



Web-based Zygodact Serial Keys

Server-based generation of Zygodact registration keys using on-rev technology

The CGI script provided with Zygodact used to require PHP or another server-side language to send a request and retrieve the LiveCode-generated serial key from the CGI script. With an on-rev server, LiveCode scripts can be used to do the same thing.

Here we show how to present a serial key to the user inside the same page that contains the form. (If you aren't using Zygodact, the general concepts are described at the bottom of this page.) You could also change the LiveCode segment to use sendmail to generate an email instead, or combine both methods so that the user sees the key on the web page and also receives an email with the information.

This page generates keys for a generic example standalone.

ENTER YOUR NAME:

Your name:

Generate serial key



To implement this web-based system, do this:

Copy the keygen.cgi file and the gen.rev stack from your application's Zygodact set into the cgi-bin folder on your on-rev server. Set the permissions of both files to 755.

You will also need to install a copy of Rev (the former name of LiveCode) into the cgi-bin folder. You can use almost any version prior to version 4.0, which changed the way the engine works. Rev version 3.5 is fine. Copy the file named "Standalone" from the Linux folder which is located inside the Runtime folder in the main program folder. Set its permissions on the server to 755. Change its name to "revolution" on the server.

Now create the irev web page that will ask for a user ID. We're using their name. For this you need a form. The form's action should post the data back to the same irev page. For example, the form on this page looks like this:

```
<FORM METHOD=POST
  ACTION="http://jacque.on-rev.com/zygodemo/getkey.irev">
  <table>
    <tr>
      <td>Your name:</td>
      <td><INPUT TYPE="TEXT" NAME="name" SIZE="25"></td>
    </tr>

    <tr>
      <td> </td>
      <td><input TYPE="SUBMIT" VALUE="Generate serial key"></td>
    </tr>
  </table>
</FORM>
```

Somewhere below the form, wherever you want the serial key information to appear, add a LiveCode segment like this:

```
<?rev
  put $_POST["name"] into tName
  if tName <> "" then
    put "Your ID and key:" & "<p>" & tName & "<br>"
    post tName to url "http://jacque.on-rev.com/cgi-bin/keygen.cgi"
    put it && the result & "</p>"
  end if
?>
```

The irev page should be stored somewhere inside your http_public folder on the server, just like any other web page on your site.



How it works

The form posts data back to the same irev page it started from. The LiveCode segment gets the user name from the "name" key of the \$_POST array, and if the name is not empty, it posts the name to the URL of the keygen.cgi script on the server. The keygen.cgi script creates a serial key as usual (for Zygodact, it does this by loading the gen.rev stack as a library) and returns the new key to the LiveCode segment in the variable it. The LiveCode segment then puts the user name and the retrieved serial key onto this page.

The reason to put both it (which contains the serial key returned from the cgi) as well as the result is because one or the other will contain a value, but not both. By putting both pieces of information onto the web page, either the generated serial key or an error message will appear on the line. This can help to identify any problems.

Caveats

This page isn't secure. Anyone who has the URL can generate serial keys. You'll want to protect the page with a log-in or by some other means.

One way to handle security would be to link the form's submit button to a payment system. Only after a confirmed payment was made would your page call the cgi script to generate a key.

Generalizing...

This example uses Zygodact's universal CGI script, but the same method can be used for any older CGI you may have. The original keygen.cgi script did not have to be changed at all to work with this page. It receives requests and returns a result the same way regardless of whether it is called from a web page, a stack, or a standalone. If you have working CGI scripts on your server already, you won't have to change them.

The main concept here is that an irev page can post data to a CGI script in the cgi-bin folder and receive the response in real time. It's a convenient method if you have currently working CGIs that you don't want to rewrite. It's also a good way to use a protected library privately when you don't want it to be accessible from the web. The CGI script itself and any stacks it uses in the cgi-bin folder can't be downloaded or even identified in a directory listing.

For more information on creating CGI scripts using the older method, see the tutorial on my HyperActive Software web site [here](#).

Code Bits

- [On-Rev CGI example](#): A rotating image display that uses very little code, thanks to iRev scripting. Gives you a look at my own garden flowers too.
- [CGIs in irev pages](#): Combine old-style Rev CGIs with on-rev scripts.

Games

- [MadLibs](#) Make a crazy story. You never know what you'll get. Madlibs is an experiment using posted data within an iRev script.

Jacque's pen

- [How to say "Jacque"](#)
- [Krill: the real story](#)
- [A Modern Guide to Child Rearing](#)
- [The MIDI Days](#)
- [Steve Jobs Framed](#)

The birds

- [Casey and Sierra](#) — the parrots.
- [Software for bird lovers](#)