

Release Notes for *GURPS Character Builder*™ 2.5.8

Thank you for installing *GURPS Character Builder*.

Note: If you're upgrading from a previous version, you'll need to update your old character sheets to the new level (see below).

For the most current *Character Builder* news, see the Steve Jackson Games website at <http://www.sjgames.com/gurps/characterbuilder/>.

Multiple Versions

If you need to keep multiple versions of *Character Builder* around (for compatibility with other users), you can install *Character Builder* in a different directory. See Multiple Installations, below.

Updating Character Sheets from Previous Versions

When you run the new *Character Builder* and open an old character sheet, you'll get a message similar to the following:

The level of the character sheet (2) is different from the level of the game system (3). You may wish to convert the character sheet to the current version with the File | Convert command.

You'll also get similar messages if you try to print old character sheets, or load old data sheets.

Why You Need to Update Previous Character Sheets

A new version of a game system template may have several new features that make it incompatible with the previous version.

If you don't update your character sheets, they may not print properly, and you will not be able to add items from the new data sheets to the old character sheet. You may encounter undefined functions, errors and other unpleasantness.

How to Update Character Sheets

See the *Character Builder* Reference Manual for details on updating your character sheets.

Note that some characters will not have the same character point total after conversion. This is because some advantages (Hidebound, Versatile, Manual Dexterity, etc.), were previously defined to include bonuses and penalties for a number of skills. Since these adjustments are situational, rather than permanent, they cannot be added by the advantages to modify the base skill level. Thus, you need to decide whether the character should keep the higher level, or the higher cost for the affected skills.

Integrating Custom Changes into the New Version

If you've created your own character templates, data sheets or print templates you will need to convert them to use the new features.

What's Going On?

First, a little bit about how the character templates, data sheets and print templates work together.

The character sheets provide a set of functions that are used in the item definitions in the data sheets and the print templates. If you use a print template or data sheet with the wrong character sheet, the required functions may be missing or different. *Character Builder* can detect this difference (by comparing levels) and will alert you when it occurs.

Updating Character Templates

If you've made a new character template that is a copy of the character templates with different configuration parameters, initial stats, or have preselected some skills, just convert the template in the standard fashion:

- Open the character template
- Run the normal conversion (see "converting old character sheets" in the *Character Builder* help).

- Save the converted template with a new name (ending in `.cst`) in the source directory where all the templates are stored.
- When you're satisfied that the converted template is correct, copy it over the original.

If you've changed the layout of the dialogs, added new stats, lists, etc., you'll have more work integrating the changes. There's no magic bullet: take a copy of the new character sheet template and make the same changes you made the last time.

Updating Data Sheets

If you load a custom data sheet based on the previous version of a data sheet after you've loaded the new (standard) data sheet, you'll get the following message:

The level of this data sheet (2) is different from the level of the data sheets already loaded (3).

To find the level of the current character sheet templates:

- Create a new **GURPS** character sheet with **File | New....**
- Select the `GURPS.cst` character sheet template.
- Click **OK**.
- Select the **Modify | Character Sheet Info...** command.
- The level is specified in the **Level** edit field.

The `GURPS.cds` file also contains the current level in the line near the top of the file that begins with "level".

All character sheet templates, print templates and data sheet templates should be at this level.

Macro Data Sheets

To update the level of your macro data sheets:

- Open the data sheet in **Character Builder** with **File | Open....**
- Select the **Modify | Info...** command.
- Change the Level edit field to the level of the character sheet template (which is 3 in the 2.3 release).

If You Edited the Data Sheet by Hand

Edit your `.cds` file and find the line near the top that says

```
level 2
```

Change the old level to the new level. Usually the old character sheet will just work with the new macros, but if you made custom changes to the `.inc` file you'll also have to edit the new `.inc` file.

If You Edited and Saved the Data Sheet with **Character Builder**

You probably have a lot of work ahead of you. Items are usually defined using macros. When you save a data sheet with **Character Builder** the items are saved without the macros. That means that when you load the data sheet again, you won't have the macro definitions from the new macro include file, which means that your old data sheet will probably not work with the new character template.

You'll need to change the items so that they use the new item definitions. To see what you need to do, look at an item from the new data sheet. You'll also need to change the level of the data sheet to 3:

- Choose the **Modify | Info...** command while the data sheet is open.
- Change the Level to the appropriate value.
- Click **OK** and save the data sheet.

Doing this without changing the item definitions is almost certain to be insufficient.

In general, it's better to use macro data sheets or to edit the data sheets with a text editor so that you can take advantage of the macro facility.

If You Changed an Include File

You're going to have to do major integration work. You'll need to make those same changes to the new include file. If you're doing this, it's probably best to rename everything so that future installations overwrite your changes. The same is true if you modified data sheets directly.

Updating Print Templates

If you try to print your old character sheets with a new print template, or new characters with your old print template, you'll get a message similar to the following:

The level of the character sheet (3) is different from that of the print template (2). Errors may occur during printing or viewing. Do you wish to continue?

In this case you need to update your custom print template. The first level of "fix" is to change the level of the print template:

- Edit the print template.
- Choose the **File | Template Information...** command.
- Change the Game System Level to the appropriate level.
- Click **OK**.
- Save the print template.

Multiple Installations

If you want to keep different versions of **Character Builder** around for compatibility with other users, or for evaluation purposes, you can install the application in a different directory. Simply choose the desired target folder during installation.

By default both installations will use the same configuration parameters in the Windows registry. If you want to keep different preferences for each installation, follow these steps:

- Create a text file named `gcb.ini` in the directory where you installed the new application (the new source directory).
- Add the following text (assuming that you installed to the folder `C:\Program Files\New GURPS`):

```
[creator]
Source Directory=C:\Program Files\New GURPS
```

When **Character Builder** starts up, it will look for a file named `gcb.ini` in the directory where the application is installed. If that file is found, Metacreator will store all the configuration data in that file.

If you want to store the `.ini` file in another directory, you can redirect the location of the `.ini` file by adding the following entries to the `gcb.ini` file in the directory where the application is installed:

```
[Redirect]
Profile=C:\My Directory\gcb.ini
```

Finally, run the original installation of **Character Builder**, select the **Utilities | Preferences...** command and set the **Source Directory** to the installation directory you wish to use with that version of the application. The two installations will now be independent of one another.

Character Builder Changes

The following is a summary of the changes to **Character Builder** since version 2.2.

GURPS Game System Template Changes

- Added Template Builder and Race Builder templates and filters to make creating character and race templates easier.
- Added `GURPS3x5.prt` print template to print character summaries on 3" x 5" index cards.
- Perform (and reverse) aging rolls when character age is increased in the Information dialog.
- Added data sheets for **GURPS Bio-Tech**, **GURPS Cliffhangers**, **GURPS Spirits**, **GURPS Black Ops**, **GURPS High-Tech**, **GURPS Low-Tech**, **GURPS Rogues**, **GURPS Swashbucklers**, **GURPS WWII**.
- Added **Data | Generate Appearance...** command.
- Made TL print in standard location when skills have specializations.
- Display * by level or cost of items that have a bonus to indicate that the cost has been modified by another item. This allows you to quickly identify skills whose costs have been adjusted by an advantage or disadvantage.
- Added the ability to include college skills directly in the spell list, in addition to printing them in the grimoire.
- Load data sheet for characters created through templates as the data sheets are needed for specific characters, rather than load all data sheets that might be needed for characters included in a character template data sheet.
- Added rule sets for Modern, Space and Fantasy to help weed out items that are inappropriate for certain settings.
- Added a `GURPSModern.cst` character sheet template with the appropriate settings for a modern campaign.
- Added `GURPSCliffhangers.cst` template for Cliffhangers campaigns.
- Macro data sheets are now the preferred data sheet format, instead of binary data sheets.
- Allow defines to be specified in macro data sheets.
- Make the preference setting for metric display measurements in metric on the character sheet.
- Character sheet changes.
 - Fixed a bug that caused income to not be totalled correctly.
 - Put the name of the list in the header line for each item window so that it's easy to see what items are in the list if the window's title bar is too short to display the file name and list name.
 - Make cost of racial quirks and disadvantages not count against limits
 - Add unaging and other longevity advantages to the list that don't require age rules to be followed.
 - Fixed problem in jobmult function that caused named wealth levels to improperly affect the salary in jobs.
- Character development changes.
 - If user selects Undo, pay raises/cuts are rescinded.
 - If no critical failure specified, the normal failure is used for critical job roll failures.
 - Added pay cut job failure.
 - Get the name of the job and add it directly when the character development script is run and there is no job, displaying any requirements for the job.
- Character Template Changes
 - Speeded up generation of characters that have a selection of skills from a general class.
 - Allow requirements for template.
 - If TL negative on template generation, perform integrity check on items.
 - Sort template types before displaying them.
 - Added Officer lenses to templates.
 - Allow multiple templates to be generated into the same character sheet. This allows for lenses and multitemplate characters.
 - If an attribute is already at the level the template indicates (or higher), don't lower the attribute's value.

- Allow cost of item to be specified in a required item -- i.e., Strategy=#2 means to spend at least two points on Strategy.
- Make Literacy take high and low TL into account (Primitive, High Technology).
- Allow multiple instances of the same item to be added when creating a character from a template when ?multiple specified for a choose item directive for a template.
- Record beginning points, maximum disadvantages and character type in the templates, and set them in the destination character when generating a character from a template.
- Common filter changes:
 - Display list of items that modified the cost of items at the bottom of item lists, marked with an asterisk. Items that have those bonuses and penalties are marked with asterisks in the list.
 - Put comma between height and weight.
- GURPSStandardRTF.flx Changes
 - Output picture if present.
 - Set keep with next paragraph to keep stat block together.
- GURPSControl.flx changes:
 - Display skill level for damage.
- Don't print the Campaign Disadvantage option by default.
- Eliminate limit on number of damage types on weapons.
- Fixed problem with lameness affecting flying speed.
- Added .dlist files to allow users to control which data sheets are loaded when the **Add...** shortcuts are selected.
- Specify argument types for macros to make editing them easier in macro data sheets. This means that you don't have to know the exact syntax for requirements to add them to an item.
- Added Minimum and Maximum attribute advantages that prevent you from selecting attributes higher or lower than a specific value. These can be used in race or character templates that have a minimum or maximum attribute level.
- Added Cliffhangers and Modern character sheet templates.
- Don't display sex in GURPSStandardRTF.flx.
- Display advantage name and level together, placing parenthetical text afterwards.
- Fixed problem in character development with studying skills that have bonuses due to advantages.
- Make Hobby option work in all cases.
- Minimum level option on skills.
- Make sure that skill costs are always a multiple of half a point.
- Fixed problems with alcohol tolerance.
- Various Data Sheet Fixes
 - Added macro argument types to martial arts macros.
 - Adding No Cost option to equipment allows the character to have the item at no cost.
 - Added "generic" Hobby and Professional skills.
 - Make Magic Breath and Magic Jet be melee skills.
 - Fixed miscategorization of Low Empathy.
 - Make Claim to Hospitality have an arbitrary cost (eliminated levels).
 - Optionally add Super Jump if a sufficiently high level of Bouncing is taken.
 - Specify skill category for Flexibility bonus to make sure bonus is added to all Mechanic skills.
 - Fixed problem with Stretching without flexibility and skill bonuses.
 - Fixed requirements for speed reading.
 - Add bonus for Lipreading to Acute Vision and Alertness.
 - Prevent generic advantages from being chosen randomly.
 - Added Technophobia/Fear of Technology.
 - Removed the adjustments on Versatile and Hidebound.
 - Added separately named Wealth advantages and Poverty disadvantages. Multimillionaire is now a separate advantage with a prerequisite.
 - Automatically add Status advantage for appropriate levels of Wealth.
 - Added separately named Appearance advantages and disadvantages (Attractive, Ugly, etc.). The old-style level-based Appearance advantage and disadvantage are still available, but are

converted to the named item when encountered in character sheets to conform to SJ Games practices.

- Made various flight spells and psionic skills affect the Flight skill.
- Added Dreamer.
- Added Favors (used to be just an option on Ally, Contact, etc.).
- Limit Reputation level to 4.
- Fixed defaults on certain Guns and Driving skills.
- Get rid of the Manual Dexterity bonus to mental skills. The bonuses can be applied, but only on a task-by-task basis, at the GM's discretion.
- Iron Hand penalty for manual skills only if both hands affected.
- Made Crossbow ST independent of character ST.
- Fixed social skill penalty on Oblivious.
- Make One Eye and No Depth Perception affect skill defaults appropriately.
- Added many Sports skill specializations.
- Added Languages sublist to make adding a list of languages easier.
- Display damage for various kinds of teeth.
- Fixed cost of Body of Wood.
- Added Accessibility to Super Limitations.
- Allow automatic zero-cost Berserk (for weres).
- Fixed armor location for Plate Legs.
- Fixed parry on Quarterstaff (sword tech.).
- Allow Throwing skill on thrown missile weapons.
- Specify cost on zero costable items to fix problem with [Var] showing up in templates for things like High Pain Threshold.
- Improved treatment of weres.
- Make Weres satisfy requirement for non-reciprocal damage.
- Allow Regeneration to be added for weres at zero cost.
- Fixed computation of cost for Dependency to exclude Causes Aging from the multiplication.
- Lucid Dreaming is ME.
- Make Accessibility be available under super limitations.
- Fixed problems with Body of Wood, Forest Warning.
- Fixed bug in color coding in total display when established characters increased attributes.
- Fixed problems with Guerilla/Partisan, Artilleryman, Charlatan, Overlord and Elementalist templates.
- Fixed count of spells for Priest template.
- Fixed problem with Empathy, Telepathy, Detect Lies and Psychology.
- Make generic options keep values on conversion.
- Fixed problem on conversion for Berserk.
- Added Hidden Lore specialty skills.
- Set skill name options on racial group skill bonuses so that lists of specific skills can be specified in addition to categories of skills.
- Fixed problems in Increased Density.
- Reduce points on Sadism, etc., when Murder Addiction present.
- Fixed cost of Reach on Spear.
- Added Soldier skill.
- Put a TL on Cyberaxe.
- Fixed default on Head Butt.
- Fixed cost level on Animate Plant
- Fixed problem with Telepathy and Empathy Bonus on Detect Lies and Psychology.
- Fixed various minor problems in Fantasy Folk races.
- When adding a race to a character, set the attributes to the racial norms if they're currently lower than the norms
- Make appearance variable value be negative for appearance disadvantages and positive for advantages. This allows other items (Terror) to detect exactly which Appearance is present.
- Additional arguments for race macro (powers, spells, etc.)
- Improve handling of damage in natural attack macro.

- Set the values of attributes to the racial norm when adding a race.
- Add separate macro for engineering skills so that the Mathematical Ability bonus works properly with the TL.
- Fixed problem with levels being different for senses in Acute Senses than in the additional attributes.
- Make deafness affect cost of language skills
- Added Acute Sense category to acute senses so they can be easily listed in templates.
- Added Hmong, Vietnamese.
- Added Gunner (SAM).
- Always adjust Advantages:Status instead of the status variable. Make negative value for status indicate a disadvantage. This fixes a problem with Status disadvantage acting like the advantage (bestowing a bonus on Savoir-Faire, for example), and letting Status work properly when added to a Choose grouping in a template.
- Make Jitte/Sai be close combat skill.
- Check requirements after adding martial arts styles to alert user if there are some unsatisfied requirements.
- Added a mechanism to set different prices for equipment from which the user can choose.
- Fixed problem with damage displayed for weapons that have maximum damage like 1d+2.

Application Changes

- Added group files, which can list the player characters, NPCs for a scenario, etc., to allow easy printing and filtering of groups of characters.
- Added form size to print templates.
- Allow printing multiple multiple characters on the same sheet to make efficient use of 3" x 5" sheets with multiple cards per sheet.
- Added the script debugger window to make debugging scripts and filters easier.
- Added rule sets, which allow common sets of rules to be shared among groups of character sheets that use the same rule set.
- Added mechanism to indicate that macro arguments are of a particular type: requirements, adjustments, automatic items and categories. The same editing dialogs used to edit properties of items can now be used to edit arguments that represent these properties, relieving the user from having to know the syntactic structure of those arguments.
- Added the `$$argtype` macro command, which allows types of macro arguments to be declared so that the Macro Data Sheet editor can bring up a dialog to edit arguments of that type.
- Use eight-byte floating point numbers (doubles) for floating point computations to increase accuracy.
- Added the Suppress Sublist rule type.
- Added the `$$repeat` macro command, which makes it easy to generate repeated sequences of text in filters and data sheets.
- Track the number of pixels per inch in the character sheet so that when the character sheet is read on another machine with a different pixel resolution the windows can be properly sized.
- The open/closed state of shortcut menu items is remembered.
- Added `%x`, `%X`, `%o` and `%b` formats for `format()` to allow formatting hex, octal and binary numbers.
- Added the `@sub`, `@var` and `@for` filter commands.
- Allow the assign to be omitted in filter assignments.
- Fixed a problem with setting non-integer weights on certain kinds of equipment (weapons, etc).
- Pressing Backspace now goes to parent item in available items and window lists.
- Handle non-breaking hyphen when converting RTF to straight text.
- Class definitions for category maps.
- Added the `@gettext` filter command to accept input from the user during filter operations.
- Added the `@output` filter command to suppress output. This allows more natural formatting in filters when only logic is being executed.
- Fixed a problem that caused items that don't satisfy the active rules to be displayed when searching for an item in the available items list.
- Added the ability to specify the category in `foreach(*List name: Category).`

- Add items to a category map when adding them to the available items list in the data sheet. This allows the `foreach(*listName : category)` construct to be much more efficient—when the category is specified, only items in the category are iterated over. This drastically speeds up generation of characters from templates where items are selected from a category.
- Added the context, and `getitem` command script commands.
- Added the `$+TickCount()` function to conversion scripsions.
- Data files can be included in macro data sheets.
- Fixed a GPF that occurred when editing requirements.
- Fixed problem with finding text in available items list, which hung under certain conditions when the search started on a folder.
- Fixed bug where really long strings in category and other data types caused a crash when editing properties.
- Fixed a problem where non-available items would be found if the items in a list are displayed alphabetically, both in searching and random item selection.
- Added `@origlist` keyword for `@foreach`.
- If an item being added in a conversion script with `add` isn't found under the specified name, also search the data sheets for items with that original name.
- Added no adjustment and no requirement flags to items to turn off item adjustments and requirements checking for that item (and its immediate children if it's a sublist).
- Added a Sublist tab to the item editing property dialog to relieve crowding on the Flags tab for item properties.
- Allow items to be selected from the available items list in the requirements dialog.
- Set the desired level of items added by prerequisites to the level indicated by the initial level (it was left at zero, so if zero was an allowed value, the initial value would be zero instead of the indicated value).
- Changed detail printing to add a workaround to a bug in Windows that caused a hang when the details were printed and they happened to be exactly a page long.
- Changed list sort dialog to allow sublists to be sorted at the end or beginning, in addition to being interspersed among normal items. Allow recursive sorting of sublists. Retain last settings in dialog in the registry so they can be restored.
- Added `datasheet @Filename.dlist`, to allow user to edit the list of data sheets that is loaded when the **Add... | Spells** and other similar shortcuts are used.
- Check the level of items added by a conversion script and ensure that they have a legal value.
- When items with exclusion requirements indicate a level for excluded items, also take the level into account.
- Display an appropriate error if no matching command for `@foreach` is found.
- Use the game system indicated in New File dialog when creating a print template, removing a separate dialog that used to get the game system.
- Allow arguments to the FileNew shortcut menu command to specify file type, game system and template for new files.
- Allow options to be renamed when added via automatic items.
- Added `?qualifier` to allow qualifiers to be set on options added to automatic items.
- Allow **replaceString** replacements to contain `\1`, `\2`, etc., to insert matched text surrounded by `()` into the returned result.
- Added the following values to the item attributes that can be referenced with `@foreach` and similar commands: `@autochargetcost`, `@checkexp`, `@checkfailed`, `@childcost`, `@default`, `@deleteauto`, `@lookuparray`, `@noautodup`, `@optrawtype`, `@optexp`, `@optrawvalue`, `@origlist`, `@sformat`, `@varname`.
- Added the `@exit` and `@message` filter commands.
- Added `@gamesystem` and `@ruleset` to special defines that can be accessed in a character sheet.
- Removed the restriction on the contents of text lines in macro data sheets.
- Added the ability to draw regular polygons in print templates.
- Added the horizontal center and vertical center commands for centering objects in print templates.

- Fixed a problem with using clipboard commands while editing the name of a sublist or category in a macro data sheet.
- In automatic items, values of options are not reset if the new value is smaller than the current value.
- Improved editing of macro argument dialogs.
- Added ?choose, ?list, ?checkexp and ?checkmsg directives for macro argument type definitions.
- In some cases editing the arguments didn't pay attention to the edit cells as text setting.
- Fixed a problem with not going to the next argument if the first argument of a macro was edited with a dialog instead of as text.
- Allow local macros to be defined in macro data sheets.
- Added the **listDirectory** function.
- Added the `menucommand` conversion script command.
- Fixed a bug that prevented an item in another list from being used as to satisfy a requirement for an item with the same name in the current list.
- Fixed editing of adjustments to explicitly deal with reverse adjustments.
- Allow category and class to be specified for macro data sheet sublists.
- Preserve macros found inside macro data sheets.
- Fixed a bug that caused a GPF if you pasted plain text into a print template that had tabs defined for the default formatting.
- Add select item ability to options so that the text of an item can be set by selection in the available items dialog.
- Added change expression capability to list windows so that changes to other variables can be detected and the total prompt area for the list can be properly updated.
- Added no check on insert flag on items.
- Fixed problem with searching for a string in the available items list when all items are displayed alphabetically and the first item is highlighted.
- Eliminated some fixed-length limits in resolving option values.
- Fixed a bug with the focus not being set on the New Option dialog.
- Fixed a needless error message that occurred when you created a new auxiliary cost option in a data sheet.
- Fixed a problem with items excluded by a number of items in a category at a certain level requirement.
- Added the `addoption` command.
- Fixed problem with getting references to local/global variables confused in subroutines when references were made to globals in the assignment part of a `var` command.
- Fixed problem with references to character sheet variables getting mixed up with local variables.
- Eased some length limitations on expansion of variables in conversion scripts.
- Center `getitem` window in conversion scripts.
- Added optional title to `getitem` conversion command.
- Fixed position of header after macro data sheet window was resized.
- Keep top index when changing shortcuts out so that if the shortcuts are similar they don't keep changing what's displayed.
- Fixed bug that caused an array passed into a subroutine not to work properly with listboxes in dialogs.
- Changes to allow complex expressions on rules to function.
- Fixed a bug that caused an illegal script invocation if you changed an edit field that had a script and then executed a shortcut or menu script command.
- Added the `switch`, `keepnext`, `optqualifier`, `copydefines` and `dialogsize` conversion script commands. Added three-argument version of `showreq`.
- Fixed bug that allowed only 19 macro arguments instead of the documented 20.
- Fixed a crash that occurred if you changed an edit field that had a check message that failed.
- Increase the depth of the expression evaluation stack to 40, to allow `qindex` to have more arguments.
- Added requirement rule.

- Fixed problem with searching for a single "/" with strindex.
- Fixed problem with adding strings that contained fractions and numbers: evaluating 1+"1/4" returned 2 instead of 1.25.
- Added the \$adding\$ keyword to adjustments.
- If an item added automatically is added as a sublist is already present and no auto dup is set, use the existing sublist.
- Added 4-argument showreq and checkreq commands.
- Don't include the item itself in the item reference combo box to prevent recursion.
- Added \$\$self to macro processing.
- If the user cancels satisfying requirements in a way that causes an item that has just been added to be rescinded, make sure the item is deleted from the list.
- Increase maximum size of expression component of an option to 512
- Added getmap, putmap, listmap, clearMap, replaceToken, checkRequirements and optNumberValue functions.
- Allow alias references for options with optTextValue.
- Prevent infinite recursion when computing the contribution of options to total adds and multiplies.
- Allow custom editing of items by script.
- Added ?seloptval automatic item directive for letting the user choose the value of options on automatic items.
- Added ctrlsize conversion command for setting the size of controls in script dialogs.
- Specifying a single blank for the the total prompt for a list eliminates the total prompt area below the item list.
- Suppress check expression errors during character sheet loading.
- Added ?checkreq directive for automatic items.
- Added ?cchoose automatic item directive, which displays the cost of the chosen items as they are selected.
- Added rdonly flag for edittext controls in conversion dialogs.
- Allow countItems to count any items, without performing an attribute value comparison.
- Added option for disallowing satisfying items.
- Added optAvailable function.
- Added listOptions(type, name, array) for listing option information into an array.
- Added optOrigname, optrawvalue, optdisplay, optactive, opttype, optrawtype, optkeepvalue.
- Added outputFileNames global variable.
- Ensure that current list item is visible when resizing list window.
- Fixed crash that sometimes occurred when pasting multiple items into a list where the list check expression failed.
- If the list check expression fails, delete all items that were pasted, rather than just the last one.
- Added ability to merge a second data sheet into an existing data sheet.
- Added original name to option data.
- Added macro data sheet merge capability.
- When pasting items into data sheets, paste items into the existing category and sublist hierarchy rather than duplicating it.
- Added @adj to fields that can be obtained through @foreach.
- Added @datasheets to fields that can be accessed in filters.
- Fixed a bug in saving requirements after editing them that caused extra garbage to be saved.
- Retain the requirements on an item during conversion if the old item has them and the data sheet definition does not.
- Added Inherited flag in options to prevent options from being inherited by items in a sublist.
- Fixed optCount() function, which didn't count options on parent items.
- Fixed bug that caused @if checks on strings in filters to fail.
- Fixed problem with list window not being filled in when a node in the tree list of a data sheet was deleted.
- Specify option requirements.

- Added variteminfo function.
- Allow references to other defines inside a define passed into a macro argument.
- Allow multiple categories in inCategory().
- Added the three-argument concat() function.
- Don't execute a script on an automatic item if a script is already executing.
- Fixed problem with conversions on character sheets that preopen windows.
- Fixed a problem with the no-argument @bgcolor command.
- Added the ability to specify default active values for rules define in another rules file. This lets you override the active value of a rule defined in the main rules file for a ruleset defined in a new rules file.
- Remove limit on length of item name used in add conversion command.
- When adjustments of the form "ListName:ItemName"+1 are done, the item's current name will also match, in addition to the variable name and original name.
- Fixed a problem with foreach(*list : category) that allowed one iteration even though there were no items in the specified category.
- Added the ?ncopt directive for automatic items.
- Remove limitation on length of expression results evaluated by \$\$exp.

Changes in Version 2.2

The following is a summary of the changes to **Character Builder** since version 2.1.

Application Changes

- If the user selects the item info command for an item already in the list, make sure the item's keyword is in the help file before going to the help engine.
- Added **totalItemValues** function.
- When adding a new option, also attempt to open a sublist in the options list that is the same as the first category name for the item.
- Apply selection rules when adding items to satisfy a requirement in a category.
- Allow arrays to be passed to subroutines as arguments in command scripts.
- Allow the user to click on the checkbox of an option in an item without having to first put the focus on the checkbox option.
- Fixed a focus problem in the Modify Defines dialog.
- Retain the size of the item dialog if the user resizes the window.
- Added @atDefault to obtain the setting of the Default checkbox on an item
- Validate input and make sure that values greater than 32,767 aren't entered for number and sides of dice.
- When setting the default value in a script, make sure that the resulting level obeys the check expression.
- The first die roll for the best N of M die rolls wasn't being displayed. This problem was fixed and the resulting die rolls are displayed in descending order for ease of use.
- Added ability to sort items by class.
- Fixed a crash that occurred when you Browse the picture library and no character sheet is open to accept the picture.
- If an illegal file name is provided an error occurred when trying to save the file. Fixed this by forcing the file name to be empty and setting the directory to a known location when the error is detected.
- Change fomat() to always format %2d type numbers with a period and no group separators.
- Rand pips change wasn't detected when you edited the properties of an item.
- Made dialog for item category satisfaction resizable.
- Fixed memory leak in printing auxiliary files.
- Don't list items that cause recursion in the "Select n items from a Category" dialog.
- Allow access to qualifier in option display expression.
- Fixed a memory leak in the file printing dialog.
- Set tab stops in edit function dialog.
- Fixed problem with strings that are too large for dupstr to work.

- Allow a configuration parameter to be saved to the registry even if there is no entry in the preferences file for it.
- Fixed problems with searching and pasting into large text files.
- Added @see to information available about items.
- Added ability to display progress dialog in conversion scripts, adding the `showProgress`, `closeProgress`, `closeProgress` and `pauseProgress` conversion commands, plus the `$+canceled` variable.
- Display the game system help when the Help button is pressed in the item editing dialog, instead of the generic help.
- Allow the list to be chosen in the List Summary dialog. Sort the categories in the List Summary dialog.
- Added a three-argument version of the `checkReq` conversion script command that checks requirements on a named item without adding it to the character sheet.
- When converting a character sheet and differences are found, name the new file with a `_new` suffix on the base name instead of changing the extension to `.new`.
- Fixed a problem with moving/pasting sublists whose items inherit options, when those items have bonuses that affect the cost.
- Fixed a problem with `replaceString` and regular expressions anchored at the beginning (starting with the `^` metacharacter).
- Fixed a crash that occurred when requirements were copied and pasted in data sheets being edited by the application.
- Increased size of selection rules dialog.
- Allow references to item information in expression selection rules without having to call `itemInfo`. For example, the expression `@dsid != "` can be used instead of `itemInfo('@dsid@') != "`.
- Added Modify | Font... command to macro data sheets.
- Allow @trans commands in filters to be executed anywhere in the script. This allows different translations to be made at run time based on information in configuration parameters, for example.
- Use binary search to speed up search for internal functions. Other changes to make symbol lookups faster.
- Fix problem with Windows XP not working properly when the help file isn't in the same directory as the application.
- Check for null window handle when making sure that the selection dialog doesn't cover another window.
- Fixed the formatting of numbers using the `%2n` format.
- Fixed a problem in print templates with multiple lines of rotated text.
- Added `$+inSourceList` function to conversion scripts.

GURPS Game System Template Changes

- Added character development dialog and command.
- When optimizing a character attempt to use defaults to reduce character cost. This does not provide the absolute optimal cost, but provides a lower cost.
- Change Weapon Master to affect weapon damage and specify the weapon or skill involved.
- Correct Divination (Belomancy).
- Categorize certain spells as combat and thief
- Let the user skip adding a group of advantages, disads, etc. during character generation through a template.
- Italicize colon after Advantages:, etc.
- Allow zero-level spells. This allows the user to add spells for that aren't yet usable but indicate that time has been spent studying them.
- Added Very Fit.
- Allow Divination Talent to be a "Magery" advantage for fulfilling other spell requirements.
- Changes to allow No Physical Body to correctly display ST, Fatigue and HT values.
- Included Combat Reflexes and any global bonuses in the Active Defense in the Combat Dialog and elsewhere

- Corrected defaults for skills that provide their own cost functions.
- Fixed problem with non-IQ-based mental skills and Eidetic Memory.
- Allow zero-level spells.
- Display extra spell colleges/categories in available spells list if present.
- Display extra weapon damage for Weapon Master.
- Display negative amounts of money correctly.
- Removed adjustment on Sex Appeal for Appearance because it's variable.
- Accounting gets a bonus from Mathematical Ability.
- Added Cost of Living, jobs, time use profiles and other items for character development.
- Added Fanaticism (Self) for Megalomania and require it.
- Change name of No Sense of Smell/Taste.
- Fix weights of suits of armor.
- Added basic combat system armor sets.
- Added Frequency of Submission option for psychological disadvantages based on Will.
- Get bonus from Status at any level for Savoir-Faire.
- Added *Musical category to certain skills to allow categorization for jobs.
- Fixed display of Powerstone cost.
- Include the bonus for Weapon Master in weapon damage.
- Changes for No Physical Body: this specifies that ST must be 0, HT must be 1 and Fatigue is based on IQ.
- Fixed incorrect computation of base move with regard to running.
- Fix incorrect display of impaling DR for hands.
- Fixed problem with converting adds to dice.
- Fixed problem with mental skills at default having less than 1/2 point cost if Eidetic Memory present.
- Change text size to eliminate problem with Disadvantages header in print template
- Changes for very rapid healing, disease resistant to allow them to be added at zero cost by other items.
- High pain threshold fix.
- Fixed misspelling on quirk.
- Added a couple more quirks.
- Don't print armor TL by default.
- Require an argument for displaying parry on martial maneuvers rather than assuming that something with parry in the name should get a parry option.
- Allow format spec on martial maneuver macro.
- Allow omitting wealth level on jobs.
- When selecting random items, don't select items that duplicate or don't satisfy requirements.
- Fix various errors in template generation involving selection of items.
- Minor corrections to Battlesuit Trooper, Berserker, Combat Engineer, Guerilla/Partisan, Knight and Tech Wizard templates.
- Include weight when printing equipment sublists.
- Account for combat reflexes and enhanced parry in martial arts parries.
- Make one-handed versions of katana, bokken and shinai.
- Correct weights of Japanese arrows.
- Rename Upper Body Strength for consistency.
- Include parry bonuses in skill display of Parry.
- Account for parry bonuses in computation of weapon parries displayed on skills.
- Fixed roundoff errors in computing cost of Reputation.
- Include combat reflexes and other parry bonuses in display of Parry on skills and weapons.
- Account for low ST when computing level of Blacksmith skill.
- Fix cost of Lightning super power (changed from a previous edition of the book).
- Account for TL modifications from Primitive and High Technology. Make Primitive and High Technology exclusive of each other.
- Allow category on super advantage macros.

- If equipment weight greater than 1000 lbs don't include in encumbrance.
- Display price of all equipment using standard money display function.
- Change the way derived characteristics work in order to make Will be baseable on 10 or IQ, instead of just IQ.
- Categorize Racial advantages as such.
- Added display of Fright Check to items that modify it.
- Make Fast Draw (Knife from Teeth) be affected by Combat Reflexes.
- Added Methology and changed Hydrology to default to it.
- Corrected miscategorization of Super Disadvantages.
- Fix roundoff problem with very large sums of money.
- Use display money function for wealth
- Add planetology specializations.
- Account for Early Maturation in maturation age.
- Added the three different kinds of Will stats (Emotional, Physical and Mental).
- Fix requirement for legal alternate identity.
- Allow the cost per level to be set for hit points and fatigue when added as advantages and disadvantages.
- Use the common function to display money total for equipment in the print templates.
- Spells that fall into multiple colleges are listed in both colleges, with a \$\$seespell reference macro for the alternate college definition, which points to the actual definition.
- Leprechaun was missing Leatherworking
- Fixed problem with Chameleon not handling bonus correctly for Stealth
- Add translation of certain Windows specific characters to generic representations (fractions, multiply) in filters.
- Remove Inheritopts from Familiar sublist to fix incorrect cost computations when the sublist has modifiers to the cost in the options list
- Allow defaults to be specified for languages. Display other colleges for spells with more than one college.
- Fixed Burning Death, Lengthen Limb, Resist Acid, Seek Magic, Seek Metal and Seek Plastic spells.
- Added character translation for non-Windows destinations for filters.
- Add rules for alternate spell prerequisites.
- Added ALTS data sheet ids to identify spells that use alternate spell prerequisites.
- Added configuration parameter to allow specification of non-Windows character translation for filters.
- Added correction for High Pain Threshold conversion.
- Fixed the way basing Fatigue/Hits on HT/ST works in combination with racial ST and HT bonuses.
- Fixed incorrect comment in thrusting Greatsword.
- Added mental quirks to disadvantages.
- Change styles in RTF filters to conform with SJ Games naming.
- Fixed the one-college magery works.
- Corrected display of small amounts of money.
- Corrected the way fatigue is calculated when racial strength is involved and fatigue is based on HT instead of ST.
- Fixed requirements for Musical Composition.
- Added Constant Telepathic Connection for Duplication.
- Changes for full slate of Empathic skills being modified by Low Empathy.
- Changes for Low Empathy adjusting all Empathic skills
- Change a couple of prompts
- Don't put adjustments on sex appeal directly.
- Added Takes Recharge to Psionics limitations list.
- Added Length of History option to Psychometry.
- If Empathy is present in a character sheet, the first three levels of Telepathy are free.
- If Danger Sense is present, the first five levels of ESP are free.