Left', 'Right', 'Down', 'OK', 'Cancel', and 'Sub', all of which are currently unchecked.

The same as Take Input "Basic (OK/Cancel/Sub/Dir)", but outputting it instead.

- Keyboard

Automatic-Keypress (Key is auto-pressed on next frame)

Type

Input Key Code

The same as Take Input "All Keyboard Keys (100~)", but outputting it instead.

- Mouse Left/Right/Middle Click / Position

Automatic-Keypress (Key is auto-pressed on next frame)

Type

Wheel X

L. Click R. Click M. Click Position Y

Lets you automatically cause left/right/middle mouse clicks.
You can also move the mouse to a specific position.

3. [Input Enable/Disable]

You can toggle whether various inputs (Confirm, Cancel, Sub-Key, Up/Down/Left/Right, as well as keyboard/mouse/gamepad input) are enabled or disabled.

- Basic Input

Input Enable/Disable

Basic Input Input Device

OK Cancel Sub Left Up Down Right

You can set the state of the selected keys to "Movement OK / Key Input OK (default)," "Movement NO / Key Input OK," or "Movement NO / Key Input NO."

For instance, if you want to prevent the hero from moving with the arrow keys, select "Up," "Down," "Left," "Right," and "Movement NO / Key Input OK."

- Input Device

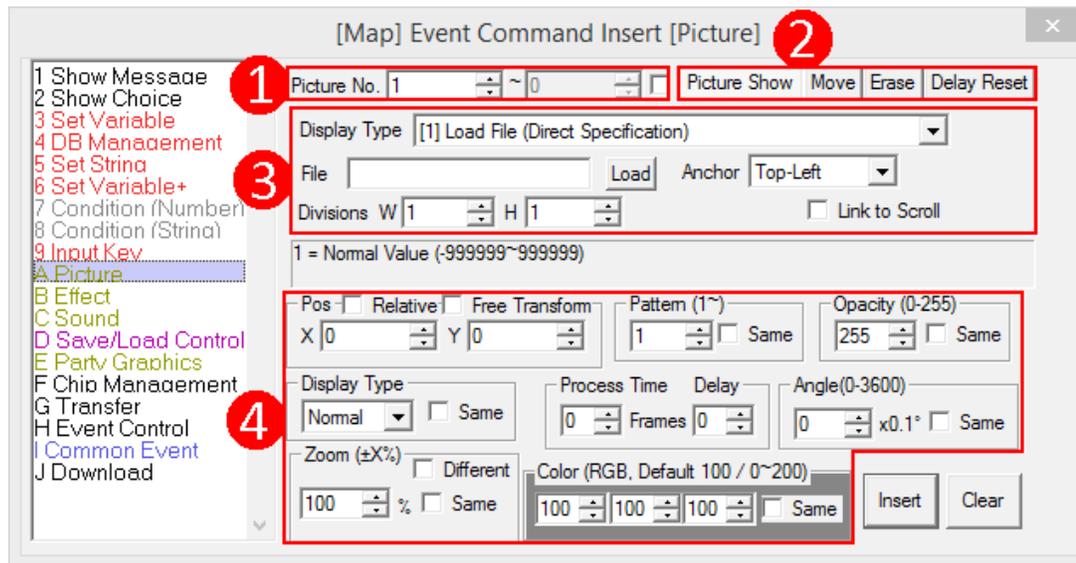
Input Enable/Disable

Basic Input Input Device

You can enable and disable all or individual keyboard keys, all mouse input, all gamepad input, or all input devices.

Event Command "Picture"

Lets you load and display images, or display strings as images.



[General Explanations]

- All number fields can have either normal integers or variables (example: 2000000 = Normal Variable #0) put into them.
- "Angle" does not function on "String Pictures" and "Simple Windows." (it is always 0).
- If "Angle" is anything but 0, checking "Different" for Zoom and setting different Width and Height zooms will not function properly.
- If you check the "Same" box next to a field, that field will automatically use "the previously specified value."
- If you never erase pictures you are not using, it adds to the drawing load and slows the game down. Holding F8 during a Test Play shows a list of currently-used picture numbers, so refer to that to cut down on unnecessary picture-displaying.
- On the other hand, image files that are not being displayed in any picture slots will be loaded from disk upon display, which causes a bit of lag on that frame. If you have pictures you use frequently, keeping it always displayed at opacity 0 in some picture slot will allow other picture slots to quickly display that image file.

[Section Explanations]

1. Picture Number

Choose which picture slot to act upon. You can also use variables here. A single picture slot holds only a single picture, so if you display another image with an already-used picture slot, the previously-displayed picture is automatically erased.

Checking the box next to the number field lets you manipulate multiple sequential picture slots at once. Higher-number picture slots are displayed above lower-number ones.

- Picture slots **lower than 100,000 display under messages and choices.**
- Picture slots **100,000 or greater display above messages and choices.**
- Picture slots **-1 to -99,999 display above the map and below events** (some chips, like those with the ★ property, will display above the picture).
- Picture slots **-100,000 or lower display below the map and above the background.**

* Picture numbers -1 or less will be affected by Effect color changes. (Added 2/27/2017)

2. Picture Show / Move / Erase / Delay Reset

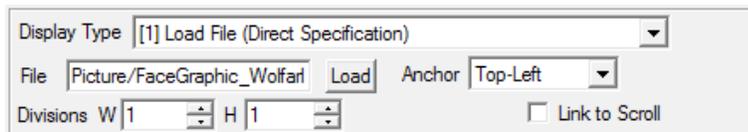
The action being done for the picture. The fields you can use vary depending on this action.

- **Show** ... Shows a new picture. If the specified picture was already being displayed, it behaves the same as "Move." If Process Time is 1 or more, it starts from a transparent state.
- **Move** ... Moves the picture. Does nothing if the specified picture is not being displayed.
- **Erase** ... Erases the picture, and erases the image from memory.
- **Delay Reset** ... Resets all future processes with an "Activation Delay" so that they do not happen.

3. Display Type ("Picture Show" Only)

There are four display types, with different ways of setting them.

[1] Load File (Direct Specification)



Select the image file to display with the "Load" button.

- **Divisions** ... When loading a single image split up into sections, specify the number of horizontal and vertical sections here. You can choose which individual section to use by specifying its Pattern number. For instance, if the image is divided up 4 wide and 2 high, it uses the following Pattern numbers.

1	2	3	4
5	6	7	8

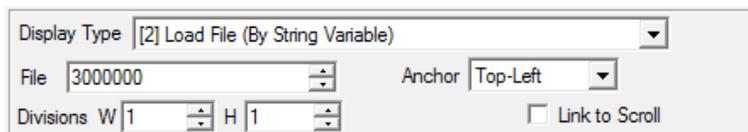
Divided 4 Wide, 2 High Case

- **Anchor** ... Set which part of the image to use as the origin point. You can select "Top-Left," "Bottom-Left," "Center," "Top-Right," or "Bottom-Right."
- **Link to Scroll** ... Check this, and the picture will stay in a fixed position on the map.

* "Link to Scroll" Warning (Added 12/30/2016)

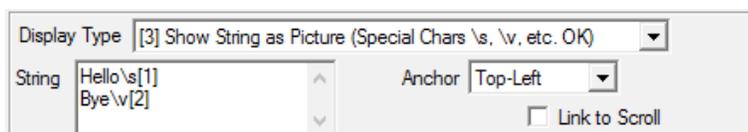
Since "linked to scroll" pictures move according to "the map's scroll amount at 100% zoom," **using Effect commands to enlarge the map can result in misaligned pictures.** If you want to match a picture's scroll link to the enlarged map, use Effect to zoom the picture by the same amount as the map.

[2] Load File (By String Variable)



Specify the filename by String Variable. For "File," input the number for the String Variable that contains the filename. Otherwise, it is the same as **[1] Load File (Direct Specification)**.

[3] Show String as Picture



Input the text to display in the "String" box. You can also use special codes like `\s[?]` and `\v[?]` (? = a number).

For a display example, see below.

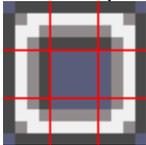


[4] Simple Window

Display Type [4] Simple Window (3x3 Divisible Base Graphic + Size) ▾
File SystemGraphic/WindowBase Load Anchor Top-Left ▾
Size W 120 H 100 Link to Scroll

For "File," choose a base image divisible into 3x3 sections, then input the target size, and a **window graphic of that size will be automatically created.**

An example base image is shown below. (The red lines are dividers for explanation purposes only.)



↑ This is an enlarged 12x12-pixel window base image. → *  Actual size!

As long as the image is constructed such that, when it is split up 3x3, it contains the 4 corners and the parts in-between, you can easily make a window with it. Here is an example of Simple Windows displayed by putting this into practice.



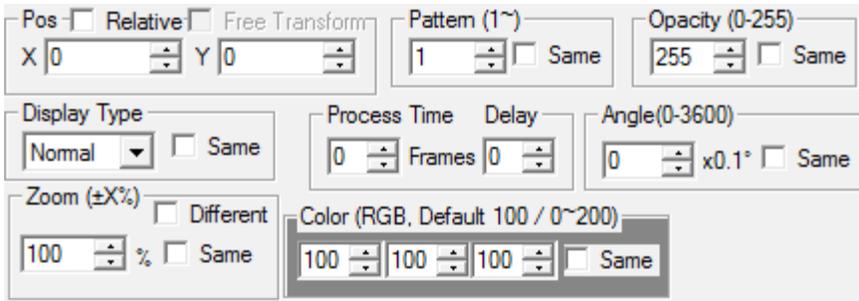
* Note: Small windows less than 2/3 the size of the base image will not display properly.

[5] Simple Window (File by String Variable)

Display Type [5] Simple Window (File by String Variable) ▾
File 3000000 Anchor Top-Left ▾
Size W 120 H 100 Link to Scroll

Specify the filename by String Variable. For "File," input the number for the String Variable that contains the filename. Otherwise, it is the same as **[4] Simple Window.**

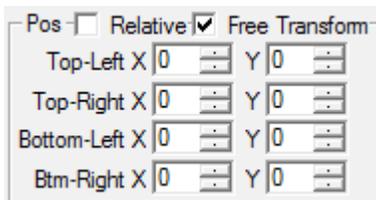
4. Picture Position/Display Settings



- Position

The position to display the picture at/move it to, in pixels.

- **“Relative”** ... Check this to use the current display position as a base onto which X and Y are added.



- **“Free Transform”** ... Check this to set individual top-left/top-right/bottom-left/bottom-right coordinates to transform the shape freely. If used properly, you can implement pseudo-3D effects.

* Each picture that uses Free Transform adds slightly to processing load.

* **“Zoom” and “Angle” use the values specified in normal position mode.**

* If you Show/Move a picture with this, then Move it without checking Free Transform, it will automatically be treated as a “Center”-anchored picture. Zoom and Angle will also be calculated for a center origin.

* Pictures displayed as normal rectangles (or squares) can be turned into free pictures.

* When setting the four coordinates with Relative, the values are added to the coordinates of “their respective points.”

- Pattern

When dividing up an image into sections, specify which section to use.

Applies only when using “[1] or [2] Load File.” Otherwise, it has no function.

- Opacity

Set the picture's opacity.

At 255, the image will be displayed as-is, and at 0, it will be invisible. 128 makes it half-translucent.

- Display Type

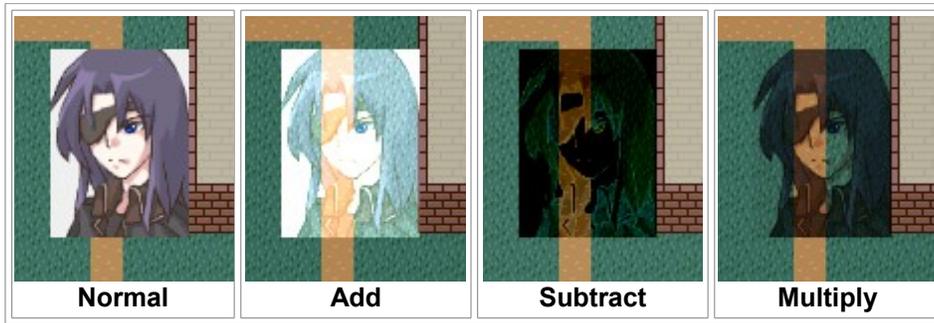
You can display the image in four different ways.

- **Normal** ... Draw on the screen normally.

- **Add** ... Draw by adding to the RGB values on the screen. In simple terms, only the bright parts of the image will brighten those parts on the screen.

- **Subtract** ... Draw by subtracting from the RGB values on the screen. In simple terms, only the bright parts of the image will darken those parts on the screen.

- **Multiply** ... Draw by multiplying the RGB values on the screen with this image. RGB 255 is considered 1 for the purposes of this multiplication, so generally, dark parts of the image will darken the screen, and white parts (255, 255, 255) will have no effect. **Also, for this display type, Opacity is ignored (always 255).**



- Process Time / Delay

"Process Time" is how long (in frames) it will take to Show, Move, or Erase the picture. The higher this value, the more time the action will take.

"Delay" is the time to wait (in frames) before beginning to Show, Move, or Erase. The higher this value is, the later the specified process will start. You can even give multiple actions varying Delays to make them start at the same time.

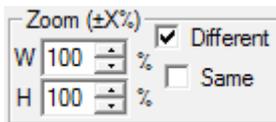
- Angle

Set the angle of the image in 0.1-degree units (in other words, 3600 is a full rotation).

Note that if Angle is anything but 0, the "Different" option for Zoom will not function. Also, values over 3600 will still result in the proper rotation.

- Zoom

Set the magnification of the picture.



- "Different" ... Check this to set separate width and height magnification.

-- **Width** ... Magnifies the image horizontally. -100% will horizontally flip the image.

-- **Height** ... Magnifies the image vertically. -100% will vertically flip the image.

- Color

Set color adjustment of the image with RGB (red, green, blue) values. 100 is normal. The values affect the strength of each color; 200 is stronger, and 0 removes that color.

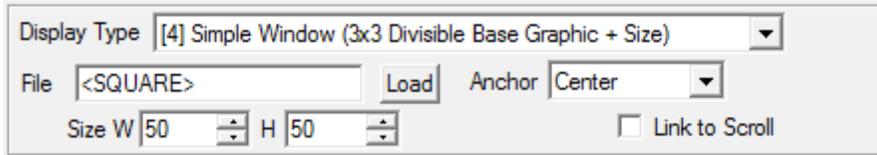
The values represent the strength ratios of RGB light, so if you set 200,200,200 for a totally black image (RGB=0,0,0), the color will not change.



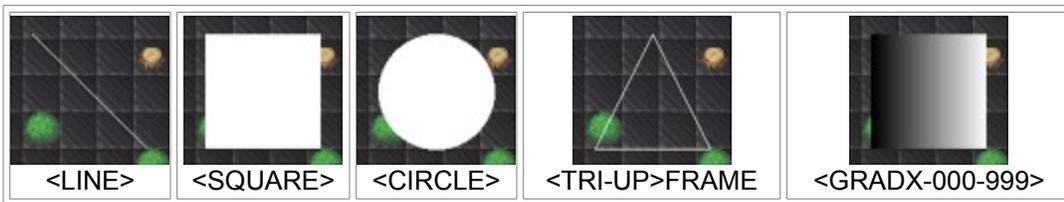
The background portion of the input area will change to the specified color.

[Secret Function: Shape Display]

With Simple Window, if you specify special strings in "File," you can display certain shapes. However, using this shape display will render the value of "Angle" useless (always 0).

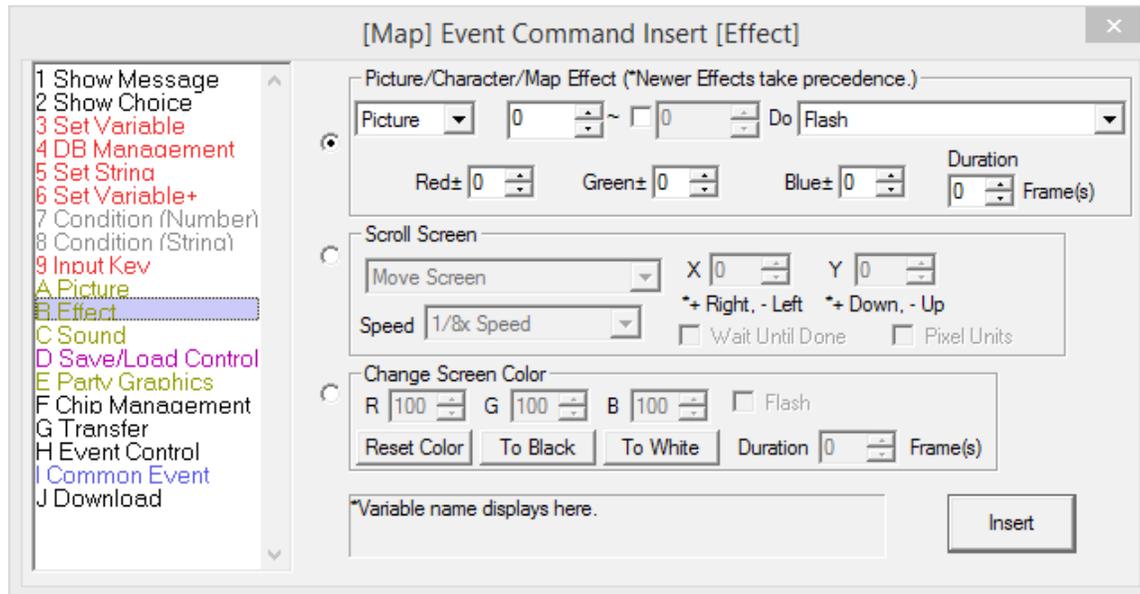


- **<LINE>**: Draws a white line from "Position" to "Position + Size." In this particular case, negative values will work for "Size."
- **<SQUARE>**: Draws a white square of the specified "Size." If you add "FRAME", it will only draw the outline (example: <SQUARE>FRAME).
- **<CIRCLE>**: Draws a white circle according to "Size." If you add "FRAME", it will only draw the outline (example: <CIRCLE>FRAME).
- **<TRI-XXXX>**: Draws a white triangle according to "Size." In place of XXXX, put "UP", "DOWN", "LEFT", or "RIGHT" to define the direction. If you add "FRAME", it will only draw the outline (example: <TRI-LEFT>FRAME).
- **<GRADX-111-999>**: Draws a rectangle of the specified "Size" with a horizontal gradient. The "111" part is the left-side color, and the "999" part is the right-side color; use the extents of RGB (red/green/blue) you want (example: 000 for black, 999 for white, 090 for green).
- **<GRADY-111-999>**: Draws a rectangle of the specified "Size" with a vertical gradient. The "111" part is the top-side color, and the "999" part is the bottom-side color; use the extents of RGB (red/green/blue) you want.



Event Command "Effect"

Applies effects to characters, pictures, or the map, and allows screen-scrolling or color changes.



[Function Explanations]

- Picture/Character/Map Effect

You can perform the following actions on characters, pictures, or the map. Be warned that changes made to values by this command are not considered when getting those values with "Set Variable+."

- **Effects on Pictures** (Specify picture number(s) with "□ ~ □" boxes)

* **Effects done on pictures are processed independently of what is specified in the "Picture" command.**

+ **Flash ...** Flashes by adding to the picture's **Red/Green/Blue** "Color" values. The range of values is ± 200 .

+ **Shake ...** Shakes the picture **a specific number of times, with extent of movement given by Shake X and Shake Y**. The shorter the "1x Duration" is, the faster the shaking.

+ **Draw Position Shift ...** Shifts the draw position of the picture by **Shift X** and **Shift Y**.

+ **Color Correct ...** Adds the specified **Red/Green/Blue** values to the picture's "Color." The range of values is ± 200 .

+ **Zoom ...** Zooms in the picture by **the given zoom amount, centered on Center X and Center Y**.

+ **Switch A (Flicker) ...** Repeatedly flicker using color change by given RGB amounts (max ± 200) on an interval. It transitions between pre-color-change and post-color-change. Setting all RGB values to 0 or setting frame interval to 0 will stop the flashing.

+ **Switch B (Auto-Flash) ...** Repeatedly flash using color change by given RGB amounts (max ± 200) on an interval. It changes without waiting, then returns to normal. Setting all RGB values to 0 or setting frame interval to 0 will stop the flashing.

+ **Auto-Enlarge [%] ...** Changes the picture's magnification by the given change in **horizontal/vertical zoom on an interval**. After each enlarge/shrink, it rests for the given number of frames. It may help to imagine it as "pulsing."

+ **Auto-Pattern-Switch (One) ...** Switches through pattern numbers one at a time on an interval, **from the start pattern to the end pattern**. Once one pattern loop is complete, the picture reverts to the originally-defined pattern number.

+ **Auto-Pattern-Switch (Loop) ...** Switches through pattern numbers one at a time on an interval, **from the start pattern to the end pattern**. Upon reaching the end pattern, it loops to the start pattern again.

+ **Auto-Pattern-Switch (Round Trip) ...** Switches through pattern numbers one at a time on an interval, **from the start pattern to the end pattern**. Upon reaching the end pattern, it goes one at a time back to the start pattern, then loops.

* Effect changes on pictures are processed independently of pictures themselves.

For instance, if you do a "Draw Position Shift," it continues to draw at "X/Y specified by Picture + position Shift X/Y." This takes effect until position Shift X/Y are returned to 0, or the picture is erased.

* "Flash" and "Color Correct" cannot exceed the 0~200 color range in Picture to further change the color.

- Effects on Characters

+ Flash ... Flashes by adding the specified **Red/Green/Blue** values. The range of values is ± 100 . Negative values make it darker. -100,-100,-100 makes it completely black.

+ Shake ... Shakes the character a **specific number of times**, with extent of **movement given by Shake X and Shake Y**. The shorter the "**1x Duration**" is, the faster the shaking.

+ Switch A (Flicker) ... Repeatedly flicker using color change by given RGB amounts on an interval. It transitions between pre-color-change and post-color-change. Setting all RGB values to 0 or setting frame interval to 0 will stop the flashing.

+ Switch B (Auto-Flash) ... Repeatedly flash using color change by given RGB amounts on an interval. It changes without waiting, then returns to normal. Setting all RGB values to 0 or setting frame interval to 0 will stop the flashing.

- Effects on Map

+ Zoom ... Zooms in the map, events, player character, ally characters, background, and fog by the given zoom amount, **centered on Center X and Center Y**. To revert to former state, set the Zoom to 100%.

* Please use **Zoom 100% or greater**. You can use less than 100%, but anything outside what is normally drawn at 100% will be black, and chips one tile beyond a loop will be drawn. To maintain processing speed, nothing is done to counteract this, so be warned. You can try to cover for this phenomenon yourself if you wish.

* Also, with the Zoom greater than 100%, the hero may move out of view when they are near the edge of the screen, but this is intentional behavior for map zoom. If you are allowing play while zoomed in, you will need to do something like make the edges of the map impassable.

+ Shake ... Shakes the screen. You can select a **Vertical Shake** or **Horizontal Shake**, or you can **Stop Shake**. **Power** and **Speed** can be configured; Power represents the size of one shake, and Speed represents the speed at which one shake occurs. The higher both of them are, the more intense the shaking.

- Scroll Screen

Scrolls the screen (shifts the center). There are four options.

- **Move Screen** ... Scroll the screen by the number of chips given by X and Y. Variables can be used in the X and Y boxes (example: input 2000000 to refer to Normal Variable #0).

- **Back to Hero** ... Revert scroll to centering on the hero's position.

- **Lock Scroll** ... Do not let the screen scroll after this.

- **Unlock Scroll** ... Cancels scroll lock. However, it does not automatically center on the hero when unlocked, so generally you will need to use "**Back to Hero**" after this. Also, if "**Wait Until Done**" is checked, event processing will wait until the scroll completes.

- Change Screen Color

Changes the RGB (Red/Green/Blue) values of the screen. 0 is the lowest value, 200 is the highest, and 100 is normal. Checking "**Flash**" makes the screen light up with that color for the given duration.

- **Reset Color** ... Resets all colors to 100.

- **To Black** ... Makes colors 0,0,0. Using Change Color or Flash makes it black.

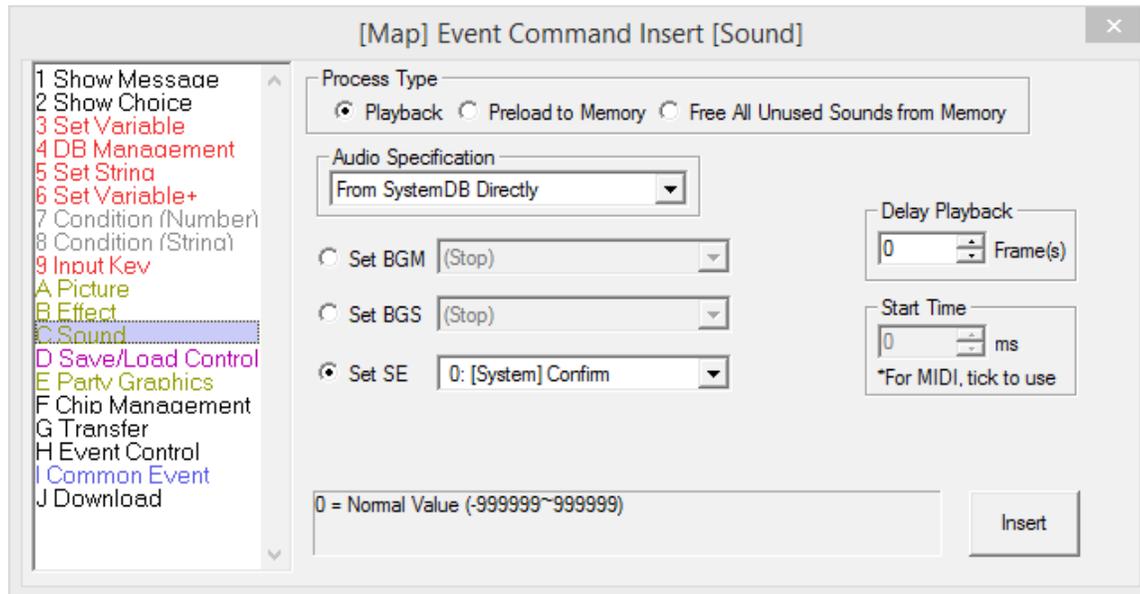
- **To White** ... Makes colors 200,200,200. Using Change Color makes it white-ish, and Flash makes it completely white.

* Since negative picture numbers are "above the map, below events," they are affected by this color change.

Pictures with a picture number 0 or more are not affected by screen color changes. (2/27/2017)

Event Command "Sound"

Plays music and sound effects.



[Section Explanations]

- Process Type

- **Playback** ... Plays BGM or SE. Generally, this is all you need.
- **Preload to Memory** ... Loads the data for the BGM or SE into memory; it does not play it.
- **Free All Unused Sounds from Memory** ... Frees all sounds not currently being played from memory.

~~—Manually Free Memory (SE Only) ... Frees the data for the specified SE in memory.~~

→ Due to not working properly, this has been removed as of Version 2.20.

- Audio Specification

Set the method of referring to the sound.

You can only load data saved in the System Database using the Directly and by Variable options.

- **By Filename** ... Specify a file path to load. You can also use variables like %s or %cdb.
- **From SystemDB Directly** ... Select a sound to play from the entries in the System Database.
- **From SystemDB by Variable** ... Specify the database number of the sound to play with a variable.

- Set BGM / Set BGS / Set SE

Each of these three types reads from System Database Types 1~3.

To make these update to the latest info, change the **Audio Specification**.

If using By Filename, specify the file path here.

- Fade Time (BGM/BGS Only)

For BGM and BGS, you can specify the fade time of music in frames. This is used to make music gradually fade in, or gradually fade out.

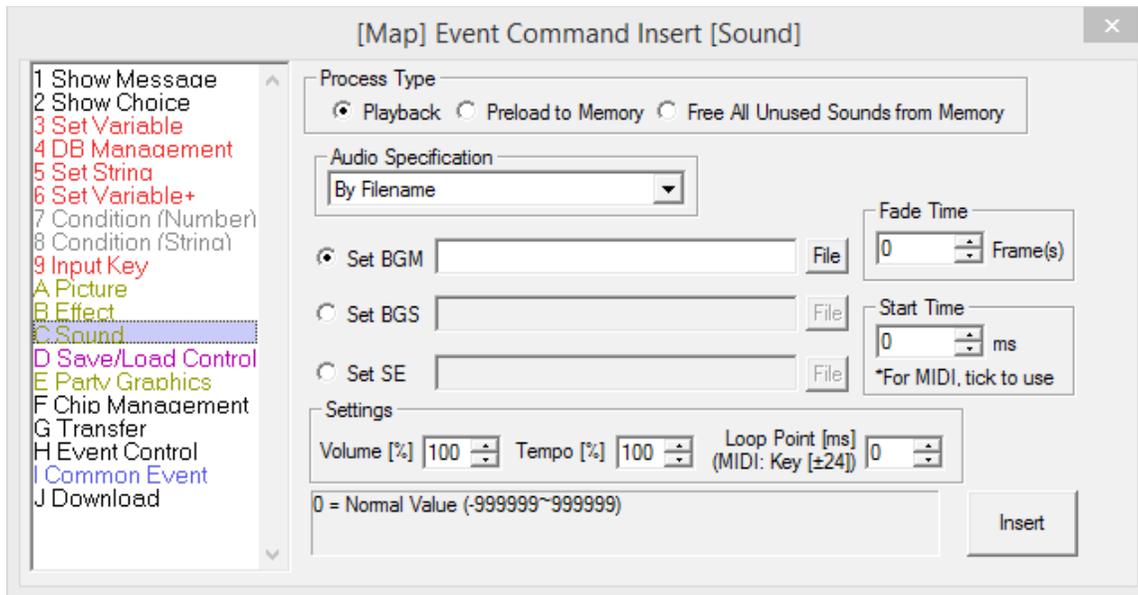
However, this will not function if MIDI playback is set to Hardware in [Basic Game Settings](#).

- Delay Playback (SE Only)

Only appears when playing SE. Delays the time at which playback starts by the specified number of frames.

- Start Time

You can make the song start playing from a specified position (units are milliseconds for MP3/OGG/WAV, ticks for MIDI). * Cannot be used for SE specification.



- Volume [%]/Tempo [%]/Loop Point [ms, BGM/BGS Only]

Only usable with By Filename. They function just like the ones in System Database Type 2 ([Type Explanations](#)).

* With BGM or BGS, if you load the same file that is already playing, the song will continue but change to the new volume and tempo settings.

[Hidden Function] You can refer to a String Variable in the By Filename field (ex: \s[1] or \cself[6]).

If you enter a multi-line string structured like this into that variable:

"Filename<Linebreak>Volume (%)<Linebreak>Tempo (%)<Linebreak>Loop Position (ms, BGM/BGS Only)"

It will ignore the values from the editor side, and instead use the numbers you provide for the volume/tempo/loop settings.

Example: SoundFolder\Test.wav

100

120

Passing this string will play Test.wav at 100% volume, 120% tempo.

Replace "SoundFolder" with the name of the folder currently holding the sound file you wish to use.

The folder needs to be within the Data folder of the WOLF RPG Editor, otherwise you will get errors.

[Hidden Function] Provided the loop start is set to "0,"

the LOOPSTART tag in OGGs and MP3 files (* must be in the first 1KB of the file) will be read, and that time is automatically set as the loop start position.

* LOOPLength will also be read. This specifies the length of the loop.

* Even in the case of MP3s, if there is a string like "LOOPSTART=123456" included anywhere in text such as the composer info, that will work.

↓ There are tools for adding the aforementioned loop settings to sound files.

[FOLE(FICUSEL Ogg Loop Editor)]

<http://forest.watch.impress.co.jp/docs/review/693911.html> (Japanese)

- Key-Change Function [MIDI Only]

For MIDI playback only, you can increase or decrease the key instead of Loop Start.

Increasing the key makes the notes higher, decreasing makes them lower. It is suitable for tuning songs to have an eerie atmosphere.

[Playing Multiple SEs]

When the same SE is played repeatedly, the previous SE is stopped once the new one begins to play. In short, you cannot play more than one instance of the same sound effect at the same time.

If you want to play a sound repeatedly without it being stopped, follow this procedure.

- [Example] **I want to play explosion sound "AAA.wav" repeatedly.** (* Written for those knowledgeable)

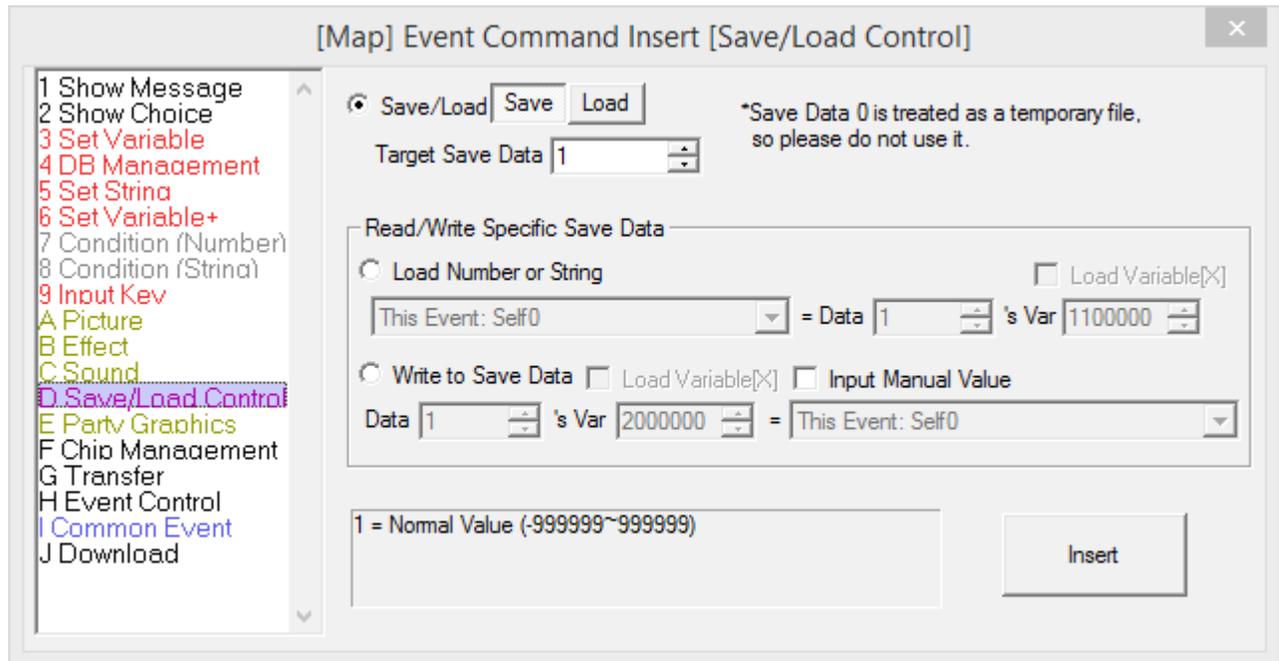
In the System Database, set SE Data 4~6 all to the same sound, "AAA.wav." Creating event commands that play sounds 4~6 in order will let you play three instances of "AAA.wav" at once.

For example: Create a Common Event "Explosion Sound." Every time it is run, play the SE "4 + CSelf10", then add +1 to CSelf10, then $CSelf10 \% 3$ (get the remainder of dividing by 3). This process should play the sounds in order of 4→5→6→4→5→6 every time it is called.

This will let you play up to 3 instances of the sound at once without much interruption. If you want to play it even more, you can use Data 4~10, etc.

Event Command "Save/Load Control"

Does processes involving saving and loading of data. You can even manipulate individual parts of save data.



[General Explanations]

- If you save or load inside a parallel event, it may be forced to end, so please do not use this in parallel events.
- "Load Number or String" cannot read from event Self Variables.

[Function Explanations]

[Save/Load]

Choose to write to save data or load from save data.

Using save slot 1 as an example, save data files are named:

"Save\SaveData01.sav"

You can specify variables (example: 2000000 [Normal Variable #0]) for the save data number.

If no save data exists to load, processing simply moves to the next event command.

If save data does exist, the event is instantly terminated, and the game resumes from the saved state.

[Load Number or String]

Loads a number/string from the specified save data, and puts it into a variable in the game currently being played. If the target save data does not exist, 0 is assigned for numbers, and a blank string for strings.

Incidentally, System Variable #24 (9000024: [ReadOnly] Load Check (1=Pass 0=Fail)) is a variable that always returns 1 if the save data exists. If you read that System Variable using this command, you can use it to check if the save data exists.

[Write to Save Data]

Writes only the specified number/string to save data.

* If no save file exists, a new one is created.

But in this case, no Variable DB type info or data is written, so be careful.

[Usage Example]

- The Save/Load functions are essential for games that require saving.
- "Load Number or String" and "Write to Save Data" are used only if you need to store info that cannot be stored in the individual save slots in a separate file. For example, it can be used to make a file that stores "the total number of times the game has been beaten."

[Special Functions]

If you specify a String Variable (3000000~) for the save data number, that string's data name is used as the save data filename. For example, if you put "SystemInfo.sav" in String Variable #0, input 3000000 for the save data number, and pick Save, it will save a file named "SystemInfo.sav" in the **same folder as Game.exe**.

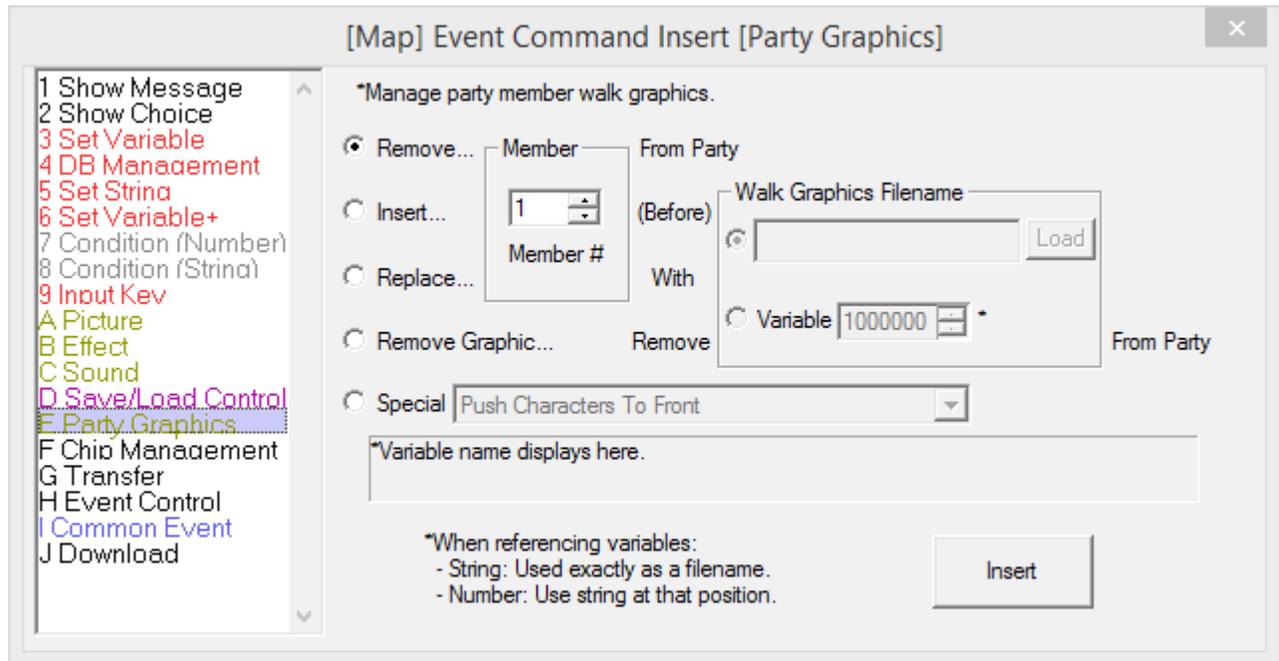
[Important Note]

If you save in the middle of an event, loading the save data will resume you in the state of "the event is not running." In other words, **loading will not resume any event commands that were in progress**, so be cautious.

If you make a mistake with the positioning of Set Variable, this can cause problems like "after loading a save made during a boss battle event, it is treated like the event ended and nothing else happens."

Event Command "Party Graphics"

Changes the graphics of the player characters. You can also form a party lineup.



***Party**... You can have the player characters walk around in a line. Here, this lineup is referred to as the "party." A party can be up to 6 people, including the lead character.

[Function Explanations]

- Remove

Removes member [#] from the party.

- Insert

Inserts walk graphics before member [#]. You can directly pick the walk graphic filename with the Load button, or load it from a String Variable. If you put a number in the Variable box, and it is a Normal Variable, it will go to read the String Variable at the position stored within. If it is a String Variable, it reads the file with that name.

- Replace

Changes member [#]'s character graphic to another graphic. You can also replace graphics for characters that currently have no graphic.

- Remove Graphic

Removes all characters with the specified graphic from the party.

- **Special**: You can choose one of the following actions.

- **Push Characters To Front** ... If there are blank characters, that empty space is removed.
- **Erase All Characters** ... Removes all character graphics from the whole party.
- **Warp Party to Hero** ... Warps all allies to the player character's position.
- **Start Hero/Party Synchro** ... Starts synchronized movement, which preserves the current relative position of allies. As an example, this would let you move the party while they stand side by side.

- **Cancel Hero/Party Synchro** ... Ends synchronization.
- **Make Party Transparent** ... Makes all members invisible. Often used during scenes.
- **Cancel Party Transparency** ... Clears invisible state for all members.
- **Save Party Members** ... Remembers all current party graphics.
- **Load Party Members** ... Returns the memorized party graphics.
- **Turn Party Following On (Default)** ... Makes allies follow behind the hero. Even if you use Move Route on allies, when the hero moves, they will move accordingly.
- **Turn Party Following Off (Stop)** ... Makes allies stop following the hero and stay the way they are. They will not be influenced by the hero's movement, so this is good for when you use Move Route on them.

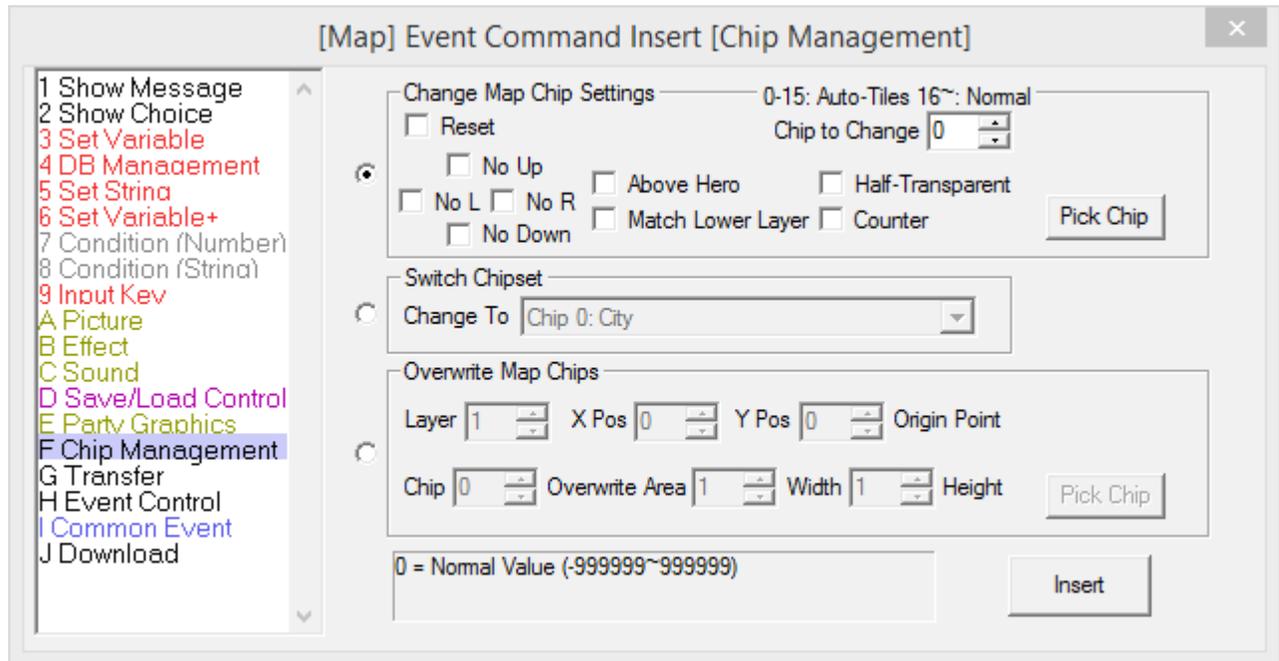
[Usage Example]



You can make a lineup like this.

Event Command "Chip Management"

Can temporarily change the properties of chips, or the current chipset, or make edits to the map itself.



[Function Explanations]

- Change Map Chip Settings

Temporarily change the passability and properties of map chips. For "Chip to Change," 0~15 are auto-tiles (the first and second row of tiles in the Map Chip Window), and 16 onward correspond to the chips in rows 3 onward.

When executed, the 8 properties will be changed. (For instance, if you uncheck everything, it becomes a chip that can be passed through any direction, with no special properties like "tall grass.") These changes persist until the map changes. The main application is for making something like a boat that can only move on water.

- **"Reset"** Option ... Reverts all chip passability settings to normal.

- Switch Chipset

Changes the chipset of the current map. You can also specify with a variable.

For example, you might switch to a night or day chipset, allowing you to get more out of fewer maps.

* Changed chipsets revert to normal when the player moves to another map.

- Overwrite Map Chips

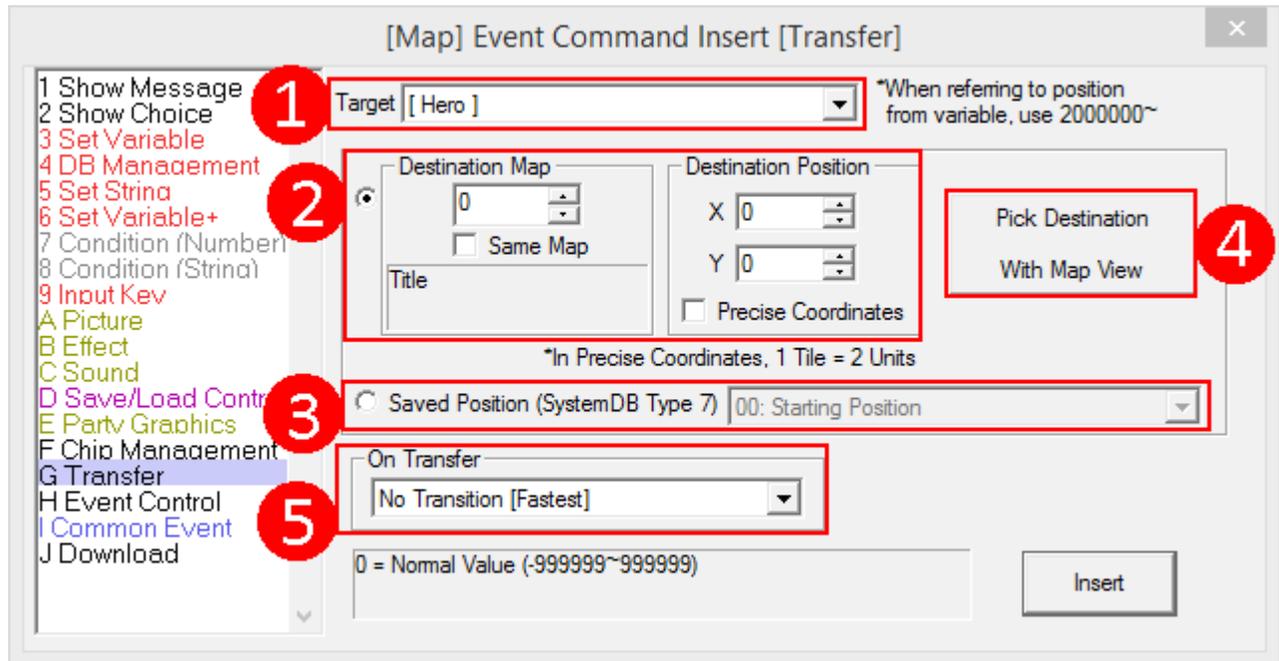
Rewrites chips at a specified position on the current map. Set the target Layer, the X and Y Position of the origin point, the Chip number, and the Width and Height of the area to overwrite.

This has various uses, such as making auto-generated dungeons like Mystery Dungeon, changing tiles to add foot-prints when you walk on snow, making a dungeon-creating RPG, or making a room form when you go to check a wall.

* Overwritten map chips revert to normal when the player moves to another map.

Event Command "Transfer"

Changes the position of the player character (hero) or events.



[Section Explanations]

1. Target

Select the object to change the position of. You can also specify by variable.

If the variable contains -1, the target is this event (or the caller event, if this is a Common Event).

2. Input Destination Manually

Manually input the destination.

Map number and coordinates can use variables (example: 2000000 = Normal Variable #0).

3. Select Saved Position As Destination

Select a destination from the positions saved using the "[save position](#)" function.

* Older versions did not have the option to pick while looking at the map, so the "save position" function was included as a substitute. As such, there is now less reason to use it.

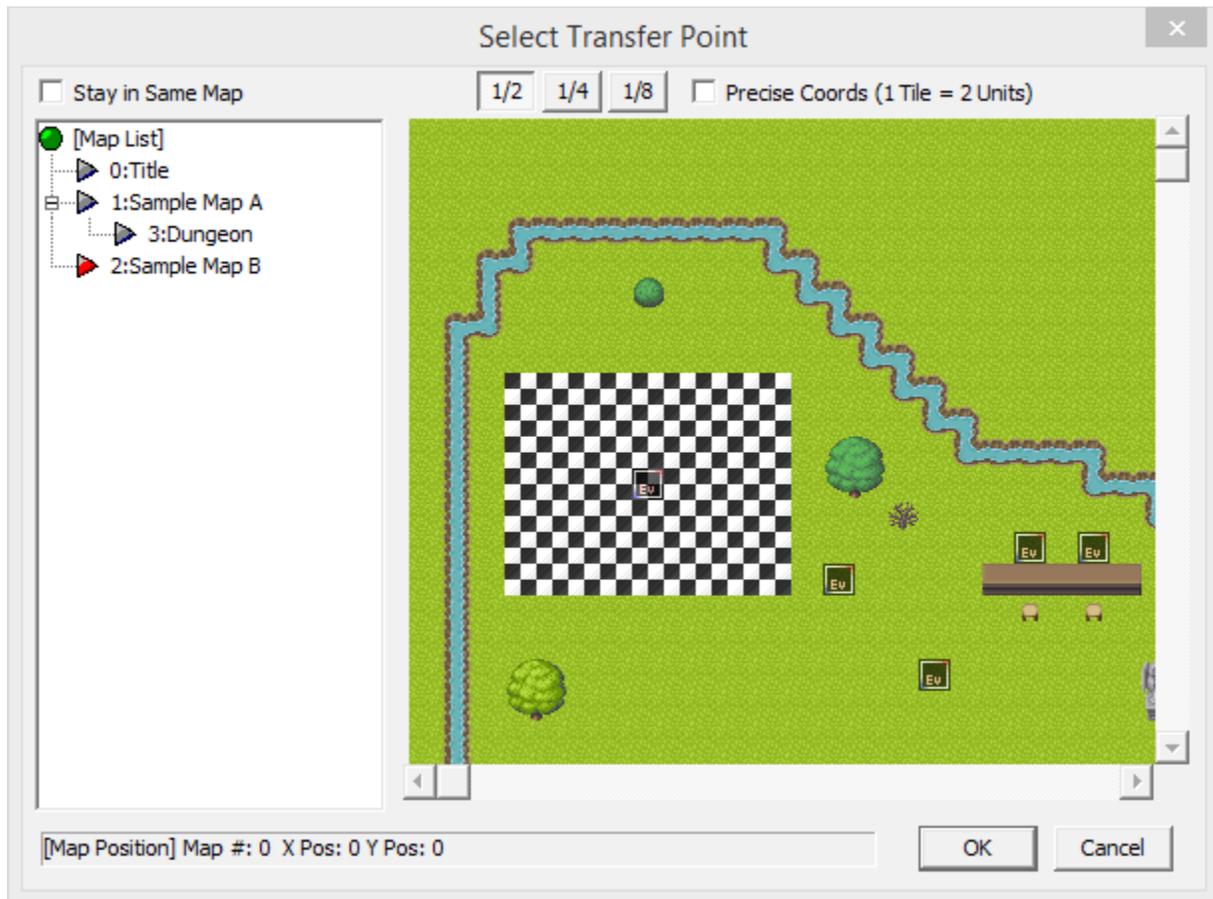
4. Pick Destination With Map View

Select a destination while looking at the map. Clicking this will display the following window.

[Select Transfer Point Window]

Select the map and position to transfer to with the mouse, and click OK.

This lets you easily set the map number, X and Y, and other options.



5. "On Transfer" Options

No Transition [Fastest] ... Performs transfer instantly upon execution.

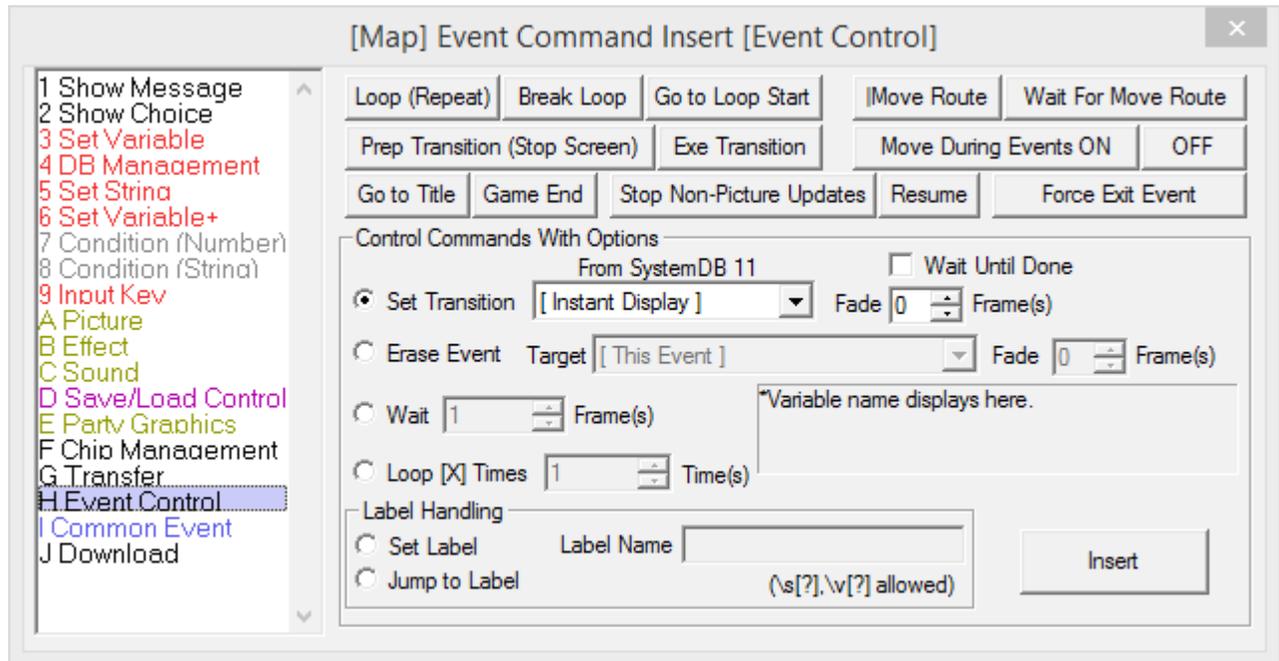
Do Transition [No Fade, Fast] ... Performs transition on transfer.

Do Transition [Fade, Slow] ... Fades out and performs transition on transfer.

* The transition graphic used is the one set by "Set Transition" under "[Event Control](#)."
(If there is none, it is just a regular fade.)

Event Command "Event Control"

Performs various other processes.



[Function Explanations]

[Loop (Repeat)]

Commands inside a loop will repeat until a "Break Loop" executes.

[Break Loop]

Gets out of the loop.

[Go to Loop Start]

Returns to the start of the current loop.

[|Move Route]

Lets you control movement of events and the hero. For details, refer to "[Move Route](#)."

[Wait For Move Route]

Keeps the next command from running until the current "|Move Route" is done processing.

[Prepare Transition (Stop Screen)]

Stops drawing the screen. However, processing will still continue.

[Execute Transition]

Executes the screen effect set by "Set Transition," and resumes screen drawing.

[Move During Events ON/OFF]

Sets whether other events can move while an event is running.
If OFF, other events will not move while events are running.

[Go to Title]

Resets the game, returning to the game start point.

[Game End]

Closes the entire game.

[Stop Non-Picture Updates/Resume]

Remembers the currently-displayed screen, and stops all non-picture drawing (that is, characters and the map). "Resume" cancels that state.

Note that "Stop Non-Picture Updates" right after a load will result in a black screen, so please wait a few frames in that case.

* By nature of "remembering the currently-displayed screen," if you "Stop Non-Picture Updates" while pictures are displayed, those pictures will still remain.

[Force Exit Event]

Ignores all the event commands to follow and stops the event.

[Set Transition]

Sets the transition type used by "Execute Transition." Transition graphics (must be a grayscale image) are read from **System Database Type 11**, "[Transition Types](#)".

[Erase Event]

Erases the specified event. Erased events cannot be executed again unless you move to another map.

[Wait]

Pauses event processing for the specified number of frames. You can use variables for the frame duration (example: 2000000 = Normal Variable #0).

[Loop [X] Times]

Loops for the specified number of times. Functions like "Loop (Repeat)," but this lets you choose a number of times to execute. You can use a variable for the loop count.

[Label Handling]

Lets you set and jump to labels. "**Labels**" are like markers in your processes; if you place one, you can jump to that position. However, **you can only move to labels within the same event.**

Note that performing "**Jump to Label**" in a numbered loop will **reset the remaining execution count of that loop.**

*** Label Handling Addendum (04/13/2010)**

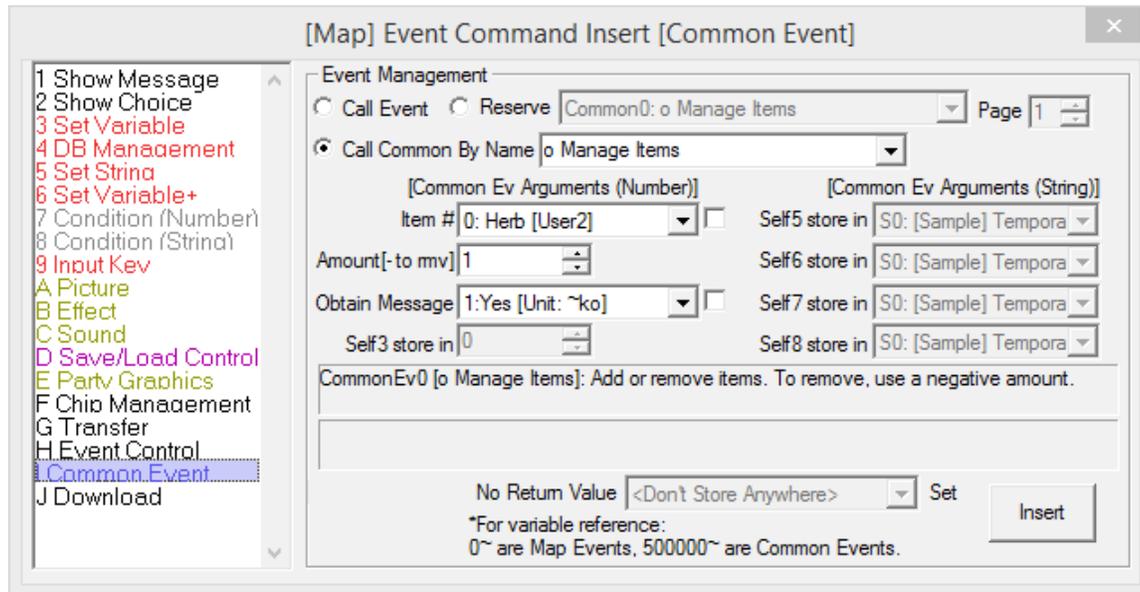
If you **jump to a label inside a conditional case**, and that condition has an "**Else Case**," regardless of what case it is, once the case is done processing, the "Else Case" will be processed next.

This phenomenon occurs because when entering a condition case by label-jump, internally, it thinks none of the conditions were met. This behavior has been left as-is for the sake of optimization and stability, so please just work around it if needed.

(In general, label-jumping into a conditional case is not recommended.)

Event Command "Common Event"

Calls Common Events.



[Function Explanations]

[Call Event]

Execute a Common Event, or another event in the same map.

If a Common Event is set up to accept arguments, you can specify them here.

(In the image above, "Item #" and "Amount[- to mv]" are arguments. These were defined in "[Define Arguments](#)," accessed from the ★. **"Set" button in the Common Event Editor.**)

When specifying with variables, 0~ refers to Map Events, and 500000~ refers to Common Events.

[Special Function] When calling a Map Event, if you specify Page 0, the currently-active page is called.

(* To avoid having this confuse beginners, **the box's minimum value is set to be Page 1. Thus, to enter 0, you must type it in manually.**) If the event does not exist, it is ignored.

* The values you provide for Argument 1 are put into that Common Event's Self Variable 0, Argument 2 goes into Common Self Variable 1... In this way, they are stored as initial values for the Common Event.

For String arguments, Argument 1 goes into Self Variable 5, Argument 2 goes into Self Variable 6, etc.

[Reserve]

Sets the specified event up to execute once the current event process is finished.

[Call Common By Name]

Calls a Common Event by its name. If multiple Common Events with that name exist, the one with the largest number is called.

~~You can also call saved Common Event files (.common) by their filename.~~

~~* However, for files that contain multiple Common Events, only the first one is executed.~~

~~You can also provide arguments, but must specify all of Self0~3 and 5~8.~~

~~→ Calling .common files is no longer supported as of 2.00.~~

[Return Value]

If a Common Event has "Return Result?" set, you choose the variable to which that return value is assigned here. You can also have it not be assigned anywhere.

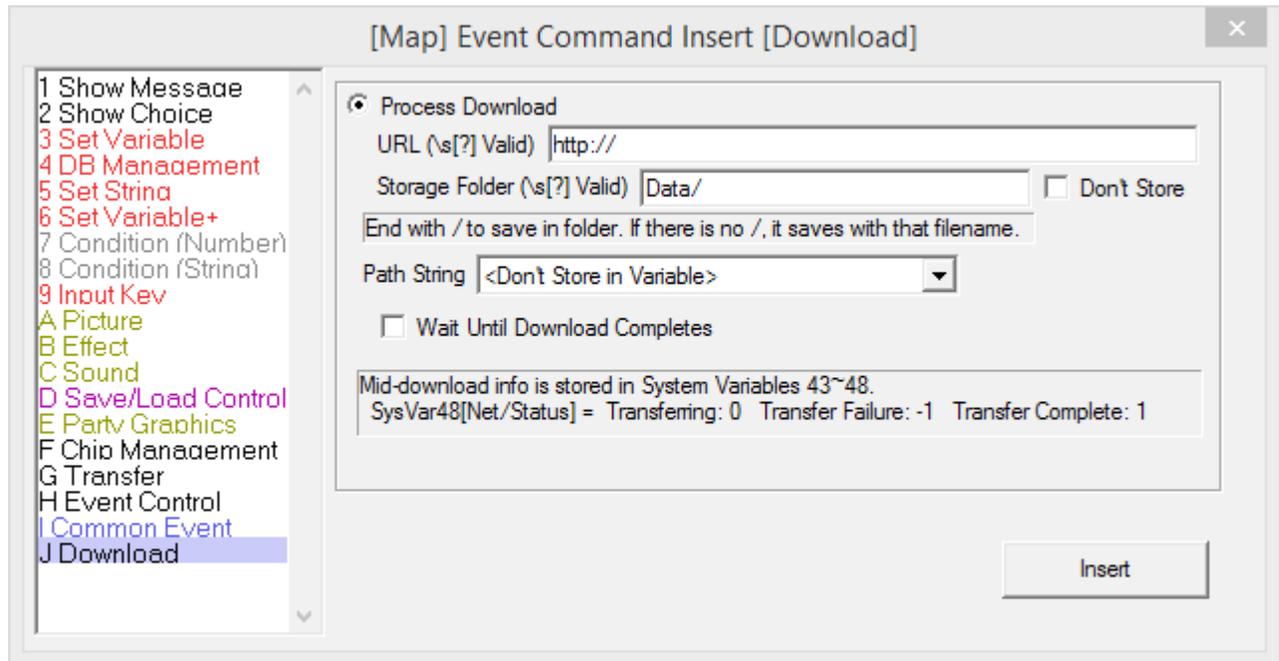
[Secret Command]

If you **click the pull-down menu while holding the Ctrl Key**, you can change the Common Event while keeping the argument values.

This should surely be useful if you change a Common Event's number, and only want to change the number and not the arguments.

Event Command "Download"

Downloads files from the internet, or calls CGIs.



[General Explanation]

- To use this command, you must check "**Enable Advanced Commands**" in "Editor Options."
- Each of the text files, in addition to normal strings, can also use special codes (such as \s[?]).

[Section Explanations]

1. URL

Specify the URL to connect to.

2. Storage Folder

The save location of the downloaded file. Use a relative path from the folder the EXE is in (i.e. "Data/"). If it ends with a "/", the file will save inside that folder. If there is no "/" at the end, the file is downloaded and saved with that filename.

3. Path String

You can store the contents of the download in a String Variable.

4. Wait Until Download Completes

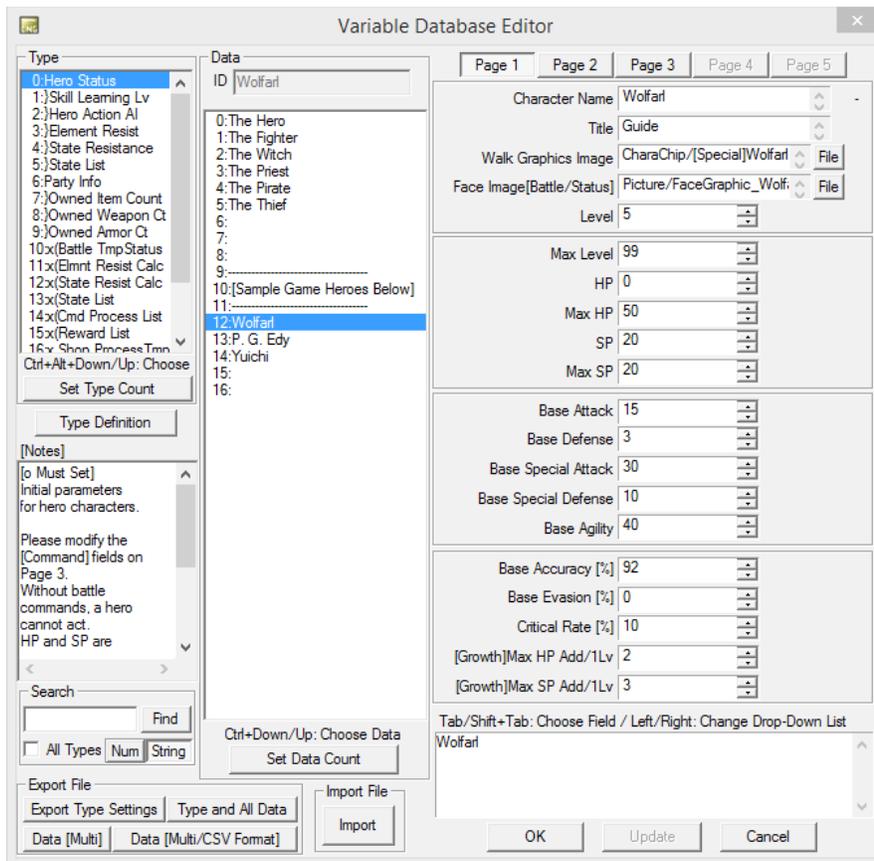
If checked, event processing is stopped until the download completes.

[Secret Trick]

If you specify "**Data/_NetDownloadData**" as the storage folder, then the next time the game starts, the files in that folder will be moved into the "**Data**" folder, preserving folder structure. (If files exist already, they will be overwritten.)

What can you do with this? It allows you to update files that cannot be replaced mid-play, like "**BasicData.wolf**." In this case, please ask the player to manually reboot the game.

[Databases]



(Sample Game's Variable Database)

- What is a Database (DB)?

A database is a "place to store data," in which you can create "database types" of up to 100 input boxes, each of which can be set to hold either "a number" or "a string." Each one of these fields can be assigned a name.

- Section Explanations

- "Set Type Count" Button

Change the maximum number of types in this database.

- "Type Definition" Button

Configure the fields for the selected database type.

- "Set Data Count" Button

Change the maximum number of data entries in the current database type.

- "Page 1~5" Buttons

Switch between the pages of fields. Each page shows 20 fields at once.

- "Export File" and "Import File" Sections (At Bottom)

Functions to save type settings or data as files, and load in data from those files.

- Drag & Drop

If you drag and drop a database type/data file created with Export File onto the window, that file will be imported. (What is imported depends on the type of the file.)

- Shortcut Keys

Tab/Shift+Tab: Move Position in Data Fields

Ctrl+↑↓: Move Position in Data Entries

Ctrl+Alt+↑↓: Move Position in Database Types

Ctrl+S: Save Database

X: Cut

C: Copy

M: Multi Copy

V: Paste

Delete: Delete

- Right-Click Menu

Cut	X
Copy	C
Multi Copy	M
Paste	V
Delete	Del
Export Type Settings to File	
Export Type and All Data to File	

Cut	X
Copy	C
Multi Copy	M
Paste	V
Delete	Del
Export Data to File	
Export Data as CSV File	

These correspond to shortcut keys and the Export File buttons.

- There are three separate databases.

- [System DB](#) ... A database created by the tool's developer (SmokingWOLF). It is used to store maps, BGM definitions, and other key information used in the game.

Changing any of the type definitions is not recommended; it may cause the game to stop working.

- [User DB](#) ... A database for users of the tool to store information that generally will not change in-game (such as attack/defense stats for equipment). Naturally, users must define all types themselves.

- [Variable DB](#) ... A database to manage information that changes in-game, like HP and MP.

Unlike the User DB, the values in this database can be changed mid-game, so it is extremely handy for storing things like character status.

For those who know C programming, you can think of it as being the same thing as a "struct."

[Hidden Commands]

While on the Type or Data lists, the following inputs can be used to insert/remove types or data entries.

- **Ctrl+Shift+F6:** Inserts a new type/data entry.

- **Ctrl+Shift+F7:** Removes the selected type/data entry.

However, using these functions will shift all type/data numbers afterward.

Using these has a very high chance of making your entire game break, so please be very careful with them.

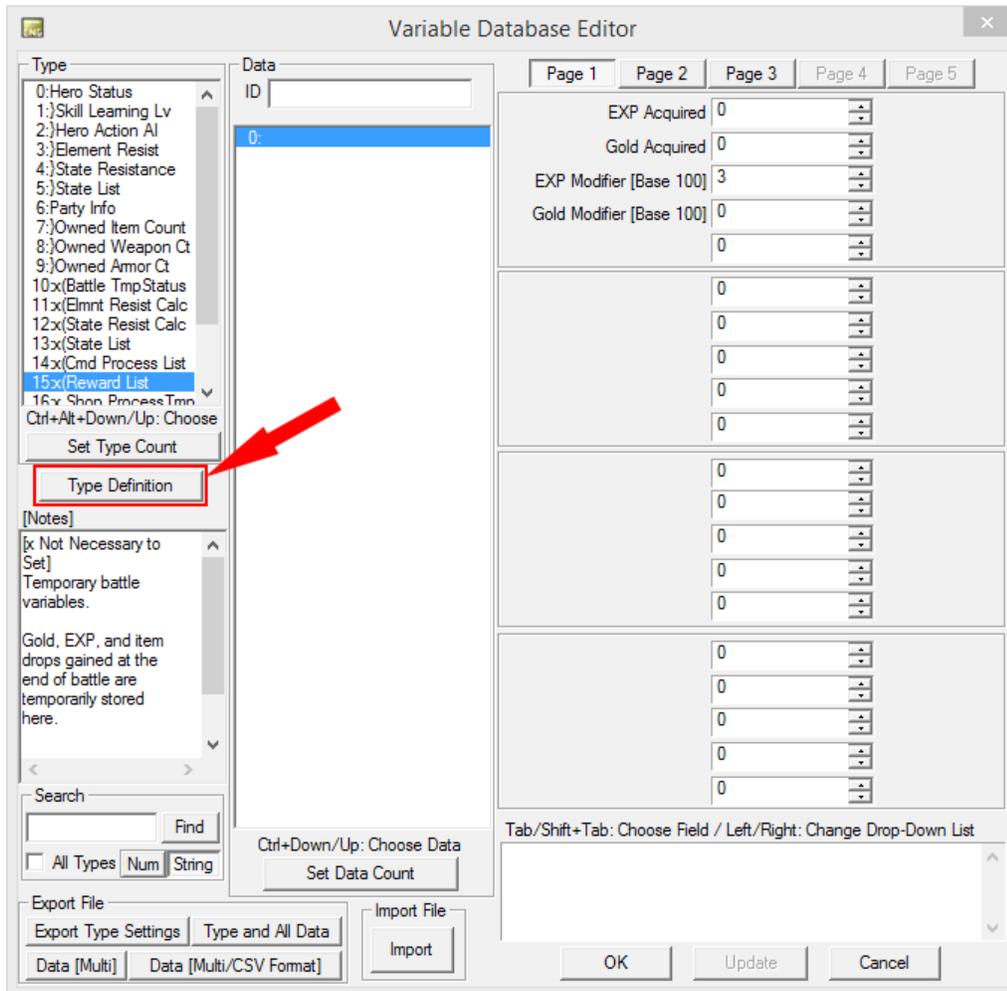
There is one other secret shortcut for experts.

- **Ctrl+R:** Return to the previously-selected Common Event. Press repeatedly to come and go.

→ This shortcut is handy when copying parts from another event.

Database: [Type Definition]

While editing a database, clicking the button below lets you configure the type you have selected.



When you click this button, the following window will be shown.

Type Definition

Type Name Data ID Specification System DB Type

Field Definition (Example: Character Name, HP, Critical Line, etc.)

Page 1 Page 2 Page 3 Page 4 Page 5

Field #	Field Name	Default	Field #	Field Name	Default
Field 0	Character Name	0	Field 10	Base Attack	10
N/A	Number String - Special		N/A	Number String - Special	
Field 1	Title	0	Field 11	Base Defense	10
N/A	Number String - Special		N/A	Number String - Special	
Field 2	Walk Graphics Image	0	Field 12	Base Special Attack	10
N/A	Number String - Special		N/A	Number String - Special	
Field 3	Face Image[Battle/Status]	0	Field 13	Base Special Defense	10
N/A	Number String - Special		N/A	Number String - Special	
Field 4	Level	1	Field 14	Base Agility	10
N/A	Number String - Special		N/A	Number String - Special	
Field 5	Max Level	99	Field 15	Base Accuracy [%]	90
N/A	Number String - Special		N/A	Number String - Special	
Field 6	HP	50	Field 16	Base Evasion [%]	0
N/A	Number String - Special		N/A	Number String - Special	
Field 7	Max HP	50	Field 17	Critical Rate [%]	3
N/A	Number String - Special		N/A	Number String - Special	
Field 8	SP	30	Field 18	[Growth]Max HP Add/1Lv	8
N/A	Number String - Special		N/A	Number String - Special	
Field 9	Max SP	30	Field 19	[Growth]Max SP Add/1Lv	8
N/A	Number String - Special		N/A	Number String - Special	

Insert, Delete, and Copy Fields

Field Insert Delete Copy Paste To

*Can use \udb[1:0] (Type 1, Data 0) or \udb[1:2:3] (Type 1, Data 2, Field 3's value) in names.
 *d[0] will be converted to Field #. Indicate difference from Field # with the value in [].
 (Ex: \d[-1] in Field 4 converts to 3) \d[?] can also be used in DB references.

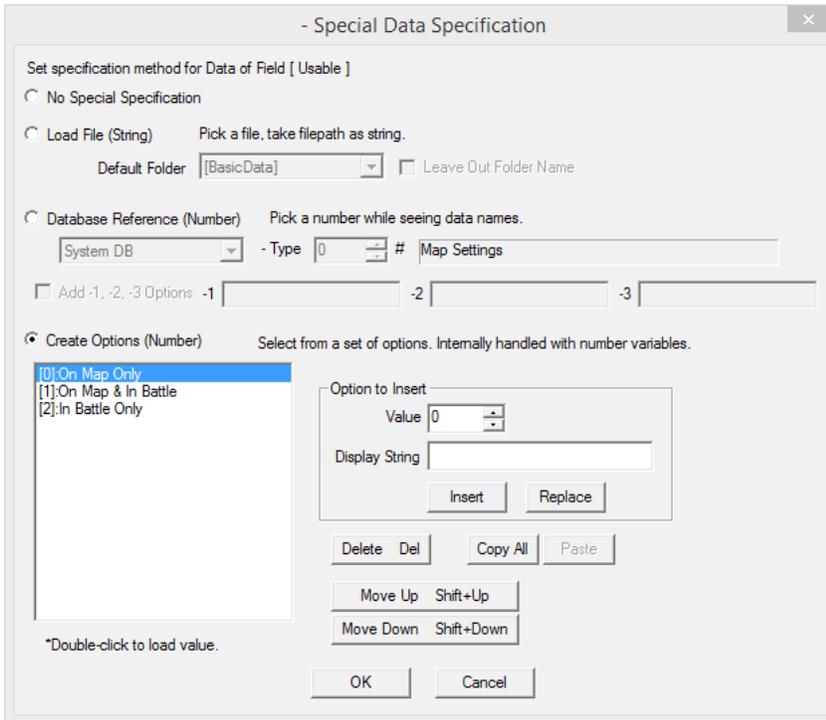
OK Cancel

[Section Explanations]

- **Type Name** ... Set the name of the database type.
- **Data ID Specification** ... Set how you define data IDs. You have the following options.
 - **Manual** ... Specify the data IDs yourself.
 - **Same As First String Data** ... The contents of the first string field becomes the data ID. This is good for when an item name or character name comes first.
 - **Same As Preceding Type Data ID** ... The data IDs for the type before this one provides the data IDs for this type.
 - **From Type in DB** ... The data IDs for the specified type in the specified DB provide this type's data IDs.
- **Field 0~99** ... Set names for each field. You can use special codes like \udb[A:B] (User DB Type A, Data B's Data ID), \cdb[A:B:C] (Variable DB Type A, Data B's Field C), or \d[X] (Field number +1*).
 - * For example, if you write \d[4] in Field 12's input box, it becomes 16. You can even insert it into other things, like \udb[3:\v[4]:\d[0]].
- **Default** ... For number fields only, set the initial value. These default values are used when a data entry is erased or cut, or you add new data entries by increasing the data count.
- **"N/A / Number / String" Buttons**
 Set whether each field stores a number or a string.
 If you choose N/A, all fields from that point on will not be used.
- **"- Special" Button**
 Click to define special kinds of input.
 Depending on your setting here, "number" or "string" will be chosen automatically. For details, see below.

[About "- Special Data Specification"]

When you click the "- Special" button, the following window will be shown.



You can choose the following four types of special data specification. The words in ()s indicate the type of data that gets stored with that method.

- No Special Specification

Do not use a special method.

This is selected by default. In this case, you are free to choose whether it stores "number" or "string" data.

- Load File (String)

Displays a file select window, and stores the file path of the selected file. Useful when you have database fields for storing image filenames.

When "Leave Out Folder Name" is checked, the folder name is removed on selection, storing only the filename in the data box.

Also, when a field uses this setting, you can drag and drop a file onto the field to input that file's address (the path name starting from "~\Data"). For example, if you drag the file "C:\WOLF\Data\Graphic\AAA.png" onto the textbox, the textbox will have "Graphic\AAA.png" inputted into it.

- Database Reference (Number)

Lets you select a number by looking at the data names of another database type (the values in []s are the numbers that will be stored). Useful for things like defining equipment for a party member.

Also, in addition to the database values (that is, the values starting from 0), you can name extra options for the values -1 to -3.

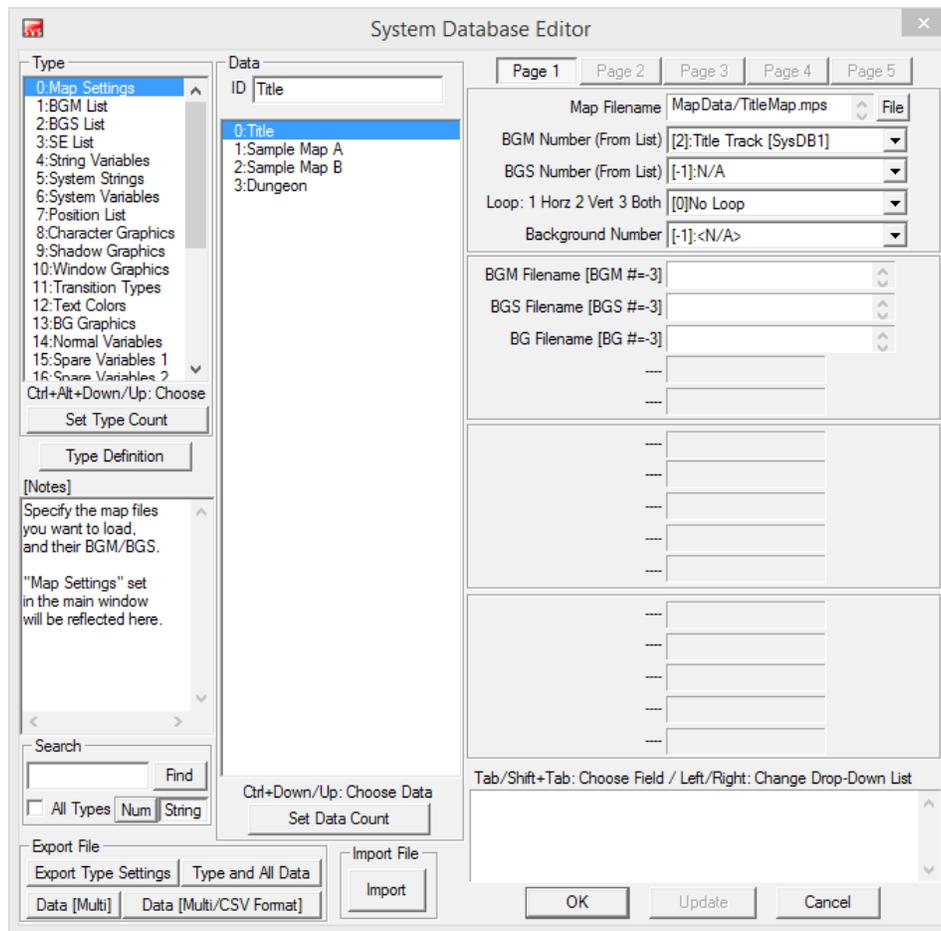
- Create Options (Number)

Manually create choices that correspond to numbers. If you make choices here that are limited to the only values that should go into that field, it can cut down on mistakes when entering values.

Database: [System Database]

A database created by the tool's developer (SmokingWOLF).

It is used to store maps, BGM definitions, and other key information used in the game.



- Using the System Database

Users of the tool (that is, you) should follow the type definitions pre-defined by SmokingWOLF, changing only the data contents. In general, please refrain from clicking "Type Definition" and changing things there.

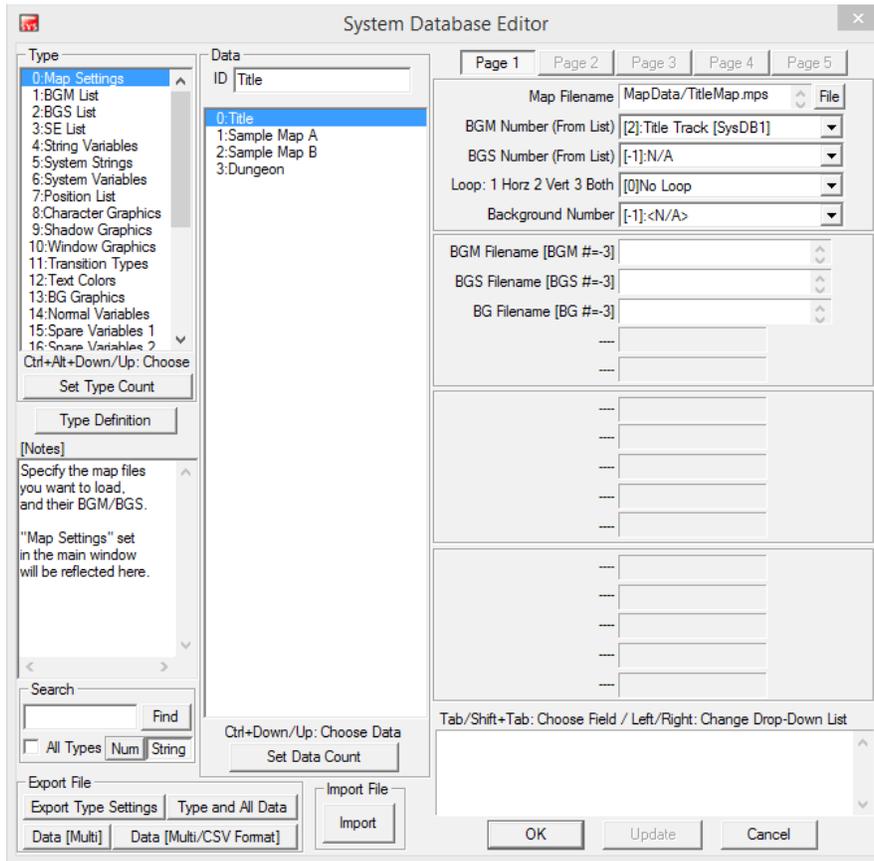
Note that files like maps, BGMs, BGSs, and SEs must be registered in the System Database to be used.

* Important: The "Type Definition" for Types 1~13 may be updated in accordance with the behavior of WOLF, so you generally should not change them.

For details on each type, refer to "[Type Explanations](#)."

System Database: [Type Explanations]

Explanations of each type in the System Database.



This explains all of the 24 types, and the data fields of each.

[Type 0: Map Settings]

Stores data about maps.

- **Map Filename** ... Specify the map file (.mps file).
- **BGM Number** ... Specify the BGM number (from System DB Type 1).
- **BGS Number** ... Specify the BGS number (from System DB Type 2).
- **Loop** ... Define loop settings. - 0: No Loop, 1: Horizontal Loop, 2: Vertical Loop, 3: Loop Both Ways

[Type 1: BGM List / Type 2: BGS List]

Stores BGM and BGS used in-game.

- **Filename** ... Specify the sound file.
- **Play Volume** ... Set the volume. If 0, it uses the default.
- **Play Tempo** ... Set the tempo. If 0, it uses the normal tempo.
- **Loop Start Point** ... Set the position to loop back to (in milliseconds) once the sound plays to the end.

[Type 3: SE List]

Stores sound effects used in-game.

- **Filename** ... Same as Type 2.
- **Play Volume** ... Same as Type 2.
- **Play Tempo** ... Same as Type 2.
- **Memory Free Mode** ... Set whether to free the sound from memory after it plays. If set to 1, it will be freed right after playback. Setting it to 1 may be good for sounds that rarely play.

Note: If set to 1, the sound will be loaded immediately at game start. (Does not function in current Ver. 2.20.)

* Incidentally, all SEs are freed from memory when you change maps.

[Type 4: String Variables]

Stores the names of String Variables (input as Data IDs). You can increase the number of String Variables by increasing the data count.

[Type 5: System Strings]

Stores the names of System Strings used on the developer side. There is no need for users to change them.

[Type 6: System Variables]

Stores the names of System Variables used on the developer side. There is no need for users to change them.

[Type 7: Position List]

Stores positions saved via the "[Save Position to DB](#)" function. It also stores your game's default start position via the "[Set Game Start Position](#)" function. **Data 0: Starting Position** stores your game's default start position for when you run **Game.exe**. **Do not delete Data 0: Starting Position!**

- **Map ID** ... Reads the Map ID from System DB Type 0.
- **X Position** ... The X coordinate of the destination.
- **Y Position** ... The Y coordinate of the destination.

[Type 8: Character Graphics]

When changing character graphics in "[Move Route](#)," the graphic numbers refer to here.

- **Filename** ... Specify the character graphic file to change to.

[Type 9: Shadow Graphics]

- **Filename** ... Specify the shadow graphic file.

[Type 10: Window Graphics]

Specify the textbox "waiting for input" pause graphic (Data #0), the choice window graphic (Data #1), and the choice cursor graphic (Data #2).

- **Graphic File** ... Specify the graphic file to use.
- **Vertical Divisions** ... Set how many sections to divide the graphic into. The "number of divisions" ends up being the number of animation frames.

* Vertical Divisions is only usable for "Pause Graphic (Data #0)" and "Choice Cursor Graphic (Data #2)."

[Type 11: Transition Types]

Stores different transition types (used by event command "[Event Control](#)").

- **Filename (Monochrome)** ... Specify the transition graphic file. Transition graphics must be grayscale (monochrome).

[Type 12: Text Colors]

Set font colors that can be used in-game.

- **Red (0-255)** ... Set the level of red.
 - **Green (0-255)** ... Set the level of green.
 - **Blue (0-255)** ... Set the level of blue.
- 255,255,255 is pure white, 0,0,0 is pure black.

* Some data entries are used on the system side.

Data #0: Default text color.

Data #10: Color of keyboard input text.

Data #11: Background color when converting Japanese text during keyboard input.

Data #12: The color of the underline in keyboard input.

Data #13: Color of ruby text ("[Special Characters](#)").

[Type 13: Background Graphics]

Stores background and fog graphics.

To use in-game, set System Variable 61 (Fog Number) or 66 (Background Number) to instantly change them.

- **Graphic File** ... Specify the background/fog graphic file.
- **X Move Speed** ... Set the speed at which it automatically moves on the X axis.
- **Y Move Speed** ... Set the speed at which it automatically moves on the Y axis.

* Background/fog graphics must be a size greater than or equal to the resolution.
(For example, if the resolution is 320x240, the graphic's size must be at least 320x240.)

[Type 14: Normal Variables]

Stores the names of Normal Variables (Data ID = variable name).

You can increase the number of variables by increasing the data count.

[Type 15~23: Spare Variables 1~9]

Stores the names of Spare Variables (Data ID = variable name).

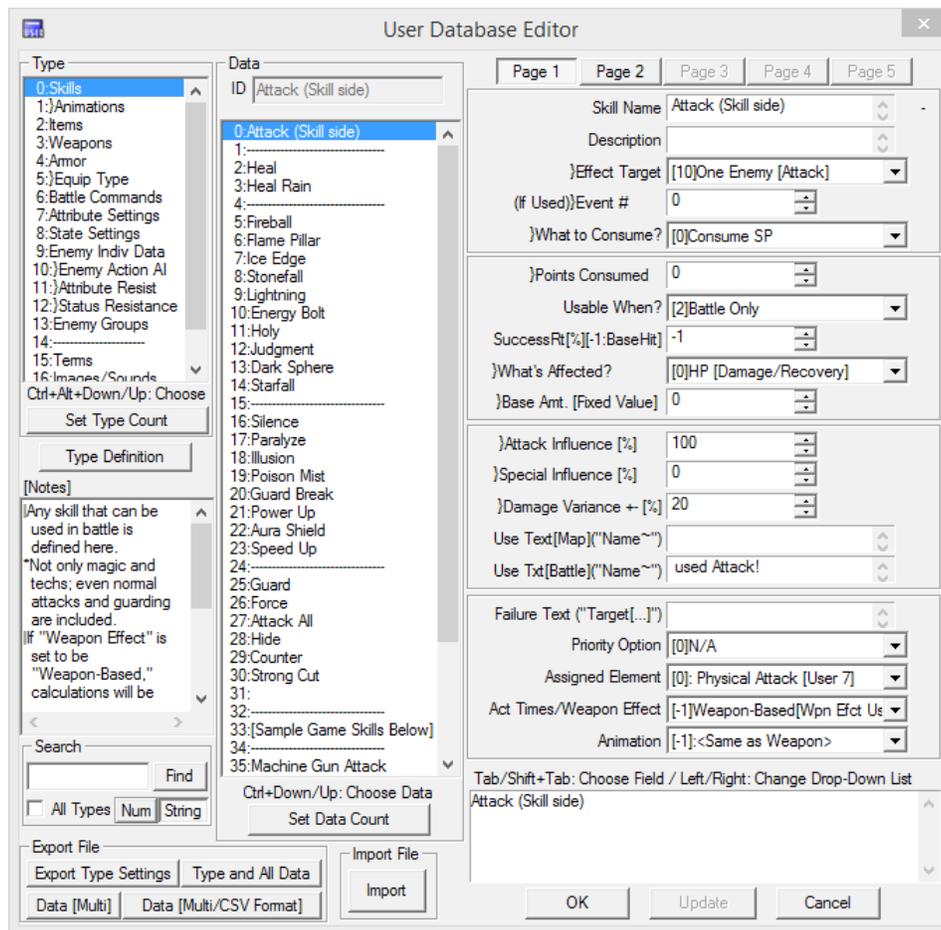
You can increase the number of variables by increasing the data count.

[Type 24: Face Graphics]

The Data IDs of this type can be checked in the "[Show Message](#)" window. The IDs normally have no other use.

- **Face Graphic File** ... A field for storing face graphic files. However, nothing will be shown unless you make something on the Common Event side to display faces.

Database: [User Database]



(Sample Game's User Database)

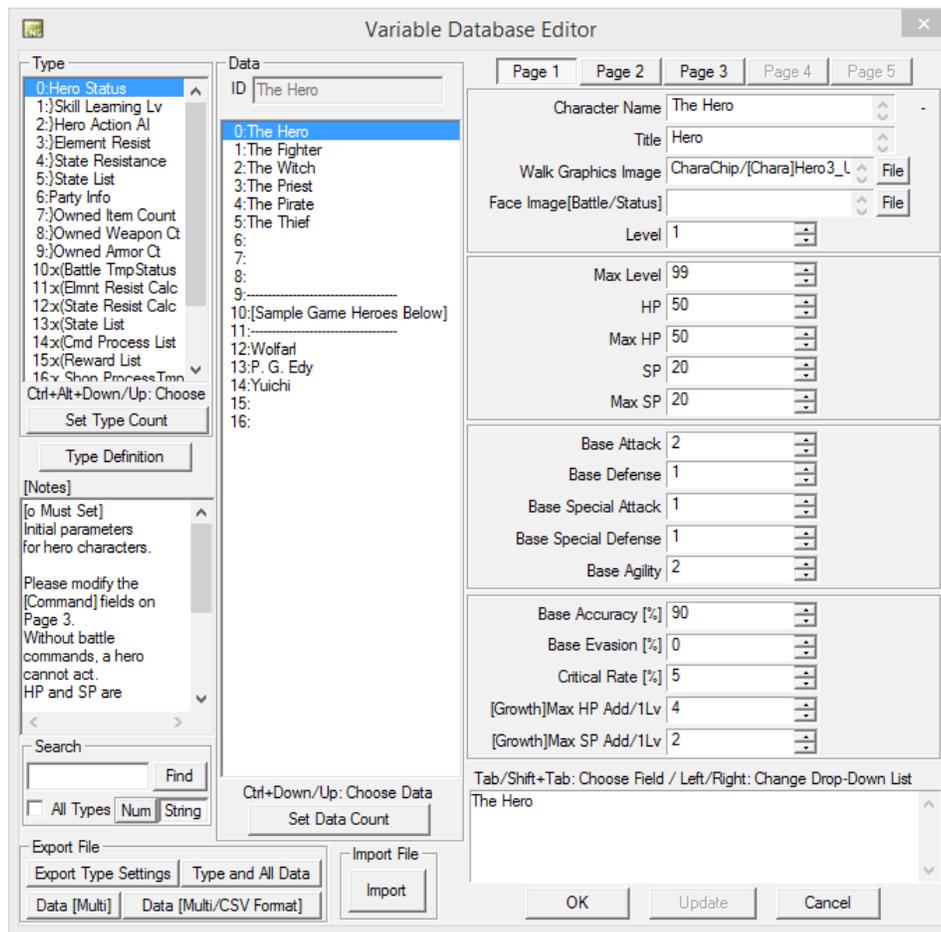
- Using the User Database

It is for storing information that generally will not change in-game, like weapons or enemy stats at the start of battle for RPGs. Generally, users must create these types from scratch.

The data in this database cannot be rewritten in-game.

To read data from the User Database, use the event command "[DB Management](#)."

Database: [Variable Database]



(Sample Game's Variable Database)

- Using the Variable Database

It is a database for managing information that changes in-game, like character status or the state of enemies in battle (i.e. HP), so it can be rewritten mid-game. Generally, users must create these types from scratch.

The values you input into this database are used as initial values for when the game starts.

To rewrite data in the Variable Database, use the event commands "[Set Variable](#)" or "[DB Management](#)."

To read data, use the event command "[DB Management](#)."

[Resource Standards]

- [Images](#)
- [Audio](#)
- [Fonts](#)

- Images -

Make sure all image resources are 'PNG with alpha blending', 'JPG', or 'BMP'.

- * As of Ver. 2.20, PNGs without alpha blending appear to no longer show as transparent. Sorry for the trouble, but please make adjustments.

[Map Chips]

There are two types of map chips defined in "[Tileset Settings](#)": Basic Tilesets, and Auto-Tiles.

- Basic Tilesets

Basic Tileset Format:

	Width	Height
320x240 Resolution	128 pixels	16 x Any number
640x480 Resolution	256 pixels	32 x Any number
800x600 Resolution	320 pixels	40 x Any number

- * With 320x240 resolution, you will be warned of issues if height is 4000 pixels or greater.

Make your images have horizontal rows of 8 tiles, with however many rows you need vertically. At 320x240 resolution, tile width is 16, times 8 is 128 pixels. At 640x480, tile width is 32, times 8 is 256 pixels. Make the height something that is divisible by the chip size.

- Example for 320x240 Resolution



This image is 128 pixels wide, enough for 8 tiles horizontally. The vertical size can be increased as much as necessary.

- * Note: If the blank areas are colors besides black, things may look a little strange during map editing, but it will not affect display in-game.
- * Also, any non-32 bit (PNG with alpha channel) may not display properly in the Editor. This also has no impact during the game.

- Auto-Tiles

Auto-Tile Format:

	Width	Height
320x240 Resolution	16 x Animation frames	80 pixels
640x480 Resolution	32 x Animation frames	160 pixels
800x600 Resolution	40 x Animation frames	200 pixels

Auto-tiles automatically connect when placed next to each other, and can animate. Images should be 5 tiles tall, with a width anywhere from 1 tile to ∞ tiles.



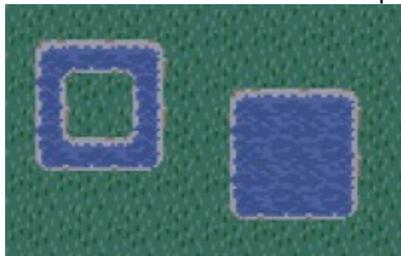
The order of tiles from top to bottom:

- **Connections Toward Center**
- **Vertical Connections**
- **Horizontal Connections**
- **Outside-Facing Connections**
- **Fully Surrounded**

This image results in a 4-frame animation. Animation animates across the tiles placed horizontally.

[Usage Example]

The tiles shown above are auto-placed like so.



[Warnings!]

Map chips will not display properly in the Editor in some cases. Please take note of the following. It appears to depend on your Windows version.

- [1] If blank areas are a color other than black, display may be slightly strange in Map Editor.
- [2] When using map tiles that are 256-Color PNGs with no transparent color set, map chips placed in the Editor may be invisible. Setting a transparent color will make them appear.
- [3] If a full-color PNG image has no alpha channel (transparency information), map chips placed in the Editor may be invisible. Setting alpha channels for the image should make them appear.

However, all of these problems should only occur in the Editor, and will not affect gameplay.

[Character Chips]

The layout for character chips differs depending on two options in "Basic Game Settings":

Character Animation Patterns (3- or 5-Pattern) and **Character Sprite Direction Type (4- or 8-Directional)**.

Please refer to the images below for how to lay out the frames.

(→ indicates what the character's direction should be in that frame).

Character Chip Format:

	Width	Height
All Resolutions	Anything	Anything (Multiple of 4)

<p>3-Pattern, 4 Directions</p>	<p>3-Pattern, 8 Directions</p>
<p>5-Pattern, 4 Directions</p>	<p>5-Pattern, 8 Directions</p>
<p>Example: Image for 3-Pattern, 4 Directions</p>	
<p>* Line up walk animation frames as given by the number of patterns. It does not matter how large the size is.</p>	

- Special Functions -

As a special function, ending a character graphic filename with "T.png" or "TX.png" allows you to place idle and movement graphics in a single image.

[T.png Case]
 Assuming 3-pattern animations:
 The frames that were normally "right foot forward, stopped, left foot forward" should become "stopped, right foot forward, between, left foot forward," making 4 horizontal frames (pattern number +1). The height stays at 4 frames.
 When the character is not moving, the "stopped" frame is displayed, and when they are moving, it animates through the right three frames.

	<p>A layout for 3-pattern, 4-direction walking, plus an idle state. The ones with a pink o are idle frames.</p>
--	---

[TX.png Case]

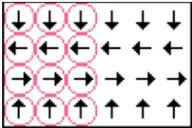
Assuming 3-pattern animations:

The frames that were normally "right foot forward, stopped, left foot forward" should become "idle 1, idle 2, idle 3, right foot forward, between, left foot forward," making the width [pattern number x 2] frames. The height stays at 4 frames.

When the character is not moving, it animates in the order "idle 2→3→2→1→2→3→ ... ", and when they are moving, it animates through the right three frames.

*TX.png Idle Animation is **OFF by default for Hero and Members(Party)**, so you need to manually turn Idle Animation ON for your Party.

You will need to create an Event (Common is recommended) for turning your Party's Idle Animation ON. Then you must do the following process:
From "[Command Insert Window]" select "Event Control" → "IMove Route".
Please select "Hero" from the main drop-down list and click "Idle Animation ON".
(Just repeat this process for the other Members of the Party).
If you do not do this, you will not have any Idle Animation for your Party.



A layout for 3-pattern, 4-direction walking, plus 3 idle frames.
The ones with a pink \circ are idle frames.

[Wait-For-Input Pause Graphic]

An image displayed while waiting for input to advance text. Stored in System Database Type 10, Data #0.

Wait-For-Input Pause Graphic Format:

	Width	Height
All Resolutions	Anything	Anything (Divisible by animation frame count)

You can add animation frames to the pause graphic infinitely by extending the height.
The number of animation frames is given in System Database Type 10, Data #0.

Wait-For-Input Pause Graphic Example



In the case of this image, going to System DB Type 10, Data #0 and setting Vertical Divisions (animation frames) to 6 will make it animate properly.

[Choice Window Graphic]

The base image for the choice selection window. Stored in System Database Type 10, Data #1.

Choice Window Graphic Format:		
	Width	Height
All Resolutions	Any multiple of 3	Any multiple of 3

The choice window divides the given image into 3x3 parts and uses them to stretch it out to the right size. As such, there are no restrictions on size.

Choice Window Example



[Choice Cursor Graphic]

The cursor image displayed when selecting choices. Stored in System Database Type 10, Data #2.

Choice Cursor Graphic Format:		
	Width	Height
All Resolutions	Anything	Anything (Divisible by animation frame count)

You can add animation frames to the cursor graphic infinitely by extending the height. The number of animation frames is given in System Database Type 10, Data #2.

Choice Cursor Graphic Example



In the case of this image, going to System DB Type 10, Data #2 and setting Vertical Divisions (animation frames) to 2 will make it animate properly.

- Audio -

There are three types of audio, BGM, BGS, and SE, each with their own supported audio formats.

BGM = Supports MIDI, WAV, MP3, OGG
BGS = Supports WAV, MP3, OGG
SE = Supports WAV, MP3, OGG

* Certain formats of WAV and MP3 may not be able to play.
Trying to play a 24-bit WAV may cause a force-quit on some systems.

[What is OGG Format?]

An audio format with compression and quality on par with MP3s. Though more obscure, unlike MP3, there is no copyright on licenses for its codec technology, so it is safe in that using it is less likely to get you in trouble.

Also, OGG promises higher sound quality at the same file size compared to WMA or MP3.

You can quickly convert from MP3 or WAV format to OGG using "Rip! AudiCO", linked in the "[Helpful Tools](#)" section.

[Warning About MIDI Format]

When MIDI playback is set to "Hardware," MIDI songs may not play properly on Windows Vista. Make sure you take this into account. There seem to be no such problems with Software playback. Also, with Hardware playback, since MIDI files are played on the PC side instead of the game side, songs may not play as you intended. Unless you have some particular reason for it, "Software" playback is strongly recommended.

- Fonts -

WOLF RPG Editor can load in included font files.
Place font files (.ttf) in one of the following folders.

- Next to the game files (Same place as Game.exe)
- Inside Data folder *(Note that if you encrypt the entire Data folder, it cannot be read from within.)

[Font Types]

TrueType Fonts (.ttf file): Should be usable without problems.
However, I have heard some fonts will not work in WOLF.
(They may be OpenType fonts with a .ttf extension?)

OpenType Fonts (.otf file): There is a high chance these will not work.

* About Loading External Font File

If a font is not installed on the player's PC, it will automatically become MS Gothic. However, font files can be packaged with the game and loaded. To do this, **include the font file (.ttf or .ttc format only)** in the **Data folder** (* only if Data folder is not entirely encrypted) or in the **same place as Game.exe**.

* Checking Font Names

Please enter the font name exactly. On Windows, you can find font names by right-clicking a TTF file, clicking Properties, and in the Details tab, it is listed as "Title."

Also, if you are releasing to English audiences, you will need to input the English font name.
(This is hard to check, but can be done with tools like FontForge.)

* Note that many fonts have strict usage rules, so be careful with any fonts you use.
In particular, you cannot package in fonts that do not allow redistribution.

[System Variable/String List]

System Variables are parameters for the game system that can be set with commands like "Set Variable."
Look further down for System Strings.

·[System Variable List](#)

·[System String List](#)

[System Variable List]

0: Face Number

Holds the face graphic number. When inputting a message, putting "@number" in the first line will have it store that number here as the current face graphic.

1: Message Box X

2: Message Box Y

Specifies the position where text is drawn for Show Message.

The position given here serves as the **top-left** coordinate of the text. If set to -1, the default position is used.

3: Choice Box X

4: Choice Box Y

Specifies the position of the window for the Show Choice command.

The position given here serves as the **upper-center** of the choice window. If set to -1, the default position is used. (It is apparently not actually centered, but about 4 pixels to the left. People have already made minute adjustments around this, so it has been left that way.)

5: Text Wait X (-1=Auto)

6: Text Wait Y (-1=Auto)

Specifies the position of the wait-for-input pause cursor on messages, relative to the top-left of the message window. The position given here serves as the **upper-center** of the pause cursor.

7: Initial Choice (0, 1, 2...)

Specifies which option to start on initially the next time a choice appears. 0 is the top choice, 1 is the second choice, etc. This variable is **automatically set to 0 once the choice window closes**.

8: Base Font Size

Specifies the default font size used by text in Show Message and Picture's string pictures.

9: Text Speed (X characters/second)

Specifies the message speed of Show Message, in units of "X characters per second."

10: Text Wait (X frames)

Specifies the wait time between the Show Message command executing and the first character of text appearing. You can use this wait time to match up the timing if you are displaying a message window.

11: Picture Text Speed (X characters/second)

Specifies the message speed of Picture's string pictures, in units of "X characters per second."

12: [Read Only] Showing Text? (1=ON)

A read-only variable. Holds 1 when a message is being displayed, and 0 at any other time.

May be useful in parallel events.

13: [Read Only] Event Active (1=ON)

A read-only variable. Holds 1 when a non-parallel event is executing, and 0 at any other time.

Note that events called from a parallel event, regardless of their run condition, are considered to be parallel events as well.

14: Choice Box Up? (1=ON)

Specifies whether the window graphic is shown (1) or not (0) when using Show Choice.

15: Direction-Key Repeat First Wait (Frames)[Until Key Press]**16: Direction-Key Repeat Next Wait (Frames)[Until Key Press]**

Specifies the time before direction keys repeat, when using the "Input Key: Take Input" with "Wait For Input" on. Imagine the "repeat" when you hold down the button as being a rhythm that goes "(press) da...dadadada."

The "first wait" is the time after the first "da...", and the "next wait" is the time between each "da" in the "dadadada." If you do not want to repeat, set these to extremely large values (like 10,000).

17: Choice Confirm Sound Effect (Sys3)**18: Choice Cursor Sound Effect (Sys3)****19: Choice Cancel Sound Effect (Sys3)**

Specifies the sound effects used for Show Choice: confirm, cursor (the sound of moving the cursor up/down), and cancel. Sound effects are read from System DB Type 3, "SE List."

20: [Read Only] Choice Cursor (0, 1, 2... N/A: -1)

A read-only variable. Holds the current position of the Show Choice cursor.

Could be used for a parallel event that displays descriptions of each choice.

21: Text Shadow? (0=OFF, 1+=ON)

If this variable is set to 1 or more, a black shadow is added to text in Show Message and string pictures, at the distance given here in pixels.

22: Choice Cursor Draw (0=Normal, 1=Add)

Specifies whether the Show Choice cursor is, in the same terms as Picture's "Display Type," drawn in Normal mode or Add mode.

23: Ruby Font Size [Implemented in Ver. 2.20]

Specifies the font size of ruby text when using the special code `\r[A,B]`.

24: [Read Only] Load Check (1=Pass 0=Fail)

A read-only variable. Always 1, so it can be used to check if save data exists when loading variables from a save in Save/Load Control. Example: When Save Data 1 does not exist, trying to read System Variable #24 from Save Data 1 will return 0. If Save Data 1 does exist, it will return 1.

25: Current Save (0~)

Stores which save data the current game was loaded from. You can change this value by setting it.

Example: If you load Save Data 3, "Sys25: Current Save" will hold the value 3.

28: [Read Only] Frame Count [Loops at 10,000,000]

A read-only variable. Holds the number of frames elapsed starting from the title screen.

29: [Read Only] Playtime (in seconds)

A read-only variable. Holds the playtime, starting from the title screen, in seconds.

30: [Read Only] Playtime (in milliseconds)

A read-only variable. Holds the milliseconds of playtime, starting from the title screen.

The value will be from 0 to 999 (milliseconds).

31: [Read Only] Program FPS (each second)

A read-only variable. Holds the internal processing FPS.

32: [Read Only] Draw FPS (each second)

A read-only variable. Holds the screen-drawing FPS.

33: Frameskip (0~2)

Specifies the current frameskip setting. The frameskip can be changed with F5.

This is changed on the Config settings, so the value will persist even when the game is closed.

35: [Read Only] Hero Moving? (1=Yes) (Between Tiles)

A read-only variable. Set to 1 only when the hero is moving from tile to tile.

If you activate a parallel event like a menu while moving to another tile, "Player Touch" events at the destination tile will no longer trigger, so this helps to handle that situation.

If the movement speed is very high, it can end up always being 0. (Because you move so fast, you go to the next tile in 1 frame.)

36: Hero Shadow # (Sys9)

Specifies the shadow number for the hero/allies. Shadow graphics are read from System DB Type 9.

37: Party Separation (steps)

Specifies the distance between allies when in a lineup. Starts at 2 (2 steps).

The distance depends on the move distance in Game Settings. **When default character movement is "1 Tile," the separation may have to be set to 0 to make allies stick together.**

39: [Read Only] Keyboard Entry 0=Enter -1=Cancel

A read-only variable. After Keyboard Input completes, this holds whether it was confirmed (0) or canceled (-1).

40: Keyboard String X

41: Keyboard String Y

Specifies where to display the string being typed in Set String's Keyboard Input.

The position given here is the **top-left** of the string.

42: Keyboard String Aliasing [0 No/1 Yes/2+ Edge] (Added in Ver. 2.02)

Specifies whether to have antialiasing (smooth) or edges on the string displayed during Set String's Keyboard Input. The default is 1, with antialiasing. 0 is no antialiasing, 1 is with antialiasing, 2 is no antialiasing + edges, and 3 is antialiasing + edges.

43: [Read Only] Net/File Size (bytes)

44: [Read Only] Net/Estimated Size (bytes) [-1: Unknown]

45: [Read Only] Net/Download Speed

46: [Read Only] Net/Uptime (seconds)

47: [Read Only] Net/Time Left

48: [Read Only] Net/-1:Fail, 0:Doing, 1:Done

Read-only variables. They hold parameters during net functionality.

51: <Key Config Below>

52: Confirm (Keyboard)

53: Cancel (Keyboard)

54: Sub Key (Keyboard)

55: Confirm (Pad)

56: Cancel (Pad)

57: Sub Key (Pad)

58: <End of Key Config>

Key configuration. Assigning keyboard or pad values (100~ or 300~) from Input Key to these lets you change the key/button settings.

59: Repeat Confirm Key? (1=ON) [Wait For Key]

If this is set to 1, during "Input Key: Take Input" with "Wait For Input" enabled, holding down the Confirm key will act as if you are pressing it rapidly. The repeat speed depends on "15/16: Direction-Key Repeat Wait."

60: Repeat Cancel Key? (1=ON) [Wait For Key]

If this is set to 1, during "Input Key: Take Input" with "Wait For Input" enabled, holding down the Cancel key will act as if you are pressing it rapidly. The repeat speed depends on "15/16: Direction-Key Repeat Wait."

61: Fog Number (Sys13)

Specifies the fog graphic (scenery displayed on top of the map and characters). Fog graphics are read from System DB Type 13.

62: Fog X Speed**63: Fog Y Speed**

Specifies the X and Y speed of the fog.

Positive X moves right, positive Y moves down; negative X moves left, negative Y moves up.

64: Fog Draw (0=Normal 1=Add 2=Subtract)**65: Fog Opacity (0~255)**

Specifies the draw type and opacity of fog. These values take effect as soon as they are changed.

66: Background Number (Sys13)**67: Background X Speed****68: Background Y Speed**

Specifies the background graphic (displayed under the map and characters) and its move speed on the X/Y axis.

71: Mouse X**72: Mouse Y**

Holds the current mouse position. Will not go any higher than the game resolution.

Example: At 320x240, 71 (X position) will be from 0 to 320, and 72 (Y position) will be from 0 to 240.

73: Text/Choice Mouse (1: Allow)

Specifies whether to allow using the mouse to advance text or select choices.

If 0, mouse input has no effect. If 1, you can click to advance text or select choices.

74: Mouse Cursor? (1=ON)

Specifies whether to show the mouse pointer.

If 0, the mouse pointer will disappear when it is over the game window.

77: [Read Only] Time [Year]**78: [Read Only] Time [Month]****79: [Read Only] Time [Day]****80: [Read Only] Time [hour]****81: [Read Only] Time [minute]****82: [Read Only] Time [second]**

Read-only variables. They hold the current year, month, day, and time.

It can be used to note the time on save files, or make events occur based on real time.

84: [Read Only] Screen Size (0=320x240, 1=640x480, 2=800x600)

A read-only variable. It holds the current size of the game screen, but for compatibility with old versions, you can only get it for the three resolutions listed: 320x240, 640x480, or 800x600.

Any other screen sizes return -1. Not recommended for use in Ver. 2.20 or later.

Instead, please use "**Sys116: [Read Only] Screen Size X**" and "**Sys117: [Read Only] Screen Size Y**."

85: [Read Only] Draw Mode (0=3D 1=Software)

A read-only variable. Holds the current drawing mode. Since Software mode is generally more intensive, you can use this for such purposes as reducing image effects when it is on.

87~91: Ally 1~5 Shadow (-1=Hero's)

Allows you to set different shadows for each ally, different from the hero's.

95/97: Background Music/Background Sound Volume [%]

Stores the currently-playing BGM/BGS's volume.

These values automatically update when the music changes.

96/98: Background Music/Background Sound Tempo [%]

You can change the tempo of the currently-playing BGM/BGS while it continues to play.

100~102: Background Music/Background Sound/Sound Effect Volume+ [%]

Applies a general volume amplification or dampening to all BGM/BGS/SE.

Unless changed by events, this will always stay at 100(%).

104/105: Scroll X/Y

Specifies the map scroll position.

This indicates the scroll amount in pixels from the map's top-left origin point of 0,0.

108: [Read Only] Commands Per Frame

Returns the number of event commands processed from when this frame began, to when this value was gotten. You could use this to judge where you need to insert waits - when this exceeds a certain value, you may want to insert a wait.

The order of event processing is:

Map Events → **Auto-Start Common Events** → **Parallel Common Events (in order of ID)**

So if you read this System Variable in the last parallel Common Event, that will gauge how many total commands were processed in each frame.

110: Current Random Seed

If you set this seed to the same value, the random numbers you get afterward will always come in the same order. Can be used at times when you want the randomness to be consistent, such as replay functions in shoot-em-ups or automated battle scenes.

111: [Read Only] Window Active? (1=YES)

A read-only variable. Holds 1 when the game window is active.

Can be used for cases like a game that uses the mouse automatically pausing when it is no longer active.

112: [Read Only] Test Play? (1=YES)

A read-only variable. Holds 1 when in Test Play mode.

If you use Set String's "Get File List of Folder" to say "if BasicData.wolf exists, quit the game," you can make it so players cannot use Test Play with the public release.

115: [Read Only] Game.exe Version (x100) [Implemented in Ver. 2.20]

A read-only variable. Returns the version of Game.exe times 100.

Only became usable in Ver. 2.20, so earlier versions return 0.

It can be used when distributing Common Events for different behavior between a specific version and previous versions. (Especially likely if there are **System Variables that do not exist** in an old version.)

Note that even "Adjust Game.exe Version" in Basic Game Settings is used to switch the version, this variable always has the latest version of the Game.exe.

116: [Read Only] Screen Size X [Implemented in Ver. 2.20]

A read-only variable. Holds the screen width specified in Basic Game Settings.

117: [Read Only] Screen Size Y [Implemented in Ver. 2.20]

A read-only variable. Holds the screen height specified in Basic Game Settings.

118: [Read Only] Tile Size [Implemented in Ver. 2.20]

A read-only variable. Holds the tile size specified in Basic Game Settings.

120: [Read Only] Reset Log 0=None/1=F12/2=Command [Implemented in Ver. 2.20]

Using F12 to reset after booting sets this variable to 1.

"Go to Title" via Event Command sets it to 2. Possible values are 0~2.

[System String List]

1: [Read Only] Confirm [Keyboard]

2: [Read Only] Cancel [Keyboard]

3: [Read Only] Sub Key [Keyboard]

4: [Read Only] Confirm [Pad]

5: [Read Only] Cancel [Pad]

6: [Read Only] Sub Key [Pad]

Read-only strings. They hold the names of each key/button currently assigned in key config. (For example, in the default state, the keyboard Confirm is Z, so #1 holds "Z". For gamepads, #4 holds "Pad 1", etc.)

10: Hero Character Chip

11~15: Ally 1~5 Character Chip

Specifies the filename of each character's graphics file.

You can change character graphics by assigning filenames to these.

20: Proxy Address [Example: aa.net]

21: Proxy Port Number [Example: 80]

Used with the Download function. Input the proxy address/port number (such as 8080) to use as strings.

By default, Internet Explorer's proxy settings are used.

25: [Read Only] Message Text

A read-only variable. Holds the last string displayed with Show Message.

It can be used to save a log of all messages.

26: Message Add Start

27: Message Add End

Specifies strings to append to the start and end of Show Message text.

For example, if you put "AAA" in #26 and Show Message "Hello!", the message actually displayed will be "AAAHello!"

Uses include having a parallel event that sets #26 to ">" (instantly display message) and #27 to "^" (automatically close message) when the Sub Key is held, giving the Sub Key a message-skip function.

30/31: Background/Fog Graphic

These specify the current background/fog image filenames. You can change the background/fog graphics by assigning filenames here. If you change one of these by assigning a filename, System Variable #61/66 [Fog/Background Number (Sys13)] will become -2.

You can also set with parameters by using multiple lines in the format "Filename (linebreak) X Speed (linebreak) Y Speed (linebreak) Link to Map (1: ON)".

33/34: [Read Only] Current Background Music/Background Sound

A read-only variable. Contains the filename of the currently-playing background music/sound.

40~49: [Read Only] Choice 1~10

A read-only variable. Contains the last-displayed Choice options.

52: [Read Only] Boot Data Folder

A read-only variable. Holds the name of the data folder (i.e. "Data").

As a hidden function, you can run the game with a command like "Game.exe -b Data2", and it will boot with the folder Data2 instead of the standard Data folder.

This folder variable is for when you are doing things with that functionality.

[Special Characters Reference]

A list of special text codes you can use in events and string pictures.

Note that all special codes are case-sensitive, meaning whether you use lowercase or uppercase matters.

[Insert Number or String]

\v[XXX]: Replaced by the value of the XXXth Normal Variable.

\v?[XXX] Replaced by the XXXth variable in Spare Variable set ?. (Example: `\v1[30]`)

\s[XXX]: Replaced by the XXXth String Variable.

\self[X] (*X from 0 to 9): Replaced by this Map Event's SelfX variable.

\cself[XX] (*XX from 0 to 99): Replaced by this Common Event's SelfX.

(0~4, 10-99 are number variables, 5-9 are strings)

\udb[A:B:C] (*A~C are numbers): Replaced by User DB Type A, Data B, Field C.

\cdb[A:B:C] (*A~C are numbers): Replaced by Variable DB Type A, Data B, Field C.

\sdb[A:B:C] (*A~C are numbers): Replaced by System DB Type A, Data B, Field C.

\sys[XXX]: Replaced by the XXXth System Variable.

\sysS[XXX]: Replaced by the XXXth System String.

(Added 01/13/2009)

* There is a **priority** order to special insertion codes, and if you follow it, you can nest them within each other. The priority order is:

Normal Variable \v < Spare Variable \v? < Self Variable \self < Common Self \cself < System Variable \sys < User DB \udb < Variable DB \cdb < System DB \sdb < String Variable \s < System String \sysS

You can put special codes of a lower rank inside of the []s of a special code.

For instance, `\cself[\v[12]]` (`\v < \cself`) will properly call the Common Self of the number given by `\v[12]`.

However, reversing it to `\v[\cself[2]]` does not work, because `\cself` has higher priority than `\v`.

* Due to this priority ranking, the DB-reading codes can include variable-reading codes like so:

`"\udb[3:\v[4]:\self[1]]"`

However, for cases like `\udb[\cdb[1:1:1]:2:3]` (`*\udb < \cdb`, yet `cdb` is on the inside), **where a higher-priority DB-reading code is put inside a DB-reading code, things will behave in an unstable manner.**

[Change Text Properties]

\f[XX]: Changes font size for following text.

\m[XX]: Put at start of line to set maximum font size for line. Use when vertical text placement is getting messy.

\c[XX]: Changes font color. Colors are from System DB Type 12.

\E: Puts edges on following text (E for "edge").

\N: Removes edges from following text (N for "normal"). (*This is the default state.)

\-[XX]: Shortens spacing of following text by XX pixels. If you use a negative value for XX, text spaces out more.

\font[X]: Changes the font to sub-font X. `\font[0]` reverts to default font.

\A+ ... Assigns antialiasing to the following text. This makes text look smoother.

\A- ... Disables antialiasing for the following text.

[Text Display Control]

**\ ** ... Replaced by a normal `\`.

\! ... Pauses message until button press.

\. ... Waits 0.25 seconds.

\^ ... Forces end of message display without input.

\> ... Makes rest of line instantly appear.

\< ... Stops instant display caused by `\>`.

\mx[??] (*?? is a number) ... Shifts position of following text by ?? pixels in the X direction.
Cleared on next line.

\my[??] (*?? is a number) ... Shifts position of following text by ?? pixels in the Y direction.

\ax[??] (*?? is a number) ... Forces internal anchor of following text to X position of ?? pixels.
Cleared on next line.

\ay[??](*?? is a number)...Forces internal anchor of following text to Y position of ?? pixels. If you want the text to overlap, inputting "■\ax[0]\ay[0]◆" will make the two symbols draw in the same place.

\sp[??] (*?? is a number) ... Changes typing speed of following text to ?? characters per second.

\space[??] (*?? is a number) ... Changes linebreak height to ?? pixels.

\r[Base String,Ruby Text] ... Puts "**Ruby Text**" on top of "**Base String**." Ruby text uses the color defined in System DB Type 12 (Text Colors), Data #13.

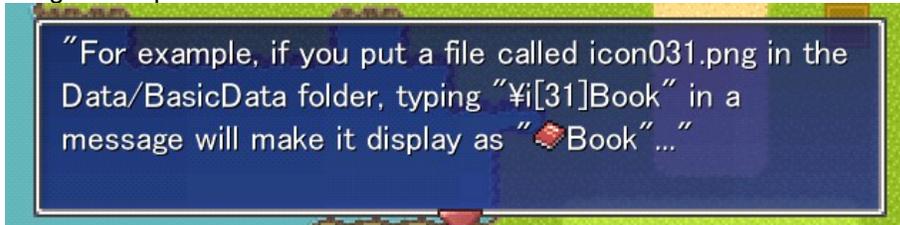
\i[num] (*From 0 to 999) ... Shows "icon<num>.png" icon in message, from the **BasicData folder in Data**.

Can be used to put icons in front of item names, or include special symbols in text. (Added in Ver. 1.03)

* Always make the filenames use three digits after "icon", such as "icon031.png".

* "icon" files will not display unless in the BasicData folder.

<Usage Example>



Here ↑

\img[filename] (*Path relative to Data) ... Shows the image "<filename>" in the Data folder. Example: "**\img[Picture/Test1.png]**" loads Data\Picture\Test1.png. Behavior is the same as icons **\i []** described above, but it may be slightly more intensive.

* If the file is not found, an error message will be shown.

<L> ... Left-align following text. (Default)

<C> ... Center-align following text.

<R> ... Right-align following text.

[Special Codes Usable in Database Field Names]

Below are special codes that can be used in the Field names defined in Database "[Type Settings](#)." They vary somewhat from the codes used in events and string pictures, so be careful.

Type [

Type Name Data ID Sp

Field Definition (Example: Character Name, HP, C

Page 1 Page 2 Page 3 Page 4

Field 0 Default

Number String - Special

Field 1 Default

N/A Number String - Special

← These can be used in the boxes marked by the arrows.

[Insert Number or String]

\udb[A:B:C] (*A~C are numbers): Replaced by User DB Type A, Data B, Field C.

\cdb[A:B:C] (*A~C are numbers): Replaced by Variable DB Type A, Data B, Field C.

\sdb[A:B:C] (*A~C are numbers): Replaced by System DB Type A, Data B, Field C.

\udb[A:B] (*A and B are numbers): Replaced by the Data ID of User DB Type A, Data B.

\cdb[A:B] (*A and B are numbers): Replaced by the Data ID of Variable DB Type A, Data B.

\sdb[A:B] (*A and B are numbers): Replaced by the Data ID of System DB Type A, Data B.

\d[X] (*X is a number): Replaced by the value of the Field number + X.

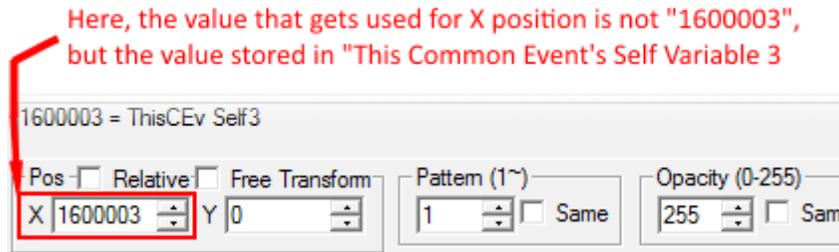
(For example, if you write `\d[4]` in Field 12's input box, it becomes 16.)

* These DB-reading special codes can also have `\d[X]` used inside of them, such as `"\udb[3:\d[0]]"`.

However, cases like `\udb[\cdb[1:1:1]:2:3]` where a DB-reading code is inside another DB-reading code will be unstable, so please refrain from doing this.

[Variable Reference List]

By putting numbers over 1000000 (1 million) in a number field, you can read the values of variables. An example is shown below.



The following values can be read.

1000000+10*Y+X

→ Map Event Y's Self Variable X

1100000+X

→ This Map Event's Self Variable X (Careful not to confuse this with Common Self Variables!)

15000000+100*Y+X

→ Common Event Y's Common Self X (0-4, 10-99 are number variables, 5-9 are strings)

1600000+X

→ This Common Event's Common Self X (Careful not to confuse this with Map Event Self Variables!)

2000000+X

→ Normal Variable #X

2000000+100000*Y+X

→ Spare Variable Set Y's #X (Example: 2100003 = Spare Variables 1 #3)

3000000+X

→ String Variable #X

8000000+X

→ Returns a random value from 0 to X.

9000000+X

→ System Variable #X

9100000+10*Y+X

→ Get or Set Event Y's Position (*1)

(X = 0: Map X Position (Normal [1 tile = 1 unit])

2: Map X Position (Precise [0.5 tiles = 1 unit])

4: Event Height (Pixels)

6: Event Direction (1-9)

1: Map Y Position (Normal)

3: Map Y Position (Precise)

5: Event Shadow Number

9: Event Character Graphic [String]

9180000+10*Y+X

→ Get or Set Hero/Member Position (X the same as (*1), Y is Member number)

9190000+X

→ Get or Set This Event's Position (X is the same as (*1))

* If you set a position using these reference numbers, the character will move there at its defined move speed. By changing position and event facing direction in tandem, **you can move characters and change their direction with only "Set Variable."**

9900000+X

→ System String #X

100000000 (10AABBBBCC)

→ User DB Type A/Data B/Field C

110000000 (11AABBBBCC)

→ Variable DB Type A/Data B/Field C

130000000 (13AABBBBCC)

→ System DB Type A/Data B/Field C

* -999999 to +999999 (999,999) are treated as standard numbers.

[Helpful Tools]

WOLF RPG Editor can use "BMP," "JPG," and "**(transparent) PNG**" formats for images, and "MIDI," "WAV," and "**OGG**" formats for audio. These are some helpful tools for creating these resource files.

- [RPG Maker Chips --> WOLF Chips Conversion Tool tkool2WOLF](#)
- [Transparent PNG Creation Tool / OGG File Creation Tool](#)
- [Character Graphics Synthesizer Tool](#)

[RPG Maker Chips ⇒ WOLF Chips Conversion Tool tkool2WOLF]

A tool that converts RPG Maker 2000, 2003, XP, or VX-format **Map Chips** and **Character Chips** to a **WOLF RPG Editor-supported format**.

You can **split up RPG Maker 2000, 2003, or VX face graphic resources**, **set transparency on images**, or **combine map chips together**; it is extremely multi-functional.

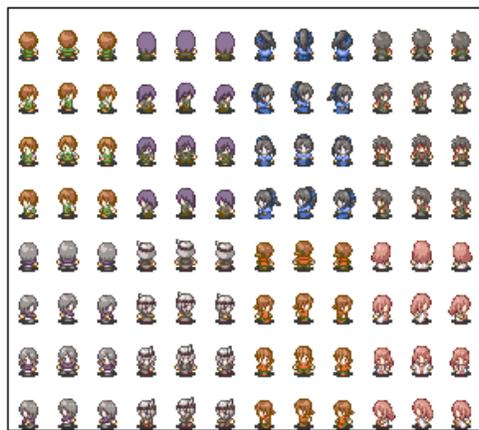
I am deeply grateful to Rikka for creating such a wonderful tool.

- "tkool2WOLF (Japanese)"

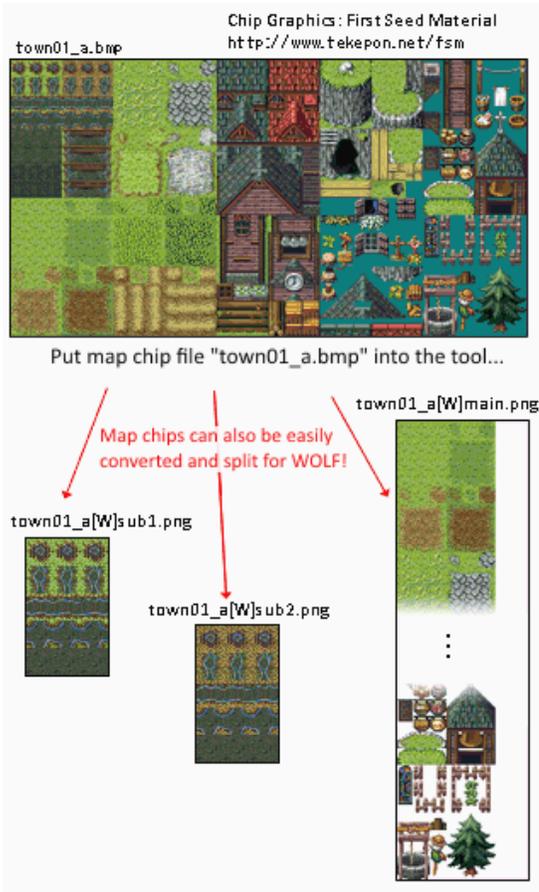
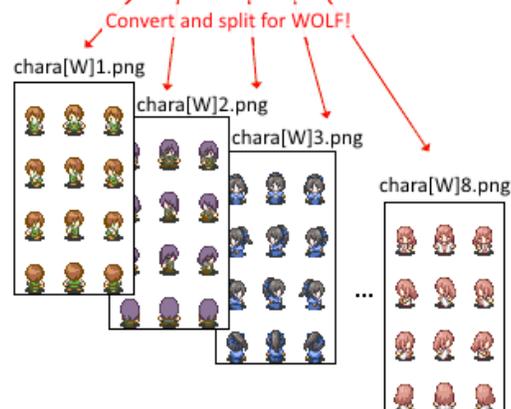
<http://www.silversecond.com/WolfRPGEditor/Data/Rikka/> (Distribution Page)

[Usage]

After starting, just drag and drop the RPG Maker chips you want to convert. Refer to the images below for examples of the basic functions.



Put character chips file "chara.png" into the tool...



[Transparent PNG Creation Tool / OGG File Creation Tool]

These tools let you easily create **transparent PNGs** and **OGGs**, even if you may not be too familiar with them.

[Graphics-Related]

- "**BTJ32 (Japanese)**" Easily create transparent PNG files from BMPs or PNGs.
<http://www.vector.co.jp/soft/win95/art/se074357.html> (Vector)

[Usage]

1. After starting, drag a **BMP** or **PNG** file onto the window (or use "**Open File**").
2. Click the "**P**" icon on top, then check the "**Transparent Background**" box that appears.
3. Click the "**Transparent Color**" button, then "**Select From Image.**"
4. The current image will be shown in the window that appears.
Right-Click the color you want to make transparent and choose "**OK.**"
5. "**Save As**"... And now, you have a transparent PNG.

[Audio-Related]

- "**Rip! AudiCO (Japanese)**" MP3 or WAV format → OGG format conversion tool.
<http://www.vector.co.jp/soft/win95/art/se202196.html> (Vector)
→ When you start, a "first-time wizard" will come up.
You can just drop MP3 or WAV files in the "**OGG Conversion**" box.
- "**KbMedia Player (Japanese)**" A player that can play back OGG-format audio.
<http://www.vector.co.jp/soft/win95/art/se175675.html> (Vector)

* Programs listed as "Windows 95/98 only" will also run on 2000 and XP.

[Character Graphics Synthesizer Tool]

A tool that lets you create a character graphics file by combining parts.

- "Graphic Synthesizer (Japanese)"

Included in the latest [WOLF RPG Editor Full Package](#).

[Usage]

After starting, select "Image Set," and choose the parts you want.
Once done, click the "Save File" button and save your image.

