

Release Notes for Metacreator™ 4.1.1

Thank you for installing Metacreator.

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 Metacreator news, see the Alter Ego Software website at <http://www.alteregosoftware.com/>.

Multiple Versions

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

Updating Character Sheets from Previous Versions

When you run the new Metacreator 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 module 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 Metacreator Reference Manual for details on updating your character sheets.

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. Metacreator 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 Metacreator 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).

If You Edited the Data Sheet by Hand

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

```
level 2
```

Look at the new `.cds` file and find the corresponding line. 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 Metacreator

You probably have a lot of work ahead of you. Items are usually defined using macros. When you save a data sheet with Metacreator 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 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 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 (2) is different from that of the print template (3). 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 *Metacreator* around for compatibility with other users, or for evaluation purposes, you can install the application in a different directory. Simply choose the target desired 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 `Metacreator.ini` in the directory where you installed the new application (the source directory).

- Add the following text (assuming that you installed to the folder C:\Program Files\New Metacreator):

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

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

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

Metacreator Changes

Changes since version 4.0.5

- When converting character sheets extra named dialog fields are now retained. If the new character sheet is based on another template, the dialog size and location of fields are copied from the old character sheet. These changes allow character sheets to have extra fields yet still be based on the same template.
- Text and groupbox controls in dialogs can now be named. This indicates that they should be copied during conversion and also allows for the displayed label to vary between character sheets based on the same template.
- Added the `recalcfiles` conversion script command.
- When configuration parameters are changed the variables in all open character sheets for the current game system are recalculated to reflect the changes.
- Default rule set values for existing configuration parameters can be specified in additional configuration preference files. Additional values can be specified for dropdown parameter types.
- Added **listExists** and **dialogFieldExists** functions.
- Added `$+optavailable()` function for conversion scripts.
- Search for item information based on displayed name first, then the original name. This allows information about items to be added to the help without having to create a unique item.
- Change to **listOptions**: the option alias will match.

Changes since version 3.9.7

- Added the concept of option sets, which allow an expandable list of name/value pairs for the values of options. This is used for defining the range, duration and target options of **Ars Magica** spells, enabling arbitrary values to be added in future supplements and by users.
- Added the ability to set the default extension for a character sheet file so that "character" files can have extensions other than `chr`.
- Added the ability to provide extension descriptions in the game system information file, to work in conjunction with the default extension for a character sheet.
- Added the **optSetValue** and **optSetNames** predefined functions.
- Added the ability to detect options in a category with **optPresent**.
- Added the ability to indirectly call subroutines by allowing conversion script replacement for a subroutine name.
- If the character sheet selected in the **File | Open...** dialog is already open in **Metacreator**, you will be asked if you want to activate the open copy instead.
- Added **sublistPresent** function.
- Made the category checking function of **optPresent** work for multiple categories.
- Menu items are now added to the Data menu (and any submenus) in alphabetical order.
- Load data sheets listed by a data sheet before loading the data sheet itself.
- Fixed a problem with editing binary data sheets: changes made to certain option fields were not detected and saved. Memory leaks in option editing also corrected.
- Allow the use of HTML Help Workshop (`.CHM`) files for Info help.

Changes since version 3.8.7

- Added `optCategory` to **listOptions**.
- Changed the behavior of **getFileValue** to read current values from open character sheets, rather than from the saved file on disk. This allows you to make changes to a laboratory and a character sheet and interact with both without having to constantly save the files.
- Added the `opencharsheet` script command to open a character sheet and execute a script.
- Added the ability to specify the source category for an item added with the `add` command in a script.
- Item selection for option values was enhanced to allow selecting a list of items. References to item values and options can now be made in the item selection format.
- Assign an automatic item ID to an item if any of the automatic item flags are set.
- References to the current character sheet's data can be made in expressions in the item selection format through `eval`.

- If a macro argument has the literal type no processing is done on its value.
- Allow wild cards in the include statement of a shortcut (.sct) file.
- Added **File | Read Details...** command to the menu for the Details window.
- Allow arguments to be specified for a `menucommand` conversion command.
- Added **FileReadDetails** menu command for shortcuts.
- Allow a print template with no generated pages to print if there are details to print.
- By default the last filter used is highlighted the next time you use a filter. If the current sheet type is different, the filter specified in the file is highlighted instead.
- Fixed a problem with the game system path not being searched when printing associated files.
- Added `createcharsheet` command.
- Added `sheettypes` command.
- Improved the automatic highlighting of conversion scripts so that the best choice is preselected for the user.
- If the value and cost of a ?? option are the same except for the presence of a '+', just print the value.
- Don't copy shortcut file, print templates, etc., when converting a file from one sheet type to another.
- When an associated file is in a subdirectory of the directory where the file is stored, maintain a relative path to it.
- Fixed problem with occasional junk showing up in file name dialog for `getfilename`.
- Fixed problem with directory names with blanks in the SearchPath.
- The `creatorh.dll` is now installed in the Metacreator directory instead of the Windows directory on Windows XP and later. This allows regular user accounts to install the application.

Changes since version 3.8.6

- Fixed problem with game system not showing up if the help file wasn't set in the game system info file.
- Fixed problem with `@lastitemadded` keyword.

Changes since version 3.8.2

- Menus from included shortcut files are merged into a single hierarchy if the menus have the same name.
- Added include command for shortcut files.
- Submenus with the same names are merged into the same hierarchy on the **Data** menu.
- Retain the original shortcut file name when updating a character sheet.
- Added ability to get help for game system supplements on help menu.

Changes since version 3.7.1

- Added custom help files, which allow users to add HTML help and information links to their own game systems, or extend existing game systems.
- Ignore articles when sorting item lists. This means in **Ars Magica**, for example, spells that begin with "The" will be sorted by the second word instead of by "The".
- Added stress dice to the die roller.
- If no items are specified for `?setitemname`, the user is prompted for the name.
- Allow `@@...@@` construct in qualifier values in automatic items.
- Allow `if ... elseif ... else ... endif` constructs within the body of a dialog in command scripts.
- Provide method to display a tool tip window when the mouse cursor hovers over a formula control in a character sheet dialog.
- Allow greater control over how skins in character sheet dialogs are tiled and stretched.
- New functions: **qlookup**, **textSectionExists**.
- **listOptions** now takes an "optQualifier" argument.
- Allow listboxes in character sheet dialogs to have no border.
- Added the `@spaceabove` filter command.
- Increased maximum number of arguments to functions to 20.

- Allow single value listboxes in script dialogs to disable the default button with `numselections 1, 1`.
- Allow selection changes in script dialog comboboxes, listboxes and checkboxes to be detected in `displayexp`.
- Improved the handling of file name extensions when saving text files.
- Added `@listopen`, `@lastitemadded` and `@currentitem`.
- Fixed a recursion problem with certain requirements.
- Added the `optionitemref` conversion script command.
- Fixed bug in `selectitem` conversion command.
- Added `%. . . %` construct to force display in blank character sheet templates.
- Getting info on a reference item now gets the info on the base item.
- Substantially increase limit on \$\$ references to variables in conversion scripts.

Changes since version 3.7.0

- Fixed a problem with the `substr` function with non-integer third argument.

Changes since version 3.6.0

- Print templates can be scaled to fit exactly in the printable area of the printer.
- Added three-argument version of `strindex`.
- Conditional submenu items in scripts. This can be used to prevent the display of menu items that aren't applicable to certain kinds of character sheets. Used in 3.5 to control display of commands that aren't appropriate for Modern and Fantasy.
- Added `?defitemval` automatic item directive to set values of items selected through `?choose` directives. Used in Savage Worlds to set value of Super Karma items.
- Added delimiter-based form of the **substr** function.
- Search for game system help in main game system help file if help not found in help file specified for a character sheet.
- Retain the help file for the source character sheet when copying over character sheet info during conversion.

Changes since version 3.5.0

- Zoomable text in print templates.
- Fixed problem with arguments that are surrounded by quotes being passed to a `$$repeat`.
- Fixed a problem with see items showing up when they shouldn't. When a see item is found, the item it points to is used for resolving the requirements instead.
- Enable the Roll button when doing a die roll command from a script.
- Improve error messages displayed when you attempt to satisfy all requirements and some cannot be satisfied.
- **ListItemInfo** and **optValue** now look at the default data sheet lists before other data sheets.
- F1 gets game system help for Group Files if the "Group Files" help keyword is present in the game system help file.
- Fixed a problem with short cut files not being found in some cases when a game system name-based shortcut file wasn't found in the source directory (but was located in the search path).
- Fixed a problem that caused automatic items to fail to add items if the window wasn't open and no target list directive was specified.
- If recursive requirements defined, check to see if the item we're looking for is already in the list of items having their requirements defined. If so, assume that it will be added, and don't bother to issue a message -- just let it go through.
- If an item is found for a requirement run the selection rules on it, and don't satisfy requirements for items that aren't selectable.
- Plot nothing if the picture reference expression resolves to the empty string in a print template.
- Asterisk wasn't always displayed for unsatisfied requirements after attempt to satisfy them.
- Fixed a problem where see items found an item that didn't satisfy the rules, so you couldn't add the item. Now subsequent items with the same name are examined till one that satisfies the

requirements is located. Note that this only works if the duplicate items are sequential -- if duplicates are located in different folders this will not necessarily find the desired item.

- If an item has the same name as another in the available items list, the secondary key is the dsID. This makes the data sheet with the blank dsID come first, providing a stable, predictable sort, and making items with no dsID get chosen by default when no duplicates rules click in when no item is defined for the preferred data sheet.
- If an item is defined in data sheet A, and duplicated in data sheet B, neither of them showed up if a duplicates only rule existed and none of the items satisfied the rule. While strictly correct, it's only reasonable that at least one of the items shows up. The first one found will not be displayed.
- Added ?listopt to macro argtype possibilities.
- Added the display expression to items to allow things such as arbitrary die types to be displayed for item levels.
- Added @optalias and @optorigname.
- Display newer types of JPEG images.
- Prevent duplicates in macro selection lists when editing data sheets.
- Added runitemscript conversion command.
- Allow empty adjustment clauses.
- Fixed a problem that caused **incategory** to ignore categories on options if there were no categories on the item itself.
- Fixed a problem that caused the same requirement to be returned twice from a data store.
- Fixed crash that occurred when you tried to insert a column and an option was present.
- Check whether the user changed the printer resolution in the print dialog, and if so, regenerate the pages as is done when the user changes the printer.
- Fixed a problem with adjustments being incorrectly set if you automatically satisfied a requirement by choosing from a list of items.

Changes Since Version 3.3.1

- Random name generation was added to the Name Finder. Rules for dwarf, elf, gnome, halfling and orc name generation are provided. Rules for random "archaic" names and inn and tavern names are also provided.
- Additional names for several nationalities were added.
- Added **Documents**, **Favorites** and **Add to Favorites** buttons to the Open and Save As dialog boxes.
- The Load Data Sheet dialog can be limited to display only those data sheets for the selected game system.
- Always display references to other items ("see" items) in the available items dialog.
- Allow game system files (character sheets, data sheets, filters, etc.) to appear in subdirectories of the source directory. This eliminates the need for prefixing all file names for a game system, allowing shorter and more user-friendly file names.
- Regular expression searching in macro data sheets.
- Limit search in data sheets to a column.
- Add and delete columns in a data sheet.
- Search and replace in data sheets.
- When the number of items in a list is limited to a certain range (1-3) in a script dialog, the number of items selected and the number required are displayed and updated immediately to make it apparent when selection is complete.
- Allow unmatched @ to appear in iteminfo strings.
- Fixed problem with resizing the auxiliary file print dialog.
- Clean up unused keys in the registry when breakpoints are all deleted for a script.
- Automatic item IDs are assigned to automatic items even when the `addnoauto` command is used.
- Allow zero length arrays to be defined in `.def` files.
- Added `autoconversion` and `numselections` commands to scripting.
- `@foreach` in filters can be nested.

- Flag an error if an attempt is made to modify an item in the system data sheet if the conversion script command is executed in the context of the available item list.
- Allow `set class` to change the class of the current item in a conversion script.
- Allow `autoitems` command to execute when the target window is closed.
- Display item references in the automatic item selection command.
- The `?clear` directive does not clear an `?if`.
- When executed in the context of an automatic item, `optPresent`, `itemInfo` and other item-related functions will refer to the parent item (the item adding automatic items).
- Added the ability to check for absence of categories in `inCategory` with `!"`.
- When automatically converting a dialog, have the new control keep the new control's default value if the old control's actual value was the same as the old control's default value.
- Allow controls in character sheet dialogs to be stretched horizontally and vertically. If a dialog contains any stretchable controls, it can be resized.
- Resize option dialog horizontally.
- Added Info button to choose item dialog so that you can get information about items when selecting automatic items.
- Added Info button to item editing dialog to allow you to get information about items while editing them.
- Added the `renamefile`, `updownbuttons` and `displayexp` conversion script commands.
- Don't complain when a character sheet tries to load a data sheet that was already found missing.
- Added `optiondisplayoutput` command to conversion scripts.
- Make Info command work in lists in dialogs.
- `foreach(@List Name)` loops through items in the destination character sheet in a conversion script (without the `@`, it loops on the source character sheet's list).
- Optimizations to speed up data sheet loading.
- Optimizations to speed up loading character sheets with lots of interrelations between different lists.
- Use the help keyword for the list when the user clicks the Help button in the item dialog -- don't assume that the list name is the help keyword.
- Fixed problem with controls being missized when you click the Options button in an item that has no options.
- Added the ability to indicate gaps in the rule lines in print templates.
- Added the ability to check for presence in one category or another with `|`.
- Added the `defineArray` function to dynamically define arrays.
- Added the `?applyrules` directive for automatic items.
- Added `$$$@applyRules` flag.
- Added `insertElement` function to insert elements into the middle of an array.
- Changed `countItems` to be able to count items in the available items list.
- Added `NFiles` and `FileNameX` global variables.
- Increased maximum size of option text to 2000.
- Added the `optdisplayvalue` function.
- Added separate name spaces for configuration parameters based on the active rule set.
- Added a Window Names tab to the data sheet information window to allow the names for windows in the requirements dialog to be populated with the correct names of windows in the character sheet, if the window names are different from the category names in the data sheet.

Changes Since Version 3.2.5

- Added a five-argument form of `listOptions`.
- Added functions: `getFilepath`, `extractText`
- Fixed a problem with distributed level adjustments that caused additions to get added every time you closed the option dialog. This fixed a problem with the Hero 5 Armor power.
- Fixed a problem with non-integer adjustments not getting undone properly. This fixed a problem with Ars Magica armor load values.
- Added `@ancestors` keyword for getting information from item.
- Added two-argument `parentItemInfo` function.

- Added four-argument **getFileName** conversion command.
- When drawing Fudge dice in the die roller, display a blank instead of 0.
- Added Item Info command to list popup menu.
- Added the `showhelp` conversion command.
- Added the ability to indicate that a list window should not check requirements when items are added to it. This was added to allow Ars Magica covenants to share spell definitions with character sheets.
- Use '/' instead of '\' for path separator to fix problems with \ appearing in file paths in variables set by scripts in character sheets. Used in ArM.
- Fixed a problem with the formatting not carrying over properly from one page to the next when printing through filters. Noticed this in the d20 combat summary.
- Fixed a problem that caused a crash that appeared during printing under Windows 98.
- Added the `deleteoption`, `addnoauto` and `changeexp` conversion script commands.
- Added the `@nextlink` filter command.
- Added the separator type for `@hline`.
- Allow numbers greater than 32,767 for **rand()**.
- Allow -1 to be specified for `convlevel` conversion command.
- Added two-argument **rolldice()** function.
- Fixed a problem with executing `edititem` script command inside a custom item script. This surfaced in the Windfall feat for d20.
- Give the user a chance to continue or cancel when ESC is pressed on an automatic item dialog while a script is running. Furthermore, no error message dialog is brought up if the user cancels automatic item dialogs when a script is running.
- Display quotes on edit variable controls that have string values.