GAx3

StarCraft II Editor Enchant Mod Develop Manual

GAx3 Current Build: 1.0.0.1248(milestone 1)

Author: 麦德三世(Renee)

Locale: Chinese Simplified (zhCN), Chinese Traditional (zhTW), Engilsh(enUS)

Supported StarCraft II Version: SC2 1.3.x & SC2 1.4.x

Index

Chapter I - What Is GAx3 and How to Use It

What Is GAx3

GAx3 is an extreme powerful mod made by me to improve the official StarCraft II Editor. It also included some of my special map technologies (Like my *StarCraft II Phasing Technology*). With this Mod, user can easily do some things that very hard to do with official editor (or even some things seemingly impossible to do!).

The mod included many *universal* trigger events/conditions/actions, as well as many *universal* data editor entries. I also did some manageable fixes for some official editor bugs. The idea of the mod is to provide more powerful editor to make the users doing maps more easier/better.

The mod considered the requirements of both the GUI Users and the Script Users. All new features provide by GAx3 have GUI Supports, you can directly see them in Trigger Editor & Data Editor. And for users that prefer to use Galaxy Scripts, GAx3 also provided a full API list (see Chapter V).

Though this mod isn't encrypted, nor had it code confused. But if you use this mod to making maps, or use some unique technique from this mod in your maps. It would be better to mention it in your maps.

Does GAx3 need to be installed? How do I use the new features of GAx3?

Just like any StarCraft II Mods, GAx3 doesn't need to be installed. You have two ways to add GAx3 dependency into your map: BN dependency & local dependency.

Note: If you want to publish your map on Battle.net, you have to use BN dependency.

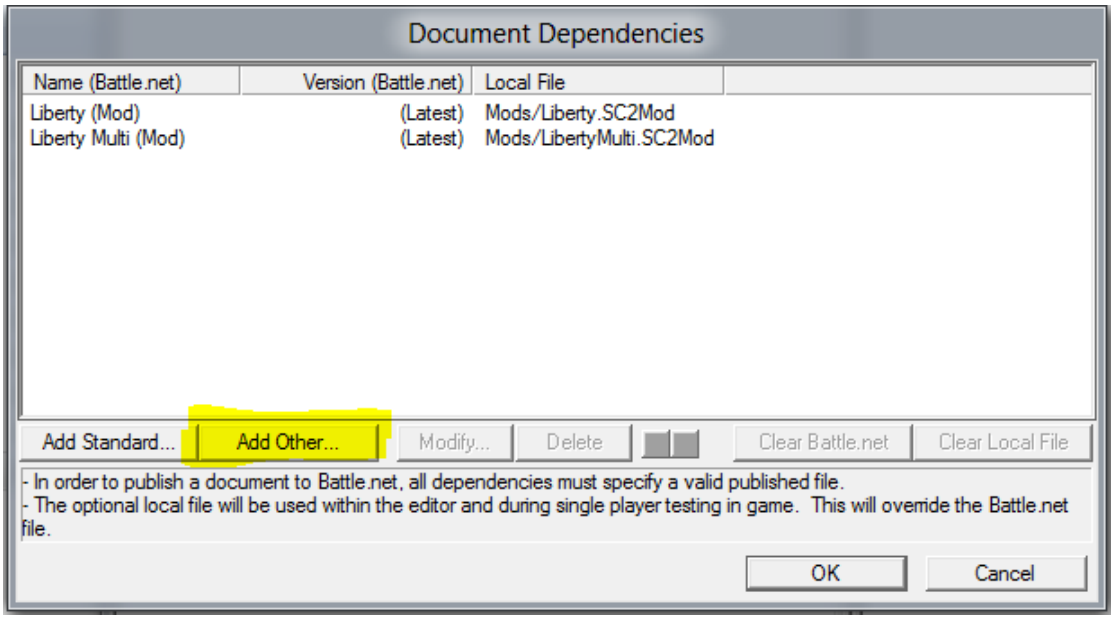
BN dependency:

GAx3 Mod has been published to the Battle.net of North America (US), China Mainland (CN), and Taiwan (TW). If you have any US/CN/TW battle.net account, you can log in any of these BN, and add GAx3 dependency online directly. (If you don’t have one, see the Local dependency paragraph blow)

GAx3 dependency not only can be added when creating new maps, but also can be added into old maps we created before. First, I will show the steps that how to add it when we

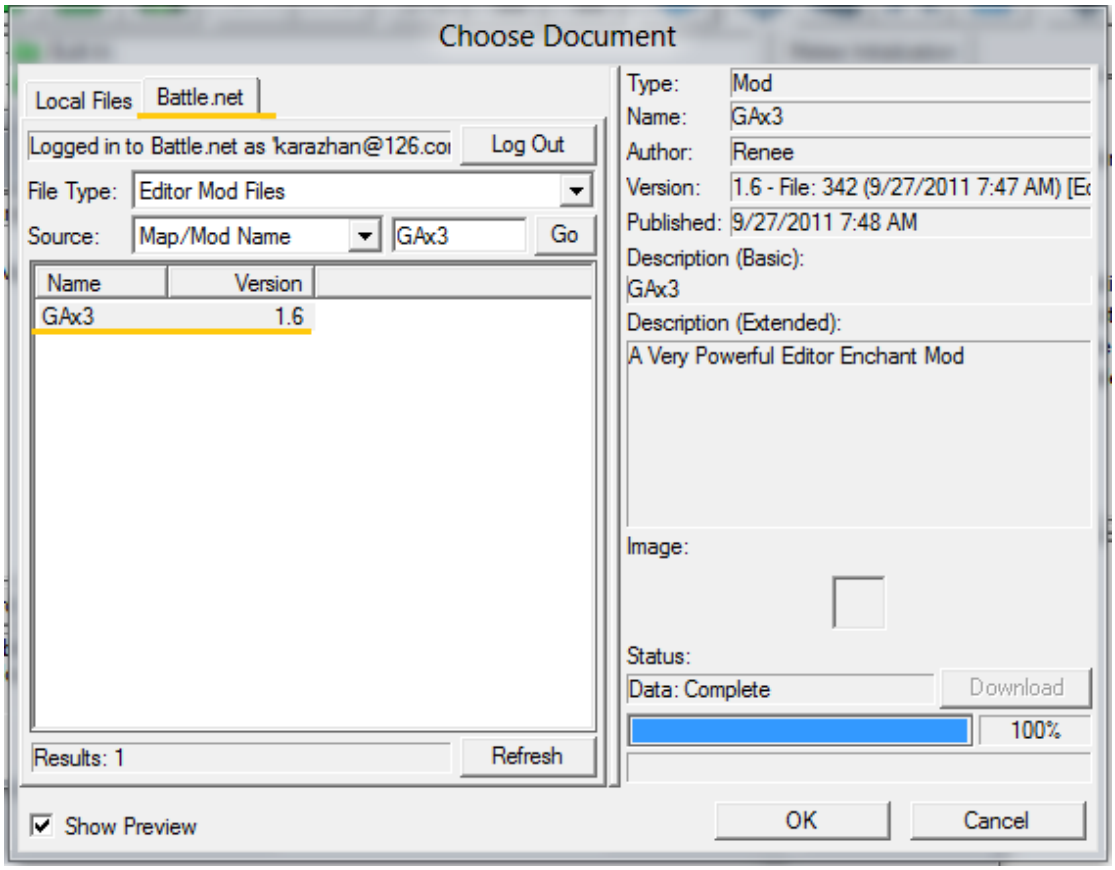
creating new maps:

As we all know, when we create new maps, we can press the "Modify..." button to modify the dependencies of the map. To use new features provided by GAx3, all your need is to press the "Add Other..." button after you've finished your map's dependencies setting:

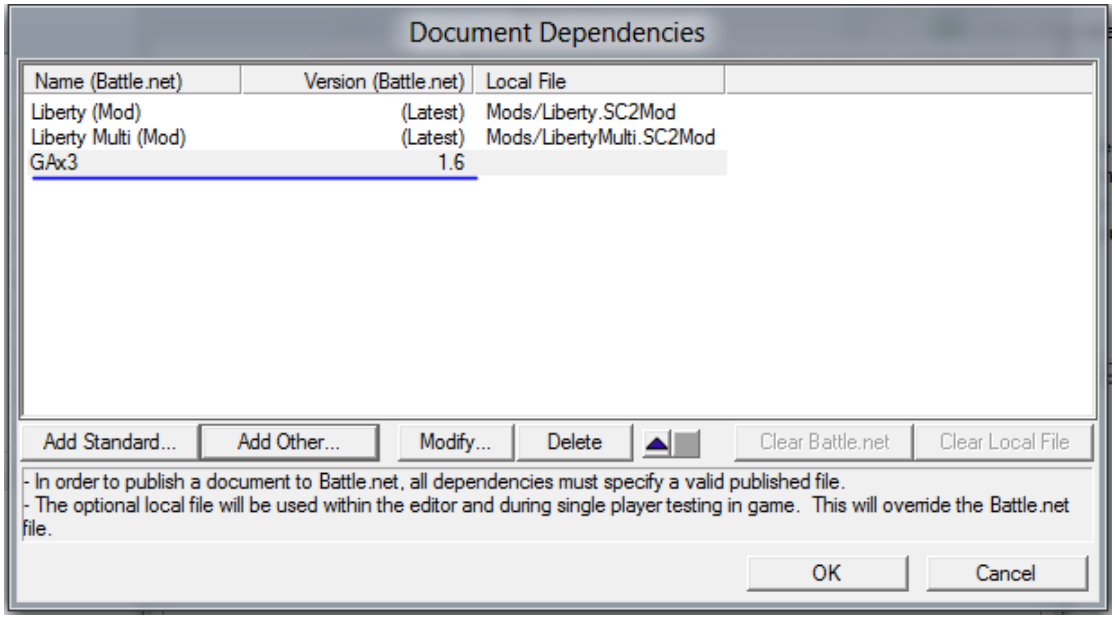


Then select the *Battle.net* sheet in the *Chose Document* window and log in, *File Type* to be "Editor Mod Files", Source to be "Map/Mod Name", Search box to be "GAx3", and press "Go"

When GAx3 showed in the list, you can select it, and press "OK" to add dependency to this mod. (If you haven't downloaded it before, there will be a download progress.)

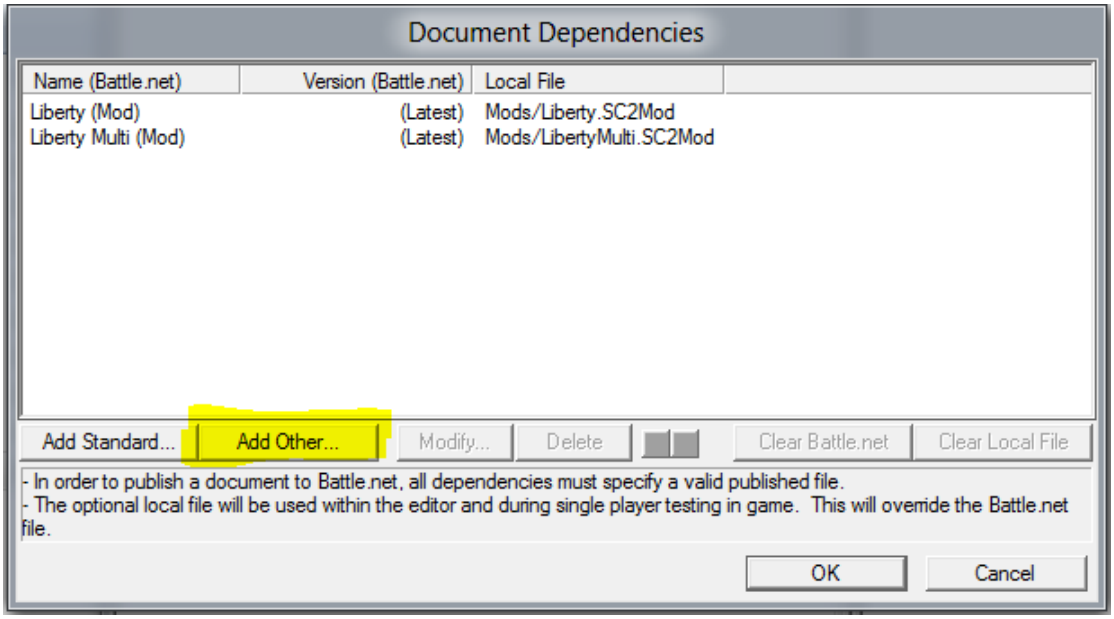


Now GAx3 have been included into your map, and you can already use all the new features of it.



And for the old maps we created before, we could also add GAx3 BN dependency into them:

First, we use the Editor to open our map, and then press the *File -> Dependencies...* menu, the dependencies list of the map would show up. Now we can press the "Add Other..." button, and follow every step above.



Local dependency:

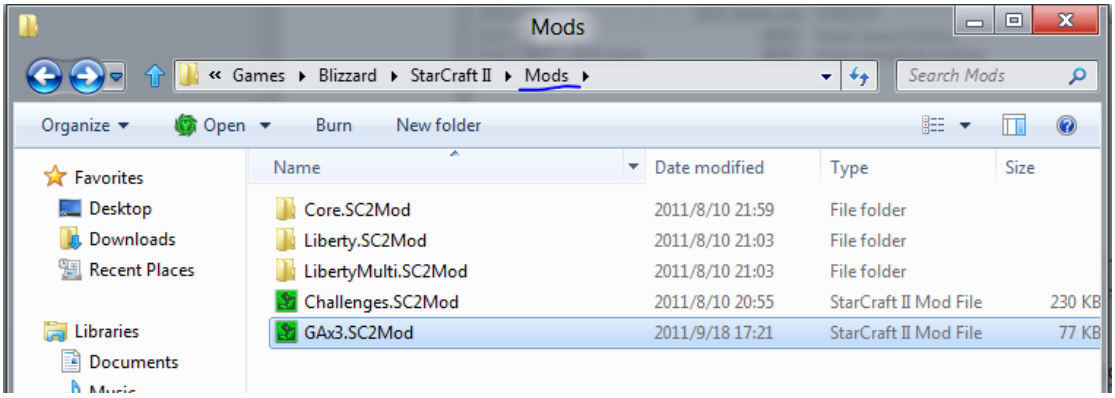
If you don't have any SC2 US/CN/TW account, you can still use GAx3 by directly download. But due to current battle.net architecture, your map cannot be published to the battle.net properly if you use the **local dependency** method, it means If you want to publish your map, you have to use the BN dependency, if anyone volunteer to help me upload this mod into other battle.net (like EU), please contact me.

Generally, you can directly download this mod in the GAx3 forum of Goblin Academy site (Recommended, Simplified Chinese forum):
http://bbs.islga.org/thread-htm-fid-133.html

GAx3 can also be downloaded in the project forum of SC2Mapster (English forum)
http://www.sc2mapster.com/forums/resources/project-workplace/

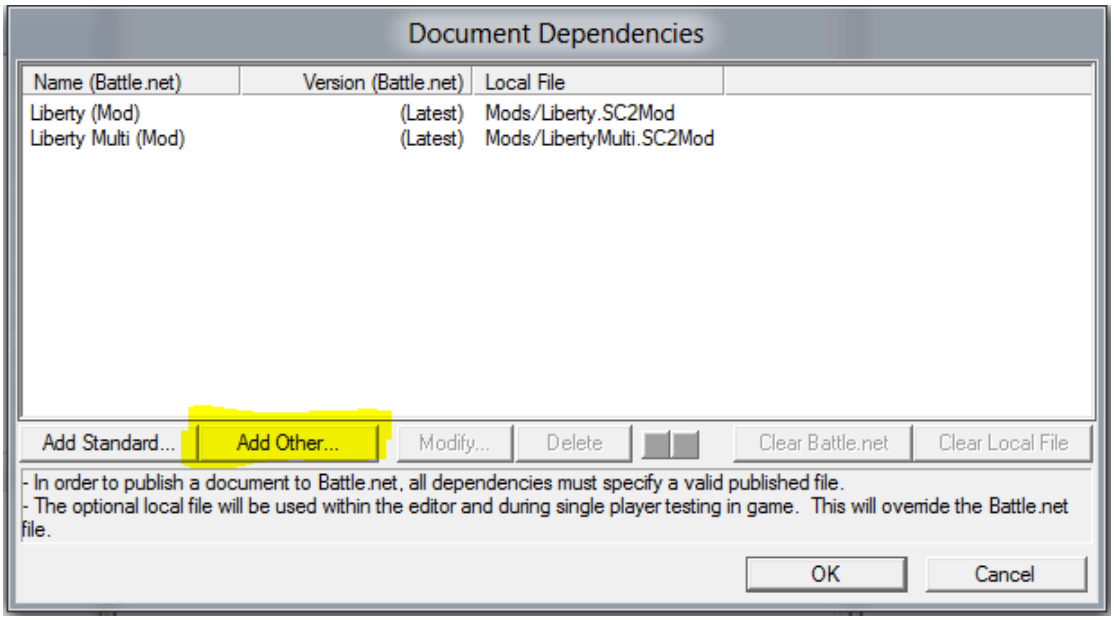
And US battle.net Custom Maps forum:
http://us.battle.net/sc2/en/forum/13437/

GAx3 Mod itself is a file named GAx3.SC2Mod. After you download it, you need to place it under the Mod directory under your StarCraft II install directory.

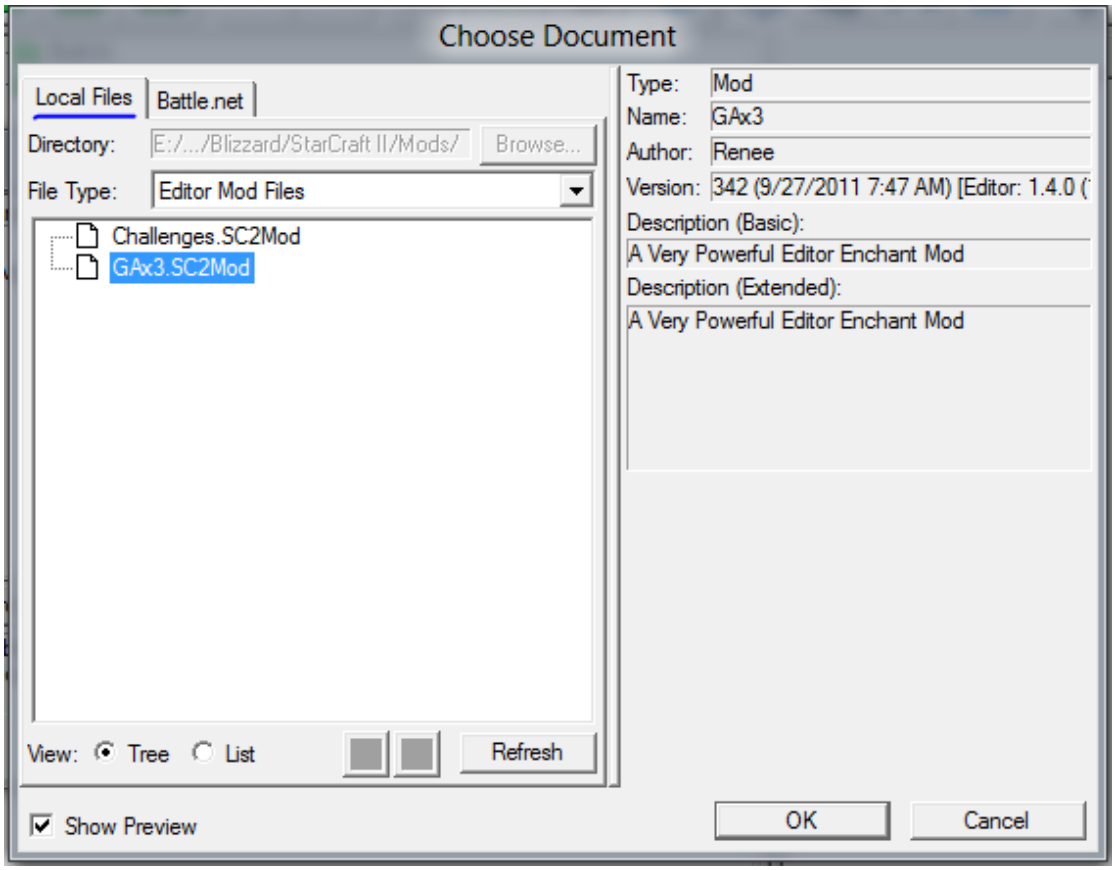


Next, I will explain how to add local dependency when creating new map and how to add it to old maps we created before:

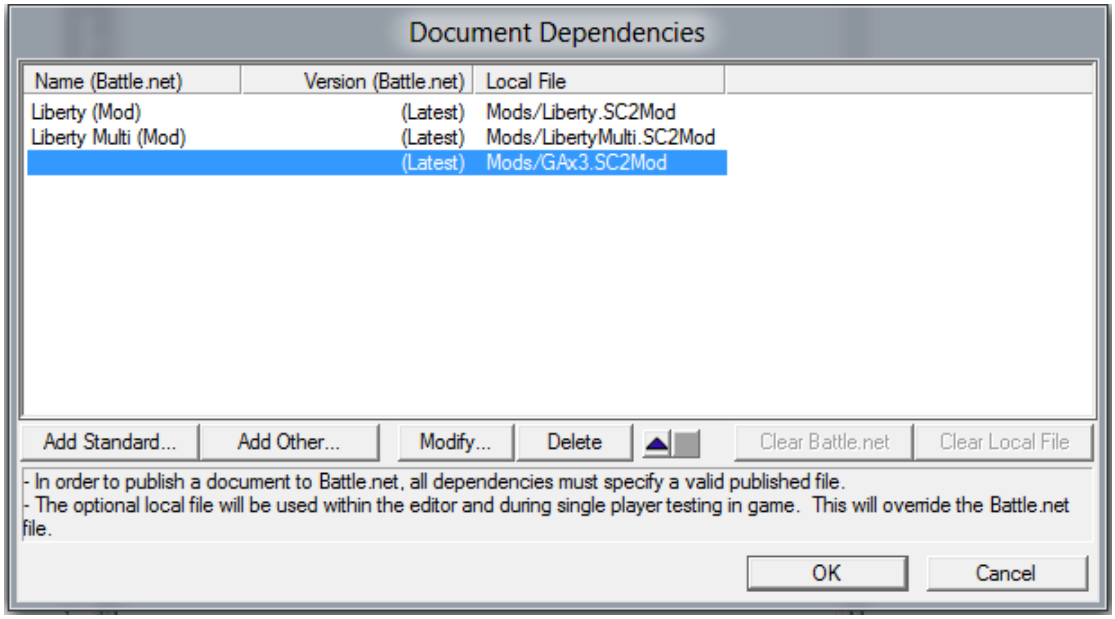
As we all know, when we create new maps, we can press the "Modify..." button to modify the dependencies of the map. To use new features provided by GAx3, all your need is to press the "Add Other..." button after you've finished your map's dependencies setting:



Then select the *Local Files* sheet in the *Chose Document* window and select GAx3.SC2Mod, press "OK".

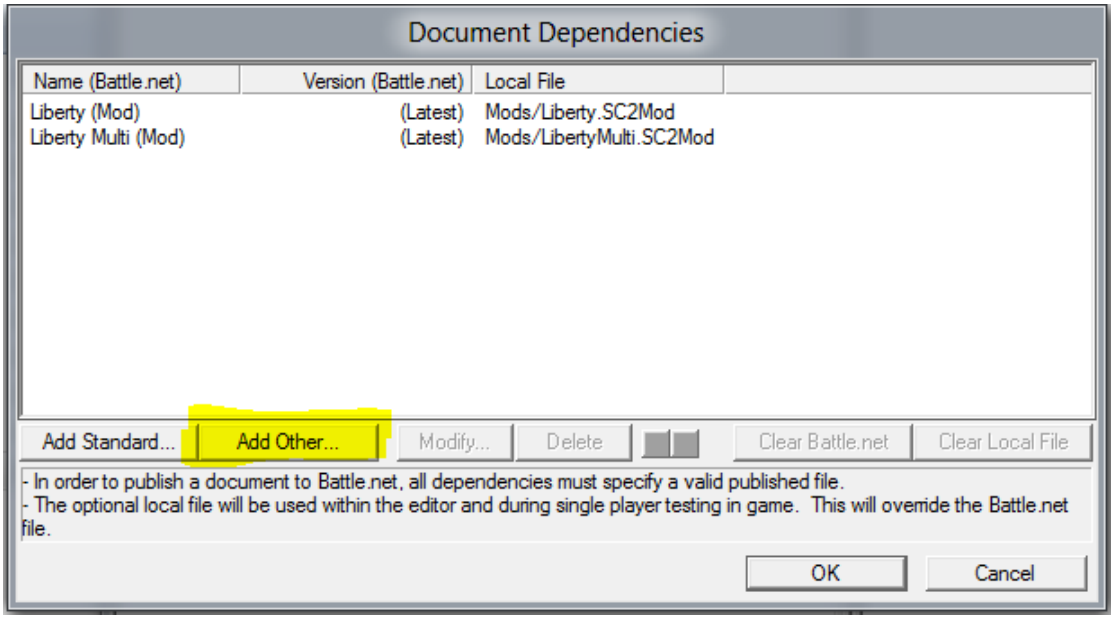


Now GAx3 have been included into your map, and you can already use all the new features of it.



And for the old maps we created before, we could also add GAx3 local dependency into them:

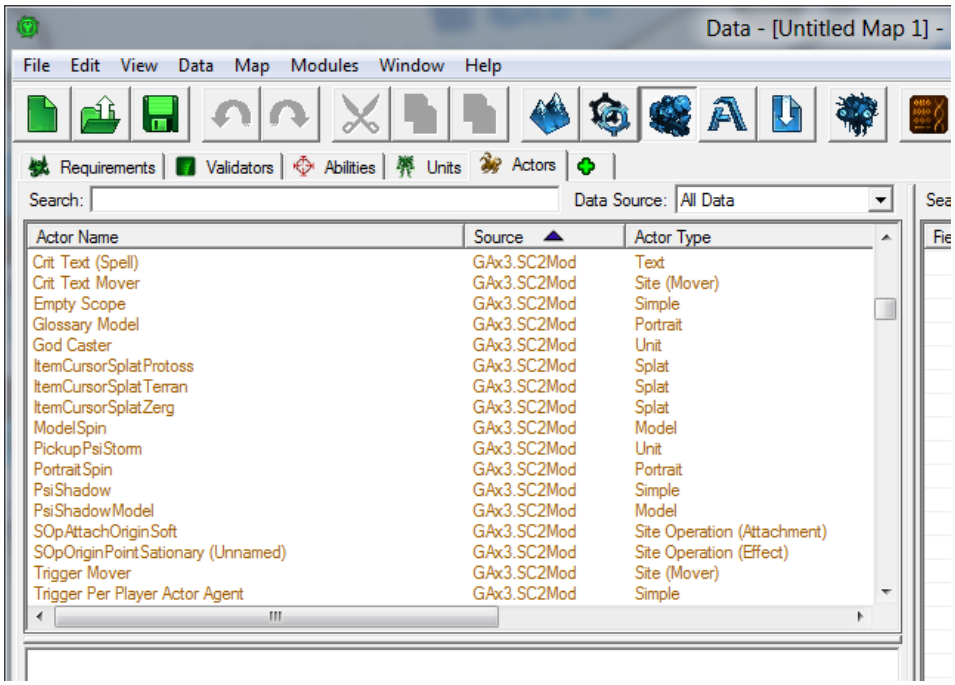
First, we use the Editor to open our map, and then press the *File -> Dependencies...* menu, the dependencies list of the map would show up. Now we can press the "Add Other..." button, and follow every step above.



How do I find the new features added by GAx3?

GAx3 new features are mainly in Data Editor & Trigger Editor. For new data features, we just need to open Data Editor, and we can see some data entries displayed as dark orange color, these golden entries are exactly the new contents of GAx3.

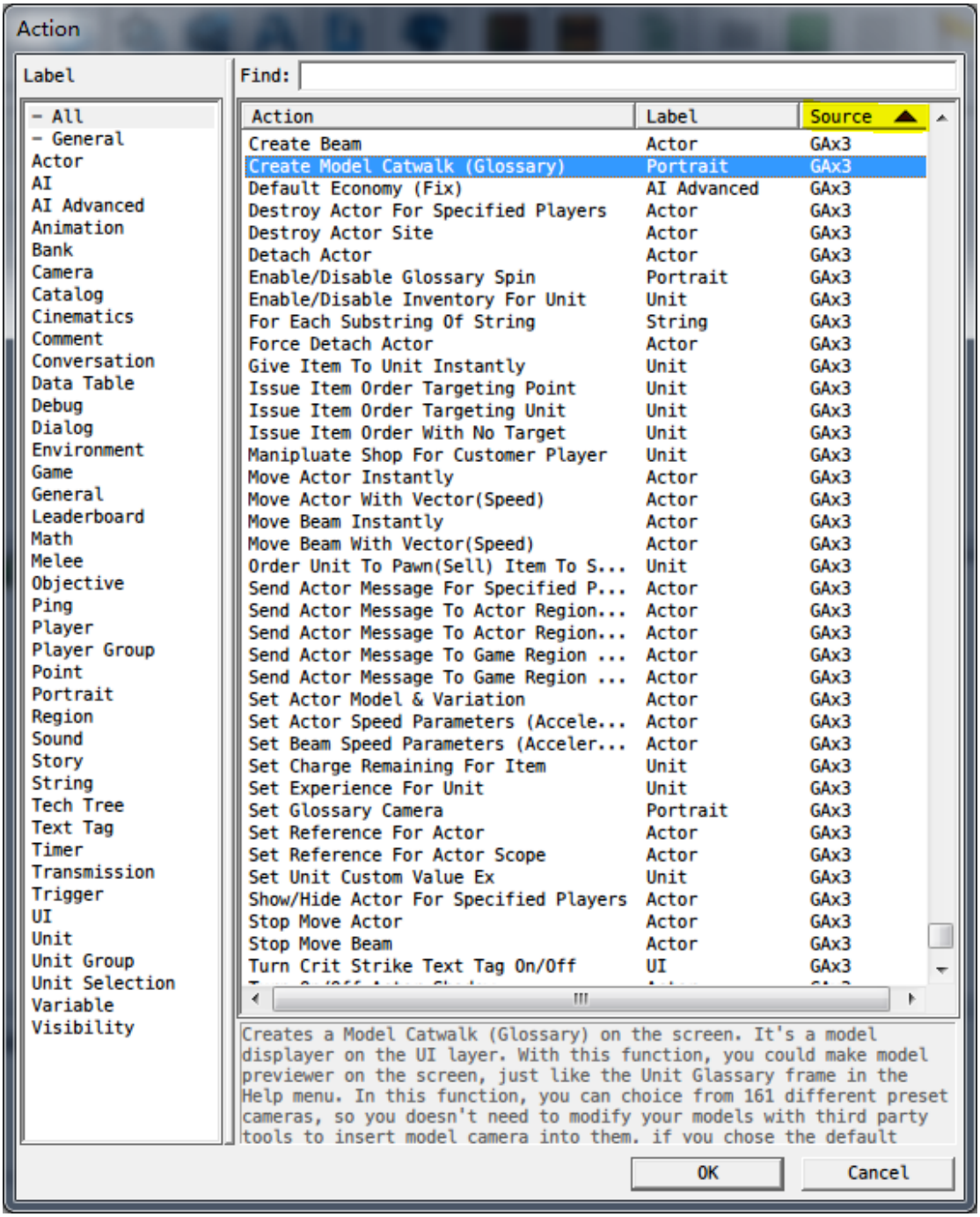
When we examine the data editor with **Table View**, we can also press the "Source" column of the list box, tell the editor the sort the items by source mod, it is convenient to arrange GAx3 entries together:



And for Trigger Editor, we can open the *View -> Show Libraries* menu to view the full GAx3 trigger lib.

These new functions can directly be used in trigger editor, when we create new event/action/conditions, we can select them in the corresponding dialogs.

We can also press the “source” column to sort the items by source mod. It is convenient to arrange GAx3 new events/conditions/actions together:



All above are just to give you a first impression of what the GAx3 is. In follow chapters, I will explain all GAx3 new features in detail.

Chapter II - Data Editor New Features

This chapter will amply explain the GAx3 new features in Data Editor. All data features here are done by raw data (Except one: the Floating Texts of the critical strike are party done by triggers, b/c the data module cannot pass the damage amount).

In the progress, you may find that most of the new data entries in GAx3 are NOT included in this chapter. This is because those entries are used to do the kernel things of the GAx3, or will be used by GAx3 trigger lib, so normal users don't need to understand them too much.

New let's begin:

Buff – Universal "Ghost Shadow" Buff

There is a buff Behavior in GAx3 named "**Psi Shadow**", it's a universal buff. Add this buff to any unit, and then they will drag "ghost shadows" when they move, just like High Templar!

These "ghost shadows" will duplicate all the information (like model, animation frame, positions, Up and Forward Rotation) of the unit at the instant of their creation.

There is some sample pictures for what will happen when we add this buff to different units":

In default condition, these psi shadows will only to be created when units move. This is because this buff has a "Target Is Moving" validator. You can modify the buff, and remove this validator, and then the psi shadow will be created even when the unit is stationary.

Buff – Spell Crit Strike & Physic Crit Strike buff

GAx3 has a buff named "**Crit 30% (Normal)**" and a buff named "**Spell Crit 30% (Spell)**". Add **Spell Crit 30% (Spell)** to any unit will cause the unit's spell damage has 30% probability to deal 1.5 fold damage amount. And add **Crit 30% (Normal)** to any unit will cause his all non-spell damage have 30% chance to deal 2 fold damage amount.

The Crit Strike system itself is done by raw data. But I used some triggers to pass the damage amount to the Crit Text Tag (The floating texts when we deal crit damage). The Crit Text Tag is turned on by default; you can use GAx3 trigger action "**Turn Crit Strike Text Tag On/Off**" to turn on/off these text tags.

In GAx3, non-spell crit damage will displayed as white color, and spell crit will be orange.

How to: Modify the Crit chance and create new Crit buffs:
It's easy: you can copy the two buffs, and modify their **Combat: Damage Response – Chance** field to modify their crit chance.

How to: Modify the spell and normal Crit damage fractions:
If you want to change the global spell and normal crit damage fractions, you need to modify other two buffs: **Take Crit Damage Once (Normal 2x) & Take Crit Damage Once (Spell 1.5x**). You can change their fraction field.

How to: Modify the Crit Text Tag:
You can change the color, size, vector of the Crit Text Tag by modify the two actors: **Crit Text (Normal)** & **Crit Text (Spell)**。

Buff –"Disable Inventory" buff

GAx3 has a buff named "Disable Inventory". Add it to any units can disable their inventories. A disabled inventor cannot pick/move/use any items. Remove the buff will undo the effect.

Actor – AoE Item Cursor Splats

Though using GAx3 Mod doesn't need you to include the official campaign mod into your map. But if you have added both GAx3 and official campaign dependencies in your map, then the "TestInventory" in campaign mod will receive a very special improvement: Any AoE items (must be ability items, e.g. items created by CItemAbil) in this inventory will display AoE cursor splats when they are in placement stage. And the model of the splats displayed will change with the player's race.

Yes, all features in this mod are support 1.3.x and 1.4.x, so this feature also works in 1.3.x (In fact, when I done the mod and start to write the manual, it is still 1.3.x, it takes me much time to localize it into English)

There three sample pictures when **Test Hero** trying to use a Psi Storm item:

When used by Terran players:

When used by Protoss players:

When used by Zerg players:

Validator – "Caster Is Behind Target" & "Caster Is In Front of Target"

When we creating abilities like "Backstab" in WoW, it is always required to judge if the caster is behind the target, or there may be some abilities need the caster stand before the target.

GAx3 has two separate validators: "**Caster Behind Target**" & "**Caster In Front of Target**" to help you do the job. Now there are sample pictures when we added a "**Caster Behind Target**" validator to a ghost's snipe ability:

When we trying to snipe the marine in the black circle. There will be an error message: "Must behind the target", because the ghost is standing before the marine.

And when the ghost is behind the target, he can now use the snipe ability:

Validator – "Target Is Moving"

This validator has nothing special, it's a supporting validator for "**Psi Shadow**", but it also can be used to determine if a unit is moving.

Other Data Entries

Other entries are used to do kernel functions of GAx3, normal users don't need to know much about them, but if you have some interest for them, please refer to **Chapter V - GAx3 API List & All XML Entries** (Page 44).

Chapter III - Trigger Editor New Features

GAx3 Millstone 1 has 67 brand new events/conditions/actions. I will introduce them by sections.

String Spilt Section

Function - String Spilt:

This function can split strings with specified separator.

Returns the indexed substring from a string. It's different from the official **Word Of String** function, allowing you to customize the separator. For example [**Substring 2 of "tree,apple,life", separated by ","**] is "apple".

Loop Action - For Each Substring Of String

This is a loop action. It will run the contained actions once for each spitted substring in the string. You can customize the Separator.

It is also useful to split a string into a string array.

StarCraft II Actor Asynchronization (Phasing Tech) Section

I have included my StarCraft II Phasing Technology into GAx3, it is the most high-ranking tech in this Mod so far. In fact, it is why I decided to create this mod in the first place.

At first, this mod was done to let users use my phasing tech functions more easily, and doesn't need to know anything about the theory behind and the code. I added more features into this mod later, and now it has become an all- purpose advanced Editor Mod.

FAQ: What Is StarCraft II Phasing Technology?

The "Phasing Technology" is originally a tech used by Blizzard in WoW, which allow each player in the same game, look at the same place and in result see different things, and/or

hear different sounds. Thus, every player can play in different "phasing", and make them feel like it is a single-player Role Player Game, they can change the world by questing or by their actions, and that won't affect other players' game progresses in the meantime.

And I successfully did a very similar Phasing system in StarCraft II, by using mix trigger and data. I've created a series of universal Phasing tech functions and trigger actions, and included them into GAx3.

With these functions and actions, could easily realize the "Phasing" effects in the WoW, such as make different player **in the same game** see/hear **different things** (including actor of Units, Doodads, Sounds, Effects, Models, Texts, Splats, and even Terrain Deformations) in the same location.

For mid+ ranked editor users who know to use the actor message, you can directly use GAx3 functions to send **ANY** asynchronous actor messages to any specified actor for specified player only, and you can also send asynchronous messages to **all actors in a specified region match specified conditions.**

And for the new actor learners who aren't familiar with actor message, you can directly use packaged trigger actions to show/hide/destroy any actors for specified players only, and **doesn't need to compile the actor messages**.

There is a sample picture for let different players see different things. Picture 1 is what the Player 1's view, and picture 2 is Player 2's view in the same place in the same time. (And of cause, they are in the same game.):

Picture 1

Picture 2

FAQ: If I hide a unit's actor for Player 1 only, can player 1 still attack this unit?

There is one thing that must be remembered in the editor: All operators to the actors will

only affect the visual layer, and will never affect the logic layer. When we hide a unit's sound and arts, the unit is still there in the logic layer, and still has his abilities, sight, and footprints – we just cannot see him. So, If we need to make a "phased" unit cannot interact to a specified player, we will also need to modify the abilities' filters, or use validators.

Deeping Reading:

If you have interest for the principles behind my Phasing system, please refer to these two tutorials about the phasing tech of mine:

《Actor Phasing Technology Part 1: Show/Hide Any Given Actor For Specified Players Only》

Simplified Chinese version:
http://bbs.islga.org/read-htm-tid-52552.html
English version:
http://www.sc2mapster.com/forums/resources/tutorials/19371-tutorial-trigger-data-show-hide-destroy-any-given-actor/

《Actor Phasing Technology Part 2: Sending ANY Asynchronous Actor Message For Specified Players》

(Simplified Chinese only):
http://bbs.islga.org/read-htm-tid-53755.html

Action - Send Actor Message For Specified Players (Phasing Tech)

This action is a vital (kernel) part of my "Phasing Tech" of the StarCraft II. It allows you to send ANY actor messages to specified players only.

With this, you could easily realize the "Phasing" effects in the WoW, such as make different player see/hear different things (including actor of Units, Doodads, Sounds, Effects, Models, Texts, Splats, and even Terrain Deformations) in the same location.

This function is for mid+ ranked editor users who know to use the actor message.

Action - Send Actor Message To Actor Region With Filters For Specified Players (Phasing Tech)

This function is another vital part of my "Phasing Tech" of the StarCraft II. It allows you to send an actor message to all the actors contained within the specified region actor for specified players only.

With this, you could easily realize the "Phasing" effects in the WoW, but It also very useful in multiplayer fps games, you can hide doodads/units at far side of player's camera to highly improve the game performance without affect other players.

This function is for mid+ ranked editor users who know to use the actor message.

Action - Send Actor Message To Actor Region For Specified Players

It's a simplified version of "**Send Actor Message To Actor Region With Filters For Specified Players**", which ignores the filter parameters. For more information, look for the hint text of the function "**Send Actor Message To Actor Region With Filters For Specified Players**".

Action - Send Actor Message To Game Region With Filters For Specified Players (Phasing Tech)

It's a simplified version of "**Send Actor Message To Actor Region With Filters For Specified Players**", that allow you send message directly to the specified game region. For more information, look for the hint text of the function "**Send Actor Message To Actor Region With Filters For Specified Players**".

Action - Send Actor Message To Game Region For Specified Players (Phasing Tech)

It's a simplified version of "Send Actor Message To Actor Region With Filters For Specified Players", which ignores the filter parameters and allows you send message directly to the specified game region. For more information, look for the hint text of the function "Send Actor Message To Actor Region With Filters For Specified Players".

Action - Show/Hide Actor For Specified Players

It's a simplified version of the "Phasing Tech" in StarCraft II. That allows you show/hide actors for specified players only. **And it doesn't need the users to write the actor messages.**

This function is for new actor learners who aren't familiar with how to use the actor

message.

Action - Destroy Actor For Specified Players

It's a simplified version of the "Phasing Tech" in StarCraft II. That allows you destroy actors for specified players only. **And it doesn't need the users to write the actor messages.**

This function is for new actor learners who aren't familiar with how to use the actor message.

Actor Reference Section

Action - Set Reference For Actor

It is essentially a native function *ActorRefSet()* that can set reference for an actor, but it doesn't have GUI version in official Editor. It is very special though: it won't send messages to the actor, thus more universal (it can avoid any unexpected triggering of the actor events while it is used in users' maps). So I added GUI support for this function.

Action - Set Reference For Actor Scope

Similar to "**Set Reference For Actor**", but this action is used to set ref for actor scope. It won't send message the actor system either.

Actor Attach Section

Actions in this section are mainly used to attach/detach/reattach actors.

Actions in official editor only allow you to attach an actor to a unit, so GAx3 added many things that allow you to **attach any actor to any actor**, even if the host isn't a unit (they need to have models, of course).

In addition, you can detach actors at any time. But the most important feature is, you can **attach an existed actor to another exist actor**, and doesn't need to create new actors, with this function, you can even attach a unit to another unit, like the Terra-Tron!

Action - Attach Model To Actor

Creates a generic actor with the specified model attached to a specified actor at the specified attach point. Use "Last Created Actor" to get the actor.

Action - Attach Actor To Actor

Creates an actor attached to a specified actor at the specified attach point. Use "Last Created Actor" to get the actor.

Action - Attach Existed Actor To Actor

Attach a existed actor to a specified actor at the specified attach point.

With this function, you can even attach a unit's actor to another unit's actor (you can do the "assemble" effect of the Terra-Tron), each of the parts can still be selected and attacked separately, and display the unit states of each unit.

But remember that it will only change the visual position of the attached unit, you will need to use trigger or data to move the actual position if you want to change the actual position of the unit.

Action - Detach Actor

Detach an actor. Make it no longer attach to any actors.

Action - Force Detach Actor

Forcedly detaches an actor.

This action is enforceable, and may change the actor in some way. So normally, it's recommended to use "Detach Actor" function instead. But some actors have site Ops that would make them continue to follow the scope even if they have been detached. In that case, you could use this function to forcedly detach them.

Actor Site Section

"Actor Site" is a new preset in GAx3 trigger editor. It can be used to allocate the 3D position of an actor. Actor Site can be determined by a 3D point (Z offset included) or a specified actor's

specified attach point. It is very convenience for us to assign the target position while moving actors and beams, it can also be used to assign the launch and impact endpoints when we create a beam.

Special Note:

"Actor Site" is essentially an actor. So if you have relevant knowledge, you can send actor message to operator them. But they will NOT be included in "Last Created Actor".

It is intended, I used a complicated way to create the actor site, make them won't affect the "Last Created Actor".

This is because, in practice situation, if the actor site will affect the "Last Created Actor", many uses will create unexpected conflict when writing triggers. Example:

```
    Actor - Create a beam from (Site From (Actor for Selendis (Carrier Hero) [29.99, 30.18])'s Weapon) to (Site From (Actor for Selendis (Carrier Hero) [3.17, 1.55])'s Center) with the model Laser Drill Tripod Attack Beam
    Actor - Move (Last created actor)'s Target endpoint to (Site from the point 3D Point 001) With Speed 2.0
```

These two actions are to create a beam between to carrier hero: launched from a carrier's weapon attach point, to the other carrier's center point. And the beam's target point will move to (Site from the point 3D Point 001) – a Site determined by a point. If the Site will change the (Last created actor) in its creation, then the (Last created actor) will point to the Actor Site instead of the Beam we created. The conflict will nullify the 2nd action.

Function – Site From Point

Constructs and returns an actor site and attaches it to the specified attach point of the specified actor.

You can then use it in actions that can move actors or beams; it can also be used to create beams.

Function – Site From Actor Attach Point

Constructs and returns an actor site and attaches it to the specified attach point of the specified actor.

You can then use it in actions that can move actors or beams; it can also be used to create beams.

Action - Destroy Actor Site

It can be used to destroy an actor site. Since the Actor Sites would not receive benefit from the auto-deletion of Galaxy.

Actor Move Section

Triggers in this section are mean to benefit the users, allow them move actors directly with triggers. These actions can easily move a specified actor to an Actor Site (can be a point, or a attach point of a specified actor).

Actor - Move Actor Instantly

Move an actor instantly to the specified site. The site can be a point (Z offset would also be included) or a specified attach point of a specified actor.

*Remember that if you use it to move a unit actor, it will only change the visual position of the unit, you will need to use trigger or data if you want to change the actual position of the unit.

Actor - Move Actor With Vector(Speed)

Move an actor toward the specified site with a specified speed vector. The site can be a point (Z offset would also be included) or a specified attach point of a specified actor.

The actor will blend move from its old position to the new position with the specified speed. Like a flying unit.

*But note that the speed vector would be calculated in 3D space, which is different from the moving of the units.

*Also remember that if you use it to move a unit actor, it will only change the visual position of the unit, you will need to use trigger or data to move the actual position if you want to change the actual position of the unit.

Action - Stop Move Actor

Stop the moving of an actor.

If you allowed the Deceleration, it will decelerate based on its current deceleration value (the default value is 1.0, you can change it with Set Actor Speed Parameters) before coming to a complete halt, otherwise it would be stopped immediately.

Action - Set Actor Speed Parameters (Acceleration, Deceleration, Speed, Max Speed)

Set an actor's moving Speed Parameters. It will affect the moving behavior of the actor when you use the Move Actor With Vector(Speed) on it.

Beam And Beam Move Section

"Beam" is a laser models in StarCraft II which used to connect to point. It is equivalent to the "Lightning Effect" in WC3 world editor. Actions in this section are used to help the user create and operator the beams and you don't need to have any actor knowledge to use them.

"Beam" is essentially an actor. So you can use most of the actor functions/action on them. But when it is come to move a beam, unlike other actors, beams have two endpoints, and cannot be moved as a whole, so I also added special actions to move the beam. It can move their endpoints separately.

Action - Create Beam

Create a beam from a site to another site with the specified model. The sites can be points (Z offset would also be included) or specified attach points of specified actors.

If you chose an attach point site, the endpoints of the beam will attach to the attach point when the target actor moves. So basically you can create 4 different type beams:

1. A 3D point -> a 3D point
2. An actor's attach point -> a 3D point
3. An actor's attach point -> an actor's attach point
4. A 3D point -> an actor's attach point

*Note that the models have bounding volume, if your camera pans too far away from the beam's launch point, the beam model may disappear. The range based on the beam model itself. You need to use 3rd party model tools to modify them. But if you using campaign Mod, it is recommended that using "Laser Drill Tripod Bigger Attack Beam" or "Laser Drill Tripod Attack Beam" as beam model, which have very long disappear ranges.

Action - Move Beam Instantly

Move an endpoint of the specified beam to the specified point instantly. The site can be a point (Z offset would also be included) or a specified attach point of a specified actor.

*Note if you use this action to move an endpoint of a beam, the endpoint will automatically cancel its original attach (If it had been attached to an actor).

Action - Move Beam With Vector (Speed)

Move an endpoint of the specified beam toward the specified point with a specified speed vector. The site can be a point (Z offset would also be included) or a specified attach point of a specified actor. The actor will blend move from its old position to the new position with the specified speed.

*Note that the speed vector would be calculated in 3D space, which is different from the moving speed of the units.

Action - Stop Move Beam

Stop the moving of the specified beam's specified endpoint.

There is an "Allow Deceleration" parameter in this action, If you allowed the Deceleration, it will decelerate based on its current deceleration value (the default value is 1.0, you can change it with Set Actor Speed Parameters) before coming to a complete halt, otherwise it would be stopped immediately.

Action - Set Beam Speed Parameters (Acceleration, Deceleration, Speed, Max Speed)

Set the moving Speed Parameters of the beam's specified endpoint. It will affect the moving behavior of the endpoint when you use the Move Beam With Vector(Speed) action on it.

Unit Spin Section

This section will introduce a utility that can be used to spin specified units, even when they are moving. You can also set the rotation rate.

Action - Unit Set Spin

Make a unit do/don't spin.

When you turned on the spin, you can also set the rotation rate. Note it will only rotate unit's art, it won't affect unit's logical facing,

*When the unit is spinning, turn off the "Turnable" flag of the unit will receive better visual effects.

Constructed GUI Messages (For Novice)

There are some actions constructed by actor messages. For users who don't know actor message.

Action - Set Actor Model & Variation

Change an actor's model to the specified model with the specified variation.

*The actor should have a model, of course.

动作 - Turn On/Off Actor Shadow

Toggle on/off an actor's shadow. You can make a unit or other things that don't cast shadows.

Model Display Control (Model Glossary) Section

Model Display Control in GAx3 is a UI control used to render model in UI layer. With functions/actions in this section, you can customize your own Model Previewer and join it into other dialog controls, create an interface like the official Unit Glossary interface in StarCraft II's Help Menu. It also fit to create hero selection UI.

You can modify the size, position, camera, model, animation and color of the control, you can even stretch it to full screen, or make the model in it spin like the official Unit Glossary(you can also set the rotation rate).。

Q: Wait a minute, why does this thing sound like a "Portrait" control?

A: Actually, it IS essentially a portrait control. I made the Model Display Control from the portrait control. In fact, you can use most of the portrait functions on the Model Display Control.

Q: Then why should I use it instead of the portrait control. Is there any special in this control?

A: The answer is simple: Camera.

Normal portraits are very inconvenient when we are trying to set the camera. Simply put, you could only choose the model cameras build-in the model (Unless you have enough actor knowledge).

But the problem is: most models lack build-in model camera. So if you want to preview them in the portrait, you need to export them, then add new model camera into them with 3rd party model tools, then import it back into the map. It is a quite unrealistic work.

While in GAx3, Model Display Control divided the camera from the model itself. It doesn't need you to choose cameras in the model, instead, it allows you to select camera from 161 preset cameras external of your model. So you doesn't need to do any modify or import/export work to your models. It's a very convenient & efficient solution.

But there is something you need to remember while using Model Display Control:

*Model Display Control is essentially a portrait control, so you can use most of the portrait functions on the Model Display Control. But there is an exception: You need to use "**Actor From Glossary**" to get a hold of the model actor for the specified Model Display Control. And if you want to change the model of in the glossary, you need to get its actor first, and then use the "**Set Actor Model & Variation**" action to modify the model.

Now, let's get some sample pictures.

Window mode:

Full – Screen Mode:

You can also combine different models in the Glossary. Behold, the ultimate Terran weapon: the **Battle-Tank-Cruiser**.

But the most interesting part is, you can make the model in the Glossary cogredient with the unit in the map!


Achievements (F9)
Menu (F10)
Message Log (F11)
Help (F12)
0
0
20
0:54
1
Battlecruiser
Kills: 0
Rank: Recruit
550 / 550
80 / 200
0
0
0
Armored - Mechanical - Massive
M
S
H
P
A
Y


Achievements (F9)
Menu (F10)
Message Log (F11)
Help (F12)
0
0
20
1:06
1
Battlecruiser
Kills: 0
Rank: Recruit
550 / 550
87 / 200
0
0
0
Armored - Mechanical - Massive
M
S
H
P
A
Y

Function - Actor From Glossary

Return the actor for the specified Glossary.

Though most of the portrait functions can be used on the Model Display Control, you still need to use "Actor From Glossary" to get a hold of the model actor for the specified Model Display Control. And if you want to change the model of in the glossary, you need to get its actor first, and then use the "Set Actor Model & Variation" action to modify the model.

Action - Create Model Catwalk (Glossary)

Create a Model Display Control (Glossary) on the screen. It's a UI layer model displayer in GAx3. With this function, you could make your own model previewer on the screen, just like the Unit Glossary frame in the StarCraft II's Help menu. You can also set its spin rate.

In this function, you can make your choice in 161 different preset cameras, so you don't need to modify your models with third party tools to insert model camera into them, if you chose the default cameras, it will automatically apply the Glossary camera of the model which set in the Data Editor.

Action - Set Glossary Camera

Though you can use official action: "Set Portrait Camera" to set the camera for the Model Display Control, the "**Set Glossary Camera**" action is way more convenience to do the work. It allows you to select camera from 161 preset cameras, you doesn't need to input the camera name manually.

Action - Enable/Disable Glossary Spin

Toggle on/off the spin of the Glossary, and set its spin rate.

Crit Strike Text Tag

Though the Crit Strike system is done by raw-xml, but the damage amount in its text tag still need some trigger. By default, the crit text tag is turned on. Spell Crit will show as orange text, and Physical Crit will be write text.

You can turn on/off these text tags separately.

Action - Turn Crit Strike Text Tag On/Off

Turn on/off the Crit Strike Text Tag, you can turn on/off the spell crit tag and the normal crit tag separately.


17
17
18
18
27
27
24
17
23


15
9

Any Ability Event

Normally, in SC2 Editor, you can convert most of "specified" events into "any" event by passing -1 or null as their parameters. For example, passing -1 to "player" parameter means register an "any player" event, while passing null to "unit" parameter means an "any unit" event. We can also passing null to order event to stand for an "any order" event.

But that's not so in official **ability** event: You can NOT register an "any ability" event by passing null as a parameter. So, GAx3 added a new event – "**Unit Uses Any Ability**" to solve this issue.

Event - Unit Uses Any Ability

This event fires when a unit uses any ability. You can specify which stage you want to capture – or just capture any stage.

Use "Triggering Ability" to get the ability that was used. Use "Triggering Ability Stage" to get the stage the ability was in when the event fired. Use "Triggering Ability Target Point" to get the target point for the ability. Use "Triggering Ability Target Unit" to get the target unit for the ability. Use "Triggering Player" to get the player that used the ability. Use "Triggering Unit" to get the unit that used the ability.

*Note: Since "Move" & "Attack" are all abilities, the frequency of this event will very high. You'd better to use trigger condition to do enough filter.

Unit Custom Value Ex Section

There is a feature in official editor named "Unit Custom Value", it allows you to store real value on a unit. But there is some limitation in it: You can only save 64 values on a unit, and the index should be 0-63. If we use these custom values in our publish lib, it will surely reduce the amount of custom values the lib users can use, and will cause some potential conflicts. They are definitely not universal enough.

GAx3 add a new type unit value: Custom Value Ex. It can also be used to store real values, and compare to the official one, the Ex version has two advantages:

1. The index of Custom Value Ex is string type. It can easily avoid any possible conflicts.
2. Custom Value Ex doesn't have the amount limitation as the official one. You can

store any amounts of values on one unit.

By the way, the Custom Value Ex and the official Custom Value are mutually independent. It won't affect the official value in any ways. So you can store both Custom Value Ex & Custom Value on the same units.

Action - Set Unit Custom Value Ex

Sets a unit's GAx3 advanced custom value. It's similar to the official version custom values, but allows you to use string as index instead of integer. And it doesn't have "64" store limit (while in the official version, you can only save 64 custom value on 1 unit).

It can also be consider as a custom property. You can recall your saved value using "Custom Value Ex Of Unit".

Function - 单位的强化版自定义值

Custom Value Ex Of Unit

Unit Handle ID

Handle ID is a unique identifier number for unit in the game.

Back in the WC3 world editor, the unit handle has been used widely, but mostly, it is because the WC3 editor cannot "truly" create new abilities. In the powerful SC2 Editor, the unit handle is actually not needed. But considered that many map makers are original from wc3 edit, to respect their coding habits, I still added unit handle into GAx3 mod.

In the early version of StarCraft II, we can get unit's internal handle id by get their "hidden property". But this method no long works now.

So GAx3 maintained its own handle table. Note: Handle ID in GAx3 will never be recycled. If you remove a unit which handle ID is 5, then create a new unit, the new unit won't use ID 5, and will be given a new number. This is because some user may save unit's handle id to datable as indexes - despite that it's a very bad coding habit, we cannot forbid the users to do this - And I tend to make GAx3 as universal as possible, so I decided not to recycle the unit handle id to avoid any potential conflicts.

Function - Handle ID Of Unit

Return the Handle ID of a unit. Handle ID is a unique identifier number for unit in the game.

Inventory Manipulate Section

While the SC2 Editor is very powerful, the game itself is more concentrate on other fields instead of the Hero system and the inventory system. The trigger editor support on hero system is relatively short, too. So since I tried out the whole hero system in SC2 beta, I have been planned to do a trigger lib for the inventory/item system.

This section and the follow two section of this chapter are all about new utilities for items/inventories.

This section is focused on operators for inventories. Include item orders and give item to hero instantly.

Though official trigger editor has added GUI support for "issue order to unit", but we cannot use it to order unit to use an item. These orders need 3 entities: Unit, Item and Target, while the official GUI could only set "unit" and "target" parameters. So GAx3 add item orders, allow GUI users to issue no-target/point-target/unit-target item orders. And there are some additional features, like "**Give Item To Unit Instantly**".

Action - Issue Item Order With No Target

Issue an item order to a unit with no target. E.g. Use (Instant)/Cancel an item.

Action - Issue Item Order Targeting Point

Issue an item order to a unit targeting a point. E.g. Use (Target)/Drop/Move an item.

Action - Issue Item Order Targeting Unit

Issue an item order to a unit targeting another unit. E.g. Use (Target)/Drop/Move/Take an item.

*Note: If you want to order a unit to pick up an item, the *Target* parameter and the *Item* parameter need to be the same (i.e. the item you want to pick up).

Action - Give Item To Unit Instantly

Give an existed item to unit instantly, without issue any orders to the unit or interrupt his current orders. It's not some remove/create trick.

*Note: The item has to be an un-carried item.

*In fact, in a GAx3 early internal version, there is another action - "Drop Item Instantly", but due to the great change of the game system in SC2 1.3.0, that action doesn't work anymore, thus has been removed.

Action - Enable/Disable Inventory For Unit

Enable/Disable inventory for a unit, a disabled inventory cannot pick/drop/use/move items.

Shop Manipulate Section

This section is focused on shop manipulates. The official "Issue Order" action cannot fit the situations like "order a player to purchase item from a neutral shop" for these orders need 3 entities: the Customer player, the purchase Agent unit of the player and the shop. And "Issue Order" action cannot set the Customer. So here added some operator manipulate for shops, to help the players to solve the purchase/pawn issues about the neutral shops.

* To make a unit to become a shop, the unit needs to have an "Interact" ability (i.e. an ability made from CAbilInteract), with "Share control" flag checked.

*To interact with the shop, the Customer player will also need to have an Agent within the range of the shop unit's interact ability.

*It the Customer player wants to purchase things which will cost resources, he will need to **turn on "Resource Spending" alliance aspect towards the owner of the shop** to allow the shop player spend your "money". If the shop belongs to a neutral player, you can skip this part, because neutral player can spend anyone's money by default.

Action - Manipulate Shop For Customer Player

This function will instruct the shop unit to provide services for specified player. Simply put, you can use it to make shops sell unit/item/ability to specified player. If an ability/item/unit needs to cost resources, it would ask the customer player to pay the price instead of the owner of the shop. And the sold goods will belong to the custom player (There can be

exceptions: you can modify the effect player of the abilities in Data Editor).

Action - Order Unit To Pawn(Sell) Item To Shop For Player

Order a unit to pawn a carried item to the specified shop, the refund resources will belong to the specified customer player.

*The customer player can be different from the owner of the Agent unit, it also works! The resources will still belong to the specified customer player.

*The shop needs to have a "pawn" ability (CAbilPawn), and the unit need to within the range of the shop's pawn ability.

Item Type/Charge/Cooldown Section

The official editor lack functions to directly operator the charge and cooldown, functions in this section are here to remedy these shortages. There are functions/actions could directly get and set the item charges and cooldowns – doesn't even need the users to provide the charge link or cooldown link.

Since only Instant Items (CItemEffectInstant), Target Items (CItemEffectTarget) and Ability Items (CItemAbil) has cooldowns and charges, you definitely cannot set charge or cooldown for Generic Items (CItem).

Function - Item Type Of Item

Return the item type for the specified item.

Action - Add Charge Used For Item

Consume the specified number of charges for the specified item.

*Note the *Charge Used* need to between 0-> Max Count So if an item has 5 max counts, you could consume 5 counts at most, i.e. the *Charge Remaining* cannot be a negative value.

Action - Add Charge Regen Time For Item

Adds to the amount of time it takes to regenerate the charge of the specified item.

Action - Set Charge Remaining For Item

Set the amount of charge remaining count.

*Note the *Charge Remaining* cannot be a negative value, nor can it greater than the max count.

Function - Charge Used For Item

Return the number of charges used for the specified item.

Function - Charge Remaining For Item

Return the number of charges remaining count for the specified item.

Function - Charge Regen For Item

Return the charge regen time for the specified item.

Function - Charge Info For Item

Return the specified charge information for the specified item. The information can be Max count, Start Count, Use Count, Remaining Count, Max Regen Time, Start Regen Time and Remaining Regen Time.

Action - Add Cooldown For Item

Add cooldown time to the specified item.

Function - Cooldown For Item

Return the cooldown time for the specified item.

Chapter IV - Fixes For Official Editor

GAx3 not only add many new features, but also fixed some current existed official editor bugs in its capacity allows. Now let's check the fix list:

Unlock Melee Hero Experiences

Since 1.2.0, using standard melee triggers will make any heroes in this map no longer get experiences, it troubled some Hero-Melee mods. So I did a slight fix in GAx3, make all maps/mods that added GAx3 dependency will unlock hero experiences by default.

As soon as your map added GAx3 dependency, this fix will automatically be applied.

Protoss Loading Frame Fix

When there are 10+ player in the game, the Protoss loading frame on the loading screen will lost its background picture, and become transparent like this:

GAx3 also fixed this. As soon as your map added GAx3 dependency, this fix will automatically be applied.

Zerg Player Hero Icon Fix

This is an old issue, too: Zerg Player cannot display the hero icon properly, they will see a purple box instead of a hero icon.

GAx3 also fixed this. As soon as your map added GAx3 dependency, this fix will automatically be applied.

The left picture is before the fix, the right one is the same map, but added GAx3 dependency, so it got fixed.

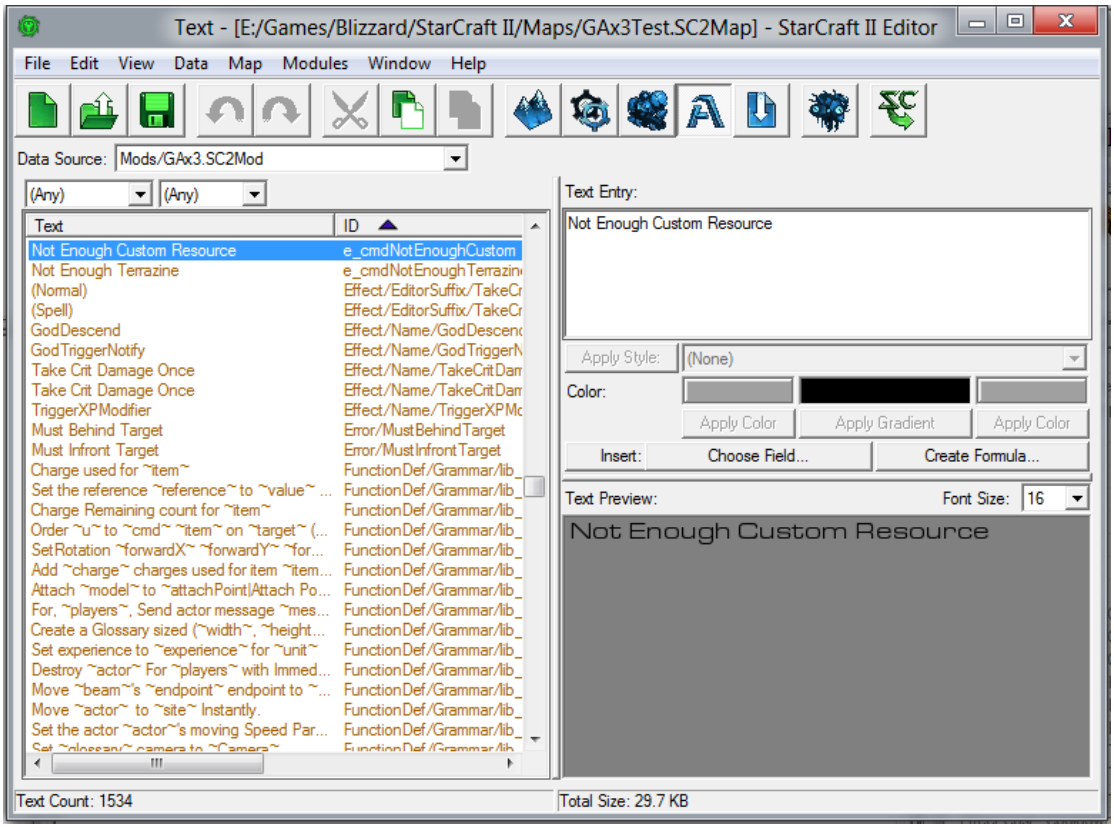
Custom Resource Error Message Fix

When we try to use a ability that need to cost custom resources or terrazines, but we don't have enough resource, the current SC2 will throw internal error strings:

e_cmdNotEnoughCustom或e_cmdNotEnoughTerrazine

GAx3 also fixed this. The game will show "Not Enough Custom Resource" and "Not Enough Terrazine" instead of two internal strings:

We can also customize the two messages: open the Text Editor module, and select "GAx3" as data source, then you can find the two messages and modify them.



Create Actor Scope (Fix)

Since SC2 1.4.0, the official "**Create Actor Scope**" action is partly broken: You cannot input the *Scope Name* parameter directly. GAx3 did a fix for it, you can find a new action in the action list named "**Create Actor Scope (Fix)**".

In the new action you can select the main actor of the scope, and doesn't need to input it manually.

This fix won't be automatically applied, because it didn't make any change to the official action. I just added a new fixed action for everyone to use.

Set Rotation (Fix)

The official constructed message function "**Set Rotation**" has a bug from the beginning: In all 6 parameters, the later 3 parameters (Up X, Up Y, Up Z) totally don't work. So GAx3 prepared another fixed function.

I will explain the fix effect with a pair of pictures.

In the following picture, I send a custom scripted message "SetRotation {1,-1,1 1,-0.5,1}" to the left Blimp, so the left one is the correct rotation result. And then I send the same 6 parameters with official "**Set Rotation**" action to right side Blimp. The rotation result is definitely different.

While in the following picture, I send the same 6 parameters with GAx3 fixed function "**Set Rotation (Fix)**". Then we shall see the two models had the same rotation result.

This fix won't be automatically applied, because it didn't make any change to the official action. I just added a new fixed action for everyone to use.

Default Economy (Fix)

The official "Default Economy" action has broken from the beginning. This action doesn't consistent with its native prototype, and if you use it in your map, it will prevent you from saving the map. And GAx3 added action "Default Economy (Fix)" to do the fix.

This fix won't be automatically applied, because it didn't make any change to the official action. I just added a new fixed action for everyone to use.

Triggering Effect Unit Owner (Fix)

The official "Triggering Effect Unit Owner" function doesn't have any big issues. The only problem is you cannot select the "Creator" player in the GUI when you set its "Location" parameter.

GAx3 added a "Triggering Effect Unit Owner (Fix)" action, and added "Creator" into the preset list.

This fix won't be automatically applied, because it didn't make any change to the official action. I just added a new fixed action for everyone to use.

PickupPsiStorm Model Fix

It's just a random fix. The ability item "PickupPsiStorm" in official Campaign mod doesn't have an actor, so it will be displayed as a sphere in the game. And GAx3 added a model for it.

This fix will automatically be applied if you added both GAx3 and the official Campaign mod dependencies.

Chapter V - GAx3 API List & All XML Entries

This chapter is exclusive for XML/Galaxy users; I listed all GAx3 API and XML data entries here. And brief explanations for the entry IDs.

GAx3 All XML Entries

AbilData.xml

GodInventory: For internal realization, please do not modify it

ActorData.xml

ModelSpin: For internal realization, please do not modify it
PsiShadowModel: "Psi Shadow" related actor
GlossaryModel: For internal realization, please do not modify it
PortraitSpin: For internal realization, please do not modify it
EmptyScope: For internal realization, please do not modify it
PsiShadow: "Psi Shadow" related actor
TriggerPerPlayerActorAgent: For internal realization, please do not modify it
TriggerSite: For internal realization, please do not modify it
TriggerSiteNoHarness: For internal realization, please do not modify it
CritTextMover: Crit Text Tag Related Actor
TriggerMover: For internal realization, please do not modify it
SOpAttachOriginSoft: For internal realization, please do not modify it
SOpOriginPointSationary: For internal realization, please do not modify it
ItemCursorSplatProtoss: Protoss AoE Item Cursor Splat
ItemCursorSplatTerran: Terran AoE Item Cursor Splat
ItemCursorSplatZerg: Zerg AoE Item Cursor Splat
CritText: Crit Text Tag Related Actor
CritTextSpell: Crit Text Tag Related Actor
God: For internal realization, please do not modify it
PickupPsiStorm: Psi Storm item model fix

BehaviorData.xml

Crit30: Universal Behavior, 30% Chance to deal melee/ranged/Splash Crit damage
Crit30Spell: Universal Behavior, 30% Chance to deal spell Crit damage
DisableInventory: Universal Behavior to disable unit's inventory
Immortality: For internal realization, please do not modify it
PsiShadow: "Psi Shadow" universal Behavior
TakeCritDamageOnce: Normal Crit related behavior
TakeCritDamageOnceSpell: Spell Crit related behavior

EffectData.xml

GodDescend: For internal realization, please do not modify it
GodTriggerNotify: For internal realization, please do not modify it
TakeCritDamageOnce: Normal Crit related effect
TakeCritDamageOnceSpell: Spell Crit related effect
TriggerXPModifier: For internal realization, please do not modify it

GameData.xml

Dflt: GAx3 Default Game Data

ModelData.xml

ModelSpin: For internal realization, please do not modify it
Star2Camera: For internal realization, please do not modify it

UnitData.xml

God: For internal realization, please do not modify it

ValidatorData.xml

CasterBehindTarget: "Caster Behind Target" validator
CasterInfrontofTarget: "Caster In front of Target" validator
TargetIsMoving: "Target Is Moving" validator
IsGod: For internal realization, please do not modify it
ProtossPlayer: Target is Protoss player validator
TerranPlayer: Target is Terran player validator
ZergPlayer: Target is Zerg player validator

GAx3 API List

API For Script Users:

string GAx3_StringSpilt (string s, int n, string sl)

string GAx3_StringSpilt2 (string s, int n, string sl)

void GAx3_ActorSendAsyn (playergroup players, actor a, string msg)

void GAx3_ActorShowAsyn (playergroup players, actor a, bool inShow)

void GAx3_ActorShowAsyn2 (playergroup players, actor a, bool inShow)

void GAx3_ActorDestroyAsyn (playergroup players, actor a, bool immediate)

void GAx3_ActorRegionSendAsyn (playergroup players, actor a, int intersect, string msg, string filters, string terms)

void GAx3_ActorRegionSendSimpleAsyn (playergroup players, actor a, string msg)

void GAx3_ActorSendToGameRegionSimpleAsyn (playergroup players, region r, string msg)

void GAx3_ActorSendToGameRegionAsyn (playergroup players, region r, int intersect, string msg, string filters, string terms)

void GAx3_TriggerAddEventUnitAnyAbility (trigger t, unitref u, int stage, bool includeSharedAbils)

actor GAx3_BeamCreate (string inModelLink, actor from, actor to)

actor GAx3_ActorSiteFromActorAttach (actor a, string inAttachment)

actor GAx3_ActorSiteFromPoint (point p)

void GAx3_ActorSiteDestroy (actor a)

void GAx3_ActorSetPosition (actor a, actor site)

void GAx3_ActorMove (actor a, fixed s, actor des)

void GAx3_ActorMoveStop (actor a, bool allowDe)

void GAx3_ActorSetSpeed (actor a, fixed ac, fixed de, fixed s, fixed ms)

void GAx3_ActorAttach (actor h, actor a, string inAttachment)

void GAx3_ActorAttachNewTo (actor h, string inActorLink, string inAttachment)

void GAx3_ActorAttachNewModelTo (actor h, string inModelLink, string inAttachment)

void GAx3_ActorDetach (actor a)

void GAx3_ActorForceDetach (actor a)

void GAx3_BeamSetPosition (actor a, string endpoint, actor site)

void GAx3_BeamMove (actor a, string endpoint, fixed s, actor site)

void GAx3_BeamMoveStop (actor a, string endpoint, bool allowDe)

void GAx3_BeamSetSpeed (actor a, string endpoint, fixed ac, fixed de, fixed s, fixed ms)

void GAx3_ActorMessageModelSwap (actor a, string inModelLink, int var)

void GAx3_ActorCastShadow (actor a, bool inShow)

void GAx3_UnitSpinSet (unit u, bool spin, fixed spinRate)

int GAx3_ModelGlossaryCreate (int offsetX, int offsetY, int anchor, int width, int height, string modelLink, string cameraLink, bool spin, fixed spinRate, string animProps, bool visible, bool waitUntilLoaded)

actor GAx3_ActorFromGlossary (int p)

void GAx3_GlossarySpinEnable (int p, bool spin, fixed spinRate)

void GAx3_UnitSetCustomValueEx (unit u, string att, fixed val)

fixed GAx3_UnitGetCustomValueEx (unit u, string att)

int GAx3_UnitGetHandle (unit u)

void GAx3_UnitAddXP (unit u, fixed xp)

void GAx3_UnitSetXP (unit u, fixed xp)

string GAx3_UnitInventoryGet (unit u)

void GAx3_UnitInventoryEnable (unit u, bool inEnable)

bool GAx3_UnitIssueItemOrderTargetNone (unit u, int inCmdIndex, unit inItem, int inQueueType)

bool GAx3_UnitIssueItemOrderTargetPoint (unit u, int inCmdIndex, unit inItem, point inPoint, int inQueueType)

bool GAx3_UnitIssueItemOrderTargetUnit (unit u, int inCmdIndex, unit inItem, unit inUnit, int inQueueType)

bool GAx3_UnitInventoryGiveItem (unit u, unit inItem)

bool GAx3_UnitIssuePawnOrderForPlayer (unit u, unit inItem, unit shop, int inPlayer, int inQueueType)

bool GAx3_UnitIssueOrderForPlayer (unit u, order inOrder, int inPlayer, int inQueueType)

string GAx3_ItemGetType (unit inItem)

string GAx3_ItemGetChargeLink (unit inItem)

string GAx3_ItemGetCooldownLink (unit inItem)

void GAx3_ItemAddChargeUsed (unit inItem, fixed inVal)

fixed GAx3_ItemGetChargeUsed (unit inItem)

void GAx3_ItemSetChargeRemain (unit inItem, fixed inVal)

fixed GAx3_ItemGetChargeRemain (unit inItem)

void GAx3_ItemAddChargeRegen (unit inItem, fixed inVal)

fixed GAx3_ItemGetChargeRegen (unit inItem)

fixed GAx3_ItemChargeInfo (unit inItem, int inType)

void GAx3_ItemAddCooldown (unit inItem, fixed inVal)

fixed GAx3_ItemGetCooldown (unit inItem)

void GAx3_CritStrikeTextTagShow (bool showNormal, bool showSpell)

string GAx3_Fix_SetRotation (fixed lp_forwardX, fixed lp_forwardY, fixed lp_forwardZ, fixed lp_upX, fixed lp_upY, fixed lp_upZ)

Internal Realization API (Has no use for script users)

void GAx3_GodInit ()

void GAx3_ResumeXPGaining ()

```
bool        libGAx3_GodDescend (bool testConds, bool runActions)

bool        libGAx3_DamageAmountSendN (bool testConds, bool runActions)

bool        libGAx3_DamageAmountSendS (bool testConds, bool runActions)

bool        libGAx3_PredamageRegisterN (bool testConds, bool runActions)

bool        libGAx3_PredamageRegisterS (bool testConds, bool runActions)

int         GAx3_Main ()
```

Appendix I - Removed Features

Since I'm the only author of GAx3 and I've been busy these years, it took me much time to do it. So, StarCraft II itself had been updated many times since I started this project. Any some great change in SC2 made many features once existed in GAx3 had become outdated/replaced by official version or just doesn't work anymore (of course, the patches of SC2 also brought some new features into GAx3 as well), and I had removed them from the mod.

In the Appendix I, I will list some feature that once existed in GAx3 since the start of this project, but have been removed before its first public releasing (Millstone 1).

*Note: Please do not read it as a sign that I will remove more outdated features in future - I won't. I did this in Millstone 1 is just because it is the first public version of GAx3, thus I can make a clear release. But for compatibility, I will never remove any feature in the future versions of GAx3, I will hide them instead of just remove them, to make sure the updating of GAx3 won't affect the maps that used older version of it.

Action - Set Experience For Unit/Action - Add Experience For Unit

StarCraft II 1.1.0 added an official feature that can modify unit's XP directly. So I removed the GAx3 version.

Fix - Actor of Three Official "Blink" Abilities

Before SC2 1.4.0, the actors of the three official "Blink" abilities has been bugged for a long time. The visual effects of their launch effects have been created in their impact point. I once fixed them into GAx3.

But in 1.4.0, they have official fixed their visual effects. So the fixed in GAx3 has been removed. But If you are still using 1.3.0 Editor, you can look at my demo to fix their visual effects by you self.

The Fixes of Three Official "Blink" Abilities (For 1.3.x users only, Simplified Chinese)
http://bbs.islga.org/read-htm-tid-215907.html

Action - Drop Item Instantly

This action can drop an item from a hero's inventory instantly, without sending any orders to

the hero. But due to some system change of 1.3.0, this action doesn't work anymore, so too became a removed feature.

Action – Destroy Dialog (Fix)

In the early version, official editor had a bug that would crash the game while you try to destroy a dialog with a Panel control in it. And GAx3 once had a fix or this action, to destroy dialog safely.

Since SC2 1.4.0 has fixed this bug, I removed it.

Function – Behavior Has Flag (Fix)

There were bugs in this function before. But since SC2 1.4.0 fixed it, this feature is no longer needed.

Model - Terrain Object Trench Diagonal

Before 1.4.0, the model of the official terrain objects: **Terrain Object Trench Diagonal** has a serious bug that will cause great damage to the maps. GAx3 once have a normalized model to replace it.

And again, 1.4.0 fixed this, then the feature has been removed.

Appendix II - Copyright & Feedback Address

Copyright Announce:

The author of GAx3 Mod AND this Manual is **麦德三世(English name: Renee)**

All of the unique techs in this Mod, such as Psi Shadow, Phasing Technology, Model Display Control, Unit Custom Value Ex, Caster is behind the target Validator… and any others are all my original creation.

Though this mod isn't encrypted, nor had it code confused. But if you use this mod to making maps, or use some unique technique from this mod in your maps. It would be better to mention it in your maps.

Question/Feedback/Suggestion/Latest News for GAx3:

Goblin Academy GAx3 Forum (Simplified Chinese, recommended)
http://bbs.islga.org/thread-htm-fid-133.html

SC2Mapster Project Forum (English):
http://www.sc2mapster.com/forums/resources/project-workplace/
SC2Mapster GAx3 Project Asset:
http://www.sc2mapster.com/assets/gax3-starcraft-ii-editor-enchant/

US Battle.net Custom Maps Forum (English):
http://us.battle.net/sc2/en/forum/13437/

Or

Send Email to me:
Mailto:Karazhan@126.com