Close Combat: The Bloody First Modding Guide

Getting Started

Single Battles

You can of course create custom maps and single battles. All the map, battle, script, and other files for these should be saved in

`Documents/My Games/CloseCombatTheBloodyFirst/Maps`

Note that in the original game release there was a

`Documents/My Games/CloseCombatTheBloodyFirst/Scenarios`

folder. This is no longer used and will be deleted by the game if empty.

Mods and Campaigns

To create a mod you need to first go into the MODS folder in

`Documents/My Games/CloseCombatTheBloodyFirst`

(this folder is referred to as your local folder in the documentation). Create a new folder inside MODS for your new mod.

The folder layout inside the mod needs to match that of the main game data. Some key folders are described below. You can, of course, inspect the existing game data to see examples of file layout. You cannot override any files in the `CORE`, `SYSTEM`, or `CONFIG` folders in a mod.

`CAMPAIGNS/MAIN/`

This is the folder where any campaign needs to be created. This is where any custom text files should be placed. They should be named `TEXT1.TXT` and onward, and must be Windows 16-bit unicode files.

`CAMPAIGNS/MAIN/DATA`

Can hold any custom CampaignParams.txt file.

`CAMPAIGNS/MAIN/DATA/BATTLES/`

Needs to contain all the scenario descriptions, scenario scripts, and display images for the battles in the campaign.

`CAMPAIGNS/MAIN/DATA/CAMPAIGNS/`

Contains the campaign description files, and the backdrop images for them. You always need a Grand Campaign.txt file and backdrop. This file defines any other sub-campaigns etc.

`CAMPAIGNS/MAIN/DATA/MAPS/`

Contains the actual map files, custom map textures and normal maps, lighting files, and CLUT maps for the display of the maps in game.

`CAMPAIGNS/MAIN/DATA/OPERATIONS/`

Contains the definition files for any operations in a similar way as for campaigns.

`DATA/BATTLE/MODELS/`

Place any custom 3D models for units or terrain here, along with their textures etc. You can also override textures for models which exist in the main game data.

`CAMPAIGNS/MAIN/DATA/SOUNDS`
You can place sound files to override existing sfx into this folder. WAV and MP3 files are accepted. If you wish to alter the actual sound definitions you would include your own versions of the `SFX0.TXT` etc files and place them in `CAMPAIGNS/MAIN/DATA`

Custom Map Textures
Any custom textures for a map should be in the same folder as the map file. Most file formats are loadable, but TGA, BMP, and DDS are explicitly supported. Any normal map should have the same extension as the custom texture and the name should have a _NORMALMAP postfix. Note that DDS files will be preferred when loading and you do not need to alter the map file if you re-encode a file to be a DDS file as the code will automatically attempt to find a .DDS version of any file before loading the explicit extension.

Scripting

You can script custom behaviors for a battle. You need to create a BSF file (this is a normal text file) - note that the `BOOTCAMP.TXT` map file has a corresponding `BOOTCAMP.BSF` script file which is denoted in the battle file with the SCRIPT entry.

For full details on the scripting language please see the Archon wiki
http://archonwiki.slitherine.com/index.php/Scripting

The script callbacks used by the game are:

```
// Called when the battle is first loaded but before units have been created.
FUNCTION OnStartup()
{
}

// called after default deployment to allow for changes to unit positions
FUNCTION OnDeploy()
{
}

// called each model tick, but only after the battle has started (Begin has
been pressed)
FUNCTION Tick()
{
}

// Called after the battle is complete and all objects have been destroyed.
FUNCTION OnShutdown()
{
}
```

The commands which can be used can be found in the AUTODOCS folder in your local folder. This is your always up-to-date documentation for the inbuilt commands for the game.

Adding Unit and Terrain Models

You can add new unit and terrain models to the data set using the import tool. You need to construct you model bearing mind a few caveats before you export it from your tool of choice. All models you wish to import must be in FBX format. You will of course need to set the appropriate data for new models in your custom terrain or unit file.

Import Tool

Information on the tool can be found on the Archon wiki:
http://archonwiki.slitherine.com/index.php/Rendering_and_Shaders#Converter_Tool

Model Layout

You should ensure that you reset the transforms on all model components before you animate or export the model.

<<include link or zip for MAX reset script>>

Terrain Object Locator Tags:

DETAIL_<0-n group number>_<0-n index in group> - Detail meshes allow you to define groups of minor details on a terrain object and the game will automatically randomize their appearance so that only one member of each detail group is visible.

#= - A tag starting with this value indicates a pick mesh, which is used to simplify hit detection on complex object. For a complex object like a tree you can create a simple mesh encapsulating all branches and foliage and tag it with a name beginning with "#=". This can greatly improve game performance. NOTE: The pick mesh must be above the more complex mesh in the object hierarchy so it has priority in hit testing.

#=trunk - If defined for a tree object, this indicates the mesh that represents the tree's trunk. Trunk and foliage are defined using different terrain data to allow for different behavior. NOTE: The #=trunk tag must be a parent tag of any other #= pick mesh tags for the tree so it has priority in hit testing.

ANCHOR - If placed on a wall object, indicates the point where the wall should anchor to the ground and rotate around.

ROOF - Used on building objects to indicate the mesh that should be hidden when soldiers are inside a building.

DOOR - Used on building objects to indicate an exterior doorway. Used for path finding and to position soldiers to fire out of the building. The locator associated with this tag should be oriented so its Z-axis points outward from the door.

WINDOW - Used on building objects to indicate an exterior window. Used for path finding and to position soldiers to fire out of the building. The locator associated with this tag should be oriented so its Z-axis points outward from the window.

OUTSIDE - Used on buildings to indicate an area within the rectangular bounding box of the building model that is not actually inside the building.

Vehicle Locator Tags:

HULL - indicates the mesh of the vehicle body. Used to calculate vehicle line of sight, targeting, and hit detection.

TURRET0 - indicates the mesh of the vehicle's turret, if it has one. This mesh will be Y-rotated as the turret rotates. Used to calculate line of sight, targeting, and hit detection.

WEAPON<0-n> - indicates a vehicle weapon. If associated with a mesh it will be Y-rotated to show traversing the weapon. If the weapon should rotate around a point other than its own center create a rotator tag on a parent object of the weapon mesh.

ROTATOR - Used to cause a weapon mesh to rotate around a point other than the center of the weapon mesh itself. Useful for ring-mount machine guns.

FIRE<0-n> - indicates a weapon firing point. Used to position weapon fire FX and spawn projectiles fired from the weapon. Fire tags should be on children of the corresponding Weapon -- i.e. Fire0 should be a child of Weapon0.

CREW<0-n> - indicates the location of a crew member in or on the vehicle. Crew models will only be shown if the crewman is manning an 'always exposed' position (see vehicle data file for details.)

For towed guns these correspond to the crew job as follows:

CREW0 - Ammo bearer
CREW1 - Gunner
CREW2 - Commander
CREW3 - Loader
CREW4 - Assistant Gunner
CREW5 - Assistant Loader

For other vehicles they are:

CREW0 - Driver
CREW1 - Gunner
CREW2 - Commander

CREW3 - Loader
CREW4 - Radio Man / Assistant Driver / Bow Gunner
CREW5 - Assistant/Secondary Gunner
CREW6 - Assistant/Secondary Loader
CREW7 - Ammo bearer

HATCH<0-n> - indicates a vehicle hatch mesh. These are hidden if the vehicle has been abandoned or knocked out.

Vehicle wrecks use the same models as the base vehicle but you can supply an alternative wrecked texture by using the same base texture name (found in uniforms.txt) and adding "_WreckText" to the name before the file extension.

Weapon Locator Tags:

FIRE - indicates a weapon firing point. Used to position weapon fire FX and spawn projectiles fired from the weapon.

WEAPON_HANDLE - indicates position where soldier's right hand should be on the weapon.

WEAPON_STOCK - indicates position where the soldier's left hand should be on the weapon when in a two-handed weapon animation.

For heavy weapons (machine guns and mortars) you can include:

CREW1 - Indicates the position where a heavy weapon gunner should be positioned relative to the weapon.

CREW3 - Indicates the position where a heavy weapon loader should be positioned relative to the weapon.

CREW7 - Indicates the position where a heavy weapon ammo bearer should be positioned relative to the weapon.

The Data Workbook

The moddable data files in the `DATA/BASE/` folder are documented in the Close Combat The Bloody First Workbook spreadsheet. These data files allow you to modify, add, or remove units, vehicles, soldiers, weapons, ammunition, and terrain types.

Each worksheet lists the data for that file in tabular format. To export your changes from the worksheet into the text data file, copy the text from the non-bold section of the worksheet table and paste it into the corresponding text data file and save the text file into the matching folder for your mod. Make sure your new data completely replaces any existing text in the text data file.

Below the tabular data in each worksheet you will find the corresponding data file name, notes about the worksheet, and a description of each data field.

The Campaign System

The Close Combat campaign system data files start with a 'grand campaign' data file that lists the campaigns in your mod. You need a grand campaign file even if your mod only has one campaign.

The Grand Campaign file is always a text file named "Grand Campaign.txt" in the `CAMPAIGNS/MAIN/DATA/CAMPAIGNS/` folder of your mod. The Grand Campaign data file will list one or more campaigns that the player can play individually or in a sequence.

In addition to the Grand Campaign file you need to define at least one Campaign. Campaign data files must be text files with a ".txt" file extension but can otherwise be named as you wish. Each campaign must define at least one Operation that is part of that campaign.

Operation data files reside in the `CAMPAIGNS/MAIN/DATA/OPERATIONS/` folder of your mod. Operation data files must be text files with a ".txt" file extension but can otherwise be named as you wish. Each Operation will define one or more battles that make up that Operation.

Battle data files reside in the `CAMPAIGNS/MAIN/DATA/BATTLES/` folder of your mod. Battle data files must be text files with a ".txt" file extension but can otherwise be named as you wish.

All of these data files will be collectively referred to as 'scenarios' when the information applies to more than one level of campaign data file.

Campaign Parameters

Campaign level game play can be modified in the CampaignParams.txt file found in the `CAMPAIGNS/MAIN/DATA/` folder. The fields are:

Dawn/Day/Dusk/NightHour: The hour of the day when each type of light level begins.

NightBattleHour: The hour of the day when night battles begin.

BattleTime: The number of campaign hours that pass whenever you fight a tactical battle.

Allied/AxisHastyAttackTime: The number of campaign hours that pass before a Hasty Attack.

Allied/AxisPreparedAttackTime: The number of campaign hours that pass before a Prepared Attack.

Hours of the day use a 24-hour clock. Hour values must be whole numbers -- minutes/partial hours are not supported.

Allied/AxisWoundedRecoveryPrecent: The percent chance a wounded soldier will recover and return to duty. Checked every morning at dawn. Wounded soldiers automatically return to duty at the start of a new Operation.

Allied/AxisIncapRecoveryPercent: The percent chance an incapacitated soldier will return to duty. Checked at the start of every new Operation.

Allied/AxisDamagedRepairPercent: The percent chance a damaged vehicle will be repaired and returned to duty. Checked before every prepared attack and every morning at dawn.

Allied/AxisKORepairPercent: The percent chance a knocked out vehicle will be repaired and returned to duty. Checked every morning at dawn.

Allied/AxisAssumedCasualtyPercent: The percent of main force strength that higher headquarters expects to become casualties during a battle. Used to generate your initial replacement pool and to evaluate your actual losses after each battle.

Allied/AxisSoldierReplacementPercent: The percentage chance a casualty will be replaced by new a new soldier from the army replacement system. This is modified by the game difficulty setting (See difficulty values, below.)

Allied/AxisVehicleReplacementPrecent: The percentage chance a lost vehicle will be replaced by a new vehicle from the army replacement system. This is modified by the game difficulty setting (See difficulty values, below.)

Percent values should be whole numbers from 0-100.

Allied/AxisHastyFatigue: Force-level fatigue change when you make a Hasty Attack.

Allied/AxisPreparedFatigue: Force-level fatigue change when you make a Prepared Attack.

Allied/AxisOvernightFatigue: Force-level fatigue change when you make a Dawn Attack.

Force level fatigue ranges from 0-100 with lower values making the soldiers increasingly tired.

Allied/AxisHastySupply: Force-level supply change when you make a Hasty Attack.

Allied/AxisPreparedSupply: Force-level supply change when you make a Prepared Attack.

Allied/AxisOvernightSupply: Force-level supply change when you make a Dawn Attack.

Force level supply ranges from 0-100 with lower values reducing the amount of ammunition your soldiers will have at the start of a battle.

Allied/AxisHastyMorale: Force-level morale change when you make a Hasty Attack.

Allied/AxisPreparedMorale: Force-level morale change when you make a Prepared Attack.

Allied/AxisOvernightMorale: Force-level morale change when you make a Dawn Attack.

Force level morale ranges from 0-100 with lower values reducing the force morale at the start of a battle.

Allied/AxisBattleFatigue: The force-level fatigue change applied after each tactical battle.

Allied/AxisBattleSupply: The force-level supply change applied after each tactical battle.

Allied/AxisBattleMorale: The force-level morale change applied after each tactical battle.

Allied/Axis Reputation Values: These represent the relative effect on the player's reputation as a commander for each type of battle result. Commander reputation is on a scale of 0-100 and begins at 50 (average.) High or low reputation influences higher command's expectations and post-battle evaluation as well as other bonuses and penalties as described in the game manual.

Difficulty Modifiers: Player difficulty settings can modify initial force supply and morale levels as well as initial force strength. Selecting a high difficulty will reduce these values while selecting a low difficulty will increase them. For example, using the stock values a difficulty setting of 3 (Veteran) will reduce initial force-level supply and morale by 10%, start the player's force with roughly 10% casualties, and reduce the rate of replacements by 10%.

The difficulty setting also determines what percentage of the support units from the scenario data file (see Battle Data File below) will appear in the players force pool on the Force Selection screen. At difficulty 2 (Line) you will receive 50% of the possible support units. At the lowest difficulty (Recruit) you will receive 100% of them.

Grand Campaign and Campaign File Format

The Grand Campaign and Campaign data files are text files where each line of text is a separate data field. The fields are listed below in the order they must appear.

Scenario Type: 0 for a Grand Campaign data file or 1 for a Campaign data file.

Condition: Should normally be 0 for 'always' but can have other values. See 'Conditions' below.

Title: Either a string tag or plain ASCII text. Maximum length for plain text is 260 characters.

Summary: String tag or plain text. A general description of the campaign that will appear on the Command Screen. Maximum length for plain text is 4096 characters.

Allied Briefing: String tag or plain text. Shown to the Allied player at the start of the campaign. Maximum length for plain text is 4096 characters.

Axis Briefing: String tag or plain text. Shown to the Axis player at the start of the campaign. Maximum length for plain text is 4096 characters.

Allied Main Force: A single numeric value that identifies the Allied main force for this campaign. This value is the index of a unit structure from the scnunits.txt data file in the `DATA/BASE/UNITS/` folder.

Axis Main Force: A single numeric value that identifies the Axis main force for this campaign. This value is the index of a unit structure from the scnunits.txt data file in the `DATA/BASE/UNITS/` folder.

Allied Upgrades: One or more numeric values separated by spaces. Identifies the upgrades that are always available to the Allied player in this campaign. This value is the index of an upgrade from the upgrades.txt data file in the `DATA/BASE/` folder. Use -1 for no upgrades.

Axis Upgrades: One or more numeric values separated by spaces. Identifies the upgrades that are always available to the Axis player in this campaign. This value is the index of an upgrade from the upgrades.txt data file in the `DATA/BASE/` folder. Use -1 for no upgrades.

Campaign/Operation Count: A single numeric value indicating the number of campaigns in the Grand Campaign or Operations in a Campaign. Must be at least 1.

Campaign/Operation Name(s): The campaign/operation count is followed by a one or more campaign name lines (in the Grand Campaign file) or operation name lines (in a Campaign file.) Use a separate line for the name. Each line should be the text file name, including the .txt extension, listed in the order you want them to play.

Possible Next Campaign Count: Single numeric value indicating the possible number of campaigns to follow this one during play. Must be 0 for the Grand Campaign.

Possible Next Campaign Name(s): The possible next campaign count is followed by zero or more campaign name lines, one for each possible next campaign to follow this one. Each line should be the text file name, including the .txt extension, of a campaign, listed in the order you want the game to check them. The first possible next campaign with a valid condition will be the one the player plays next at the end of this campaign. The Grand Campaign can never have a ‘next’ campaign.

Preview Image File Name: The name of the texture file to display on the Command Screen when this campaign is selected.

Preview Image Location: Two space-separated numeric values giving the screen offset (in pixels on a 1024 x 768 display) to use to position the preview image on the Command Screen. Offsets will be scaled accordingly for other display resolutions.

Operation File Format

Most of the operation data file fields function the same as those in the campaign data file, with the exceptions noted below.

Scenario Type: 2 for an Operation.

Condition: As in Campaign data file.

Title: As in Campaign data file.

Summary: As in Campaign data file.

Allied Briefing: As in Campaign data file.

Axis Briefing: As in Campaign data file.

Start Data/Time: The date and time the Operation begins. The format is YYYY M D H M S. Leading zeros are not necessary.

End Data/Time: The date and time the Operation ends. The format is YYYY M D H M S. Leading zeros are not necessary. The operation automatically ends when the game time advances beyond the end time.

Allied Main Force: As in Campaign data file.

Axis Main Force: As in Campaign data file.

Allied Upgrades: As in Campaign data file.

Axis Upgrades: As in Campaign data file.

Battle Count: A single numeric value indicating the number of Battles in this Operation. Must be at least 1.

Battle Name(s): The battle count is followed by a one or more battle name lines. Use a separate line for each name. Each line should be the text file name, including the .txt extension, listed in the order you want the battles to play. Moving down the battle list is Allied progress while moving up the battle list is Axis progress.

Initial Battle Index: A 0-based index into the list of battle names. This indicates the initial battle to play when the operation first begins.

Possible Next Operation Count: Single numeric value indicating the possible number of operations to follow this one during play.

Possible Next Operation Name(s): The possible next operation count is followed by zero or more operation name lines, one for each possible next operation to follow this one. Each line should be the text file name, including the .txt extension, of an operation, listed in the order you want the game to check them. The first possible next operation with a valid condition will be the one the player plays next at the end of this operation.

Preview Image File Name: The name of the texture file to display on the Command Screen when this operation is selected.

Preview Image Location: As in Campaign data file.

Starting Initiative Side: 0 = Allied Initiative, 1 = Axis Initiative, -1 = None. Determines who has the initiative (and is the attacker) at the start of the operation.

Battle File Format

Most of the battle data file fields function the same as those in the campaign data file, with the exceptions noted below.

Scenario Type: 3 for a Battle.

Condition: As in Campaign data file.

Title: As in Campaign data file.

Summary: As in Campaign data file.

Allied Briefing: As in Campaign data file.

Axis Briefing: As in Campaign data file.

Start Data/Time: The start date and time if played as a one-off battle. Format as above.

Initiative Side: 0 = Allied Initiative, 1 = Axis Initiative, -1 = None. Determines which side is the attacker if played as a one-off battle.

Weather: 0 = Clear, 1 = Partly Cloudy, 2 = Cloudy. The worse the weather the longer it takes air support to arrive.

Ground Conditions: 0 = Dry, 1 = Wet. Wet ground conditions will reduce dust from vehicle movement.

Allied Main Force: As in Campaign data file.

Axis Main Force: As in Campaign data file.

Allied Supporting Units: A space separated list of numeric values identifying the Allied supporting units. This value is the index of a unit structure from the scnunits.txt data file in the `DATA/BASE/UNITS/` folder. Note that only a percentage of these units will be available to the player based on game difficulty setting. Use -1 for no supporting units.

Allied Supporting Units: A space separated list of numeric values identifying the Axis supporting units. This value is the index of a unit structure from the scnunits.txt data file in the `DATA/BASE/UNITS/` folder. Note that only a percentage of these units will be available to the player based on game difficulty setting. Use -1 for no supporting units.

NOTE: You may assign supporting units to named ‘groups’ if desired. To create a named group include a string tag, or a literal string that *does not* begin with a number or the minus sign and does not contain spaces, within the space separated list. All supporting unit indexes following this name will be included in that group until a new group name is encountered. Named unit groups have no effect on game play other than showing the units grouped under the name on the Force Select screen.

Allied Fire Support List: A space separated list of Allied fire support available in this battle, or -1 for none. Each value is the index of a fire support asset from the firesupport.txt data file in the `DATA/BASE/UNITS/` folder. The same value can be listed more than once to allow multiple uses in the same battle.

Axis Fire Support List: As above, but for the Axis side.

Allied Upgrades: As in Campaign data file.

Axis Upgrades: As in Campaign data file.

Map File Name: The name of the map used for this battle, without the file extension.

Map Size: Two numeric values, separated by spaces, giving the expected map width and height in meters. If the map file is missing a flat, open ground map of this size will be created to allow play.

Initial Map Control: A space separated list of numeric values indicating which side controls which part of the map the first time it is played. 0 = Allies, 1 = Axis, 2 = Uncontrolled. Each value represents a 32x32m ‘deployment tile.’ The first value in the list represents the tile in the southwest corner of the map and each successive value in the list proceeds by 32m tile across the map to the east until it hits the map edge, after which it shifts north one 32m tile on the map and continues from west edge to east edge.

Victory Location Count: The number of victory locations on the map. Must be at least 1.

Victory Location List: A number of lines follow the victory location count, TWO lines for each victory location in this battle. The first line is the name of the victory location as a string tag or literal string. The second line must have three integer values, separated by spaces, giving the value of the victory location, its X deployment tile location on the map, and its Z deployment tile location on the map, respectively. Victory location value is the equivalent amount of deployment tiles it is worth to be holding this victory location at the end of the battle.

Allied Exit Direction: A single numeric value indicating the direction toward the Allied rear. Valid values are 0 = North, 1 = Northeast, 2 = East, 3 = Southeast, 4 = South, 5 = Southwest, 6 = West, 7 = Northwest. This is the direction indirect fire support will fire from and toward which soldiers will flee if routed.

Axis Exit Direction: As above, but for the Axis side. In general the Allied and Axis exit directions should be opposite one another.

Preview Image Location: As in Campaign data file. To display a battle preview image you must create a bmp file image with the same name as the map name field above and place it in the `CAMPAIGNS/MAIN/DATA/BATTLES/` folder.

Ability to Dig In: -1 = no one can dig in, 0 = Allies may dig in, 1 = Axis may dig in, 2 = Both may dig in. Used to determine which side(s) start with foxholes and gun pits if played as a one-off battle.

Possible Axis-Direction Battle Count: Single numeric value indicating the possible number of battles to play following this one if the parent operation advances in the Axis direction. May be 0.

Possible Axis-Direction Battle Name(s): The possible axis-direction battle count is followed by zero or more battle name lines, one for each possible axis-direction battle to follow this battle. Each line should be the text file name, *without* the .txt extension, of a battle file, listed in the order you want the game to check them. The first listed battle with a valid condition will be the one the player plays next when the operation advances in the Axis direction.

Possible Allied-Direction Battle Count: Single numeric value indicating the possible number of battles to play following this one if the parent operation advances in the Allied direction. May be 0.

Possible Allied-Direction Battle Name(s): The possible allied-direction battle count is followed by zero or more battle name lines, one for each possible allied-direction battle to follow this battle. Each line should be the text file name, *without* the .txt extension, of a battle file, listed in the order you want the game to check them. The first listed battle with a valid condition will be the one the player plays next when the operation advances in the Allied direction.

Associated Script File: This line is optional. If you have an Archon script file associated with your battle, add the line “SCRIPT <filename without extension>” (without surrounding quotation marks) as the last line in your battle data file. Battle scripts should be placed in the `CAMPAIGNS/MAIN/DATA/BATTLES/` folder.

General Scenario Data File Information

Scenario Conditions: The Scenario Condition field provides a simple mechanism to put branching paths in the flow of battles in an operation or in the flow of operations in a campaign. The Grand Campaign level does not support conditional campaigns.

Campaigns and Operations evaluate their possible next sub-scenarios one at a time, in order. If a scenario has a condition that does not match the current game state it is skipped and the next scenario in the list is checked until a scenario with a valid condition is found or the list is exhausted and the Campaign or Operation is considered complete.

Scenario conditions can include an optional message shown to the player or an either-or choice for the player to choose between two alternatives. The either-or choice allows the player to decide if they want to accept the branch or reject it.

Scenario conditions always start with an integer giving the condition type. Depending on the condition type they may then have one more more integer or string fields, separated by spaces, following the condition type.

Condition Types:
0 = Always

1 = If Chosen (essentially the same as Always)
2 = If Allied Reputation <numeric comparison> <value>
3 = If Allied Cohesion <numeric comparison> <value>
4 = If Current Date <numeric comparison> <YYYY M D H m>
5 = If Last Sub-scenario Result <numeric comparison> <value>
6 = If Last Battle Victory Location Control <VL Control> <VL index in battle file>
7 = If Axis Reputation <numeric comparison> <value>
8 = If Axis Cohesion <numeric comparison> <value>
9 = <Unusued>
10 = Only as an Alternate Choice
11 = Never (used internally by the campaign system but of no use when creating scenarios.)

For a normal scenario you will simply use '0' for the condition field to have it always execute.

For simple comparisons you will need to add two or three additional parameters to define the condition. Comparison types are given below. Reputation and Cohesion are on a 0-100 scale.

You can also check the control of a specific victory location from the last battle fought. The comparison field will specify who needs to control the victory location for the condition to be valid and the value field will give the index of the victory location within the battle file.

For date comparisons the value field is a space separated string of integers giving the Year, Month, Day, Hour, and Minutes to specify the date value. Leading zeros are not necessary.

Comparison Types: 0 = equals, 1 = less than, 2 = greater than, 3 = less than or equal, 4 = greater than or equal.

Value: The numeric value to compare against. Reputation and Cohesion are both a 0-100 integer scale. Victory levels are 1 = Allied Decisive Victory, 2 = Allied Major Victory, 3 = Allied Minor Victory, 4 = Draw, 5 = Axis Minor Victory, 6 = Axis Major Victory, 7 = Axis Decisive Victory.

VL Control: 0 = Allied control, 1 = Axis control, 2 = Uncontrolled.

A condition can also have a choice, warning message, and choice labels associated with it. The choice value is optional but if it is included the rest of the fields must also be included.

Choice: 0 = None, 1 = Two Alternatives.

Choice Side: If there is a two-alternative choice in this condition you must include a field saying which side gets to make the choice (0 = Allies, 1 = Axis.)

Choice Message: <Message String Tag>|<Choice 1 String Tag>|<Choice 2 String Tab>

To present the player with a simple message box use a choice value of '0' followed by a string tag for the message string to display. The message will be shown in a dialog box with an 'Ok' button. This can be used to notify the player of campaign or operational level events they may not otherwise notice.

To present the player with two alternatives use a choice value of '1' and provide three string tags separated by the vertical pipe/bar character, as shown above. The message will be presented to the player in a dialog box with the message text above and each choice string on a separate button below. Keep the choice strings themselves relatively short as they will be truncated to fit on the buttons.

An example from the stock campaign:

The Kasserine Pass Operation has a scenario condition line that reads:

5 3 2 1 0 O_KASSERINE_PASS_CONDITION | O_KASSERINE_PASS_CHOICE0 | O_KASSERINE_PASS_CHOICE1

'5' indicates the Last Sub-scenario Result condition type.

'3' indicates a 'less than or equal' comparison.

'2' is the victory level, in this case a major Allied victory.

'1' indicates two alternative choices. This is followed by the string tags for the message and the two choices.

'0' indicates the choice is for the Allied side/player.

The message string will be shown to both sides/players. The condition choice strings will be shown only to the choosing side.

After the player completes the First Blood operation in the Tunisia campaign that operation is checked to see what the next possible operation(s) might be. Longstop Hill.txt is the data file for this operation and it has 2 possible next scenarios to check in the order 1) Kasserine Pass.txt and 2) Sbiba.txt.

Kasserine Pass is checked first and the condition above is evaluated. If the player's victory level was 3 or higher (Allied minor victory, draw, or defeat) the condition is evaluated as false. If the Allied side scored a major or decisive victory in the First Blood operation the choice portion of the condition is then checked and the player will be given the choice of where to proceed.

If the player chooses the first choice the condition for Kasserine Pass is evaluated as true and it becomes the next operation to play. Kasserine Pass.txt *does not* include Sbiba.txt as a possible next operation, so taking this branch will skip the Sbiba operation completely.

If they choose the second choice the condition is evaluated as false and the next potential operation, Sbiba.txt, is checked. Sbiba has a condition of '0' or always and thus will always be the next operation to play in this case.

Only as An Alternate Choice: Condition Type 10 can be used for battle scenarios to create a permanent fork in an operation. If the two possible next battles in an operation are part of a player choice condition and the second of those has a condition type of 10, the second possibility will have its condition value reset depending on the player's choice. If the player chooses the first choice the second battle (with condition type 10) will have its condition type set to 11, or Never. If the player chooses the second option the second battle will have its condition type set to 0, or Always. Scenarios with a 'never' condition do not

show up in the game UI and can never occur. So an unchosen alternative can be cut off from the operation, will no longer appear on the UI, and never be played after the choice has been made, even though operations can ‘flow’ in both directions.

Scenario Forces: Your main force may change over the course of a game depending on the main force set in the lowest level scenario you are playing. A Battle data file main force will override an Operation file main force setting, and an Operation data file main force setting will override a Campaign level one, etc. If you begin a new scenario where the main force is different than the previous scenario there are two possible outcomes:

1) The old main force can ‘convert’ to the new structure if the scnunits.txt data file values allow for this. In this case your soldiers’ history will be preserved and the soldiers will be assigned to the new force structure according to the conversion data in scnunits.txt file.
2) If the old main force cannot convert to the new structure all history is lost and a completely new unit with new soldiers is created.

Supporting units are cleared and recreated every time you begin a new battle scenario (i.e. you move to a new map.) Supporting unit history is maintained only while you continue to fight on the same map.

Scenario Upgrades: Upgrades from the scenario files are created once when you begin a new game. Upgrades from the highest level scenario stay with you through the entire game. If you start a new Grand Campaign, for example, you will receive the scenario upgrades in the Grand Campaign.txt file and any scenario upgrades in the data files for the Campaigns, Operations, or Battles you play as part of that Grand Campaign will be ignored. If you start a new Operation then you will receive the scenario upgrades from that Operation data file and any scenario upgrades in Battle data files will be ignored.

Force Upgrades: Upgrades associated with your main force unit (as defined in ScnUnits.txt) will change when/if your main force changes over the course of an operation or campaign. Force upgrades associated with units not your main force will not be created by the game.