



Designer's Notes: GURPS Character Builder

by Bruce Kvam

I discovered **GURPS** back in 1989. I'd been playing that other game system, but was tired of the arbitrary limitations and the same old dungeon crawls. I got hold of **GURPS Basic Set** and **GURPS Cliffhangers** and embarked on a pulp campaign redolent of the *Indiana Jones* movies.

One problem, though, was the time involved in creating balanced characters. As a GM, I created a lot of non-player characters and it took hours to get them just right. Yeah, I could have just slapped down a bunch of numbers and not bothered to add them up. But I've always been a by-the-rules kind of guy. It only seemed fair for the GM to play under the same constraints as the players.

Being a professional programmer, I wrote a DOS application that created **GURPS** characters. When I started using Windows, a complete redesign from the ground up was in order.

The primary design consideration was to make the character creation process be driven by the data, rather than the application. This would allow creation of new types of advantages, skills, and powers without having to change the application.

A second, less obvious, implication was that other game systems could be supported. Taking a cue from **GURPS**, the design was generic, allowing the application to encompass other systems, eventually including *Fudge*, *CORPS*, *the Hero System*, *Fuzion*, *Call of Cthulhu*, *Legend of the Five Rings*, *Sengoku* and others.

Character Builder Features

Character Builder creates characters from the character templates in **GURPS Wizards** and **GURPS Warriors**, or from completely from scratch. Characters are based on character sheet templates, which provide a starting point for the character: the TL, beginning character points, other campaign limits, the data sheets that are loaded (essentially, which books you'll be using), etc. You can also make your own character sheet templates (which is the case for just about everything in **Character Builder**), setting your own initial TL, beginning points, maximum points in disadvantages, even the initial window layout.

As you change your character's attributes, the character point totals are automatically updated to reflect the cost. Skills whose levels depend on those attributes are also changed; during character creation, skill costs are adjusted automatically to keep the level constant (if possible). Totals are immediately updated.

When you indicate a skill is based on the default, the highest value is taken, be it a default based on a specific skill, an attribute, or an entire category of skills.

When you add an advantage that gives a bonus to a skill, the skill's cost is reduced to account for the bonus; if the skill's level is reduced below the minimum allowed, the level is set to that minimum. The original value is remembered, however, so if you crank IQ up to 17 just to see what happens, **Character Builder** will reset the skill levels to the original values when you bring IQ back down to 12.

Character Builder performs semi-automatic spell prerequisite satisfaction. For example, if you pick the Deathtouch spell, the application tells you that Wither Limb is required. If you tell the application to satisfy that requirement, it tells you that Magery 2 is required. After you pick the desired type of Magery, it tells you that Paralyze Limb requires five Body Control spells. If you pick Balance, the application will automatically add the spells required for Balance (which successively require Boost Dexterity, Dexterity, Clumsiness, Spasm, and Itch), which also happen to satisfy the five-spell requirement for Paralyze Limb. You only need to pick a few spells and the application will do a lot of the grunt work for you.

The same requirements mechanism is used for skill prerequisites and conflicting advantages and disadvantages (such as the same character having both an Appearance advantage and disadvantage).

Character Builder also implements selection rules; these suppress the display of skills and equipment that are above the character's TL so that you don't have to wade through the Astrogration and Beam Weapons skills when you're creating a medieval character. You can always turn these rules off with a click of a checkbox to see everything that's available.

If your campaign doesn't enforce all the standard **GURPS** rules, you can turn off the requirements and warnings.

And for busy GMs who need a couple of quick NPCs, you can put the Generate Character script on autopilot and have it generate a character randomly. Usually I despise random characters, but since this is template-driven, there's a lot less cleanup required than for typical randomly generated characters.

What's All in Character Builder?

GURPS Character Builder is big. It includes the data from **GURPS Basic Set**, **GURPS Magic**, **GURPS Supers**, **GURPS Grimoire**, **GURPS Martial Arts**, **GURPS Compendium I**, **GURPS Psionics**, **GURPS Wizards**, and **GURPS Warriors**. This all adds up to about 800 skills and maneuvers, 60 martial arts styles, 1000 spells, 500 advantages, 300 disadvantages, 160 powers, and 400 items of equipment.

There are about 200 high-resolution character art images, including 36 character outlines that you can base your own character pictures on, and 160 complete character illustrations from many **GURPS** books and other sources. These illustrations are categorized by keywords that you can search on (and you can add your own images and descriptions to the picture library).

There are four graphical character sheet print templates: the two-page **GURPS** character sheet, the one-page character sheet, the spell grimoire, and the martial arts record sheet. You can also create your own (see below).

Using text filters you can print out three different types of GM Control sheets with combat information for multiple characters. This and other character information can be put into RTF (Rich Text Format) files that can be edited further by word processing applications. Similar text filters can be used to copy character descriptions in three different styles (including the standard SJ Games character description style) to the clipboard for incorporation of character writeups into word processing documents.

There are three special commands, including a "finalize character" command (prepares your character for post-creation attribute costs and skill level rules), a "generate character" command (creates a character from a character template and optionally generates a random appearance and selects random advantages, skills and disadvantages according to the template), and a "character optimization" command that finds the values of IQ and DX that produce the lowest character cost.

There's a die roller (which is customizable -- you can associate any die rolls you want with the buttons) and a name finder (which lets you search a database of thousands of names, many categorized by sex and ethnicity).

Finally, there's a collection of useful **GURPS** PDF files, including several different character sheets, spells summaries, various campaign sheets, hex grids and other forms.

Customization and House Rules

There are several built-in customization and configuration features, including setting the cost of extra fatigue and hit points, converting adds to dice, the Enhanced ST rule and basing Fatigue on HT and Hits on ST. You can also change the costs of most standard items to match the requirements of your own campaign, or modify "generic" skills and advantages to create one-shot specials.

You can create your own data sheets to add campaign-specific skills, advantages and disadvantages. How hard is it to do this? Well, like everything, if what you're trying to model is simple, the solution is simple. If it's hard, the solution is hard.

Let's say your Old West campaign needs a Calf-Roping skill, a PA skill that defaults to DX-5 and requires Riding and Animal Handling. The following shows how this skill would be entered in the data sheet (you enter the data on the second line):

Macro	name	class	cat	def	prereq
skill	Calf-Roping	PA	Animal	DX-5	Riding>=12;Animal Handling>=12

Data entry is accomplished through a spreadsheet-like interface, with rows and columns. You simply enter data in one column and tab to the next.

Advantages, disadvantages, spells, weapons, equipment, etc., are entered in a similar fashion. The most complicated items to create are the character templates, because of specializations, variations based on TL and the like.

The character finalization, generation and optimization commands are examples of character sheet automation: scripts that directly manipulate the character sheet. Character Builder has a script programming language allows you to change values in the character sheet, add skills, set options on advantages, etc.

The ability to customize doesn't end there, however. **Character Builder's** multi-system heritage remains intact. The application provides all the tools to create custom **GURPS** character sheet and print templates; if you don't like the way the standard printed character sheet is laid out, or it's missing some data that you want to include, you can create your own print template.

Character Builder can also create completely new game systems. Scripts can be written to convert characters from one game system to another.

The Guts

Internally, the application is based on an expression evaluation engine that "notifies" a component of the character sheet (an item in the skill list, an attribute in the Main dialog, an active defense value displayed in the Combat dialog, etc.) when the variable associated with that component is changed.

For example, when you change your character's IQ, all skills that reference the IQ variable (or reference another variable that references IQ) are reevaluated and their new costs displayed. This is similar to the way spreadsheets work, but instead of cell references, **Character Builder** uses named variables. The result is that skills that don't depend on IQ won't be updated, giving faster response time.

Similarly, if you change the level of Magery all spell costs are reevaluated and displayed. Changing ST will cause hand weapons to be reevaluated and the new damage value displayed. If Fatigue is left "floating" (defaulting to ST), Fatigue will also automatically track ST. When you increase Linguistics, all language skill costs are reevaluated for the Linguistics bonus.

Finally -- and I'm not really sure why people always ask this -- the application is written in C++.

Article publication date: December 28, 2001

273 *Pyramid* subscribers rated this article **3.36** on a scale of 1 to 5. Visit the [ratings page](#) for more info.

Copyright © 2001 by [Steve Jackson Games](#). All rights reserved. Pyramid subscribers are permitted to read this article online, or download it and print out a single hardcopy for personal use. Copying this text to any other online system or BBS, or making more than one hardcopy, is *strictly prohibited*. So please don't. And if you encounter copies of this article elsewhere on the web, please report it to webmaster@sjgames.com.



[Home](#) - [Subscribe!](#) - [Current Issue](#) - [Playtesting](#) - [Chat](#) - - - [Feedback](#)