

# iVend Installation and Administration

*iVend version 1.0.30 of 13 April 1999*

*coding by Bill Welliver <[hww3@riverweb.com](mailto:hww3@riverweb.com)>*

## Table of Contents

What is iVend?.....	2
What do I need for iVend to work?.....	3
A Note About Compatibility.....	4
Installation.....	5
Setting Up a New Store.....	7
Customizing your store.....	9
Customizing the Database.....	9
The Checkout Process.....	10
Appendix A: Configuration Interface Settings.....	11
Appendix B: IVML Markup Tags.....	12
Checkout Tags.....	14
Other Stuff.....	15
Appendix C: iVend Store Files.....	16
Email Notification.....	16
The iVend Main Index.....	16
Appendix F: Notes about data security.....	18

## ***What is iVend?***

---

iVend is a module for the Roxen Challenger web server which allows electronic commerce functions to be provided quickly and easily. iVend supports multiple stores from a single installation, and provides direct support for SQL databases to handle product and order tracking.

iVend is currently in the early stages of development, and as such there will be bugs and missing features. Hopefully with the support of alpha and beta testers iVend will grow up to be a mature, powerful and useful tool.

## ***What do I need for iVend to work?***

---

- A system running Roxen Challenger version 1.2 or higher with Crypto enabled. See the section *A Note About Compatibility* below. iVend will not run without Crypto (you shouldn't even think about doing electronic commerce without crypto anyhow.)
- A copy of Pike in your PATH for running the setup scripts.
- An SQL database server supported by Pike and Roxen Challenger. (iVend was developed using MySQL 3.22, however other dbms systems should work as well with minimal changes to the code.)
- Administrative rights to the database system, as well as Roxen.
- A working knowledge of HTML, SQL databases, and Unix.

## ***A Note About Compatibility***

---

iVend has been tested with Roxen releases 1.2 final and higher. Because of added security features in iVend, certain releases of Roxen and Pike are missing functions that are required to function.

If you are using iVend with Roxen release 1.2, you will need to update your Roxen distribution by replacing the file `RSA.pmod`. This file is located in your Roxen server directory, under `server/lib/pike/modules/Standards.pmod/PKCS.pmod`. Replace it with a copy of `RSA.pmod` available from the patches directory at the same location you downloaded iVend. Once you replace that file, restart Roxen, and you'll be all set!

You will also need to upgrade the same file in any other installations of Pike on your system. Releases 1.3.30 and higher do not require any patches to the Roxen base release.

- Patch download site: <http://hww3.riverweb.com/dist/patches>

iVend was written tested to work with the MySQL3.22 database system. There is very little reason that another RDBMS would not work with minor modifications. The setup wizard does require that your chosen RDBMS support the `GRANT` statement, such as MySQL3.22 or higher. Your mileage may vary. If you do try another RDBMS, please let me know of your results.

## Installation

---

Installing iVend is a relatively painless process. In order to unpack the distribution, you must have `gzip` installed on your system. `gzip` may be downloaded from any GNU FTP mirror site.

1. Unpack the iVend distribution.

If you have GNU tar installed on your system, use:

```
tar xzvf ivend-1.0.XX.tar.gz (XX is the build number)
```

If you don't have GNU tar installed, you should use this command:

```
zcat ivend-1.0.XX.tar.gz | tar xvf -
```

This should give you a directory called `ivend/`. We'll call this directory the iVend distribution directory. Do not move or rename files in this directory. For example, if you unpacked the distribution file under `/usr/local/roxen`, your iVend distribution directory would be `/usr/local/roxen/ivend`.

2. Make sure that the "src" directory within the iVend distribution directory is in Roxen's module search path, located in the "Global Variables" section of the Roxen Configuration Interface. For the example above, you'd add `/usr/local/roxen/ivend/src` to your module search path.

3. Add the iVend 1.0 module to a virtual server.

4. Be sure to set all the directories in the iVend section of the Roxen Configuration Interface. If you don't provide the correct directory information, iVend will not work properly:

- *iVend Root Directory*: This is the iVend distribution directory.
- *iVend Config Directory*: This is usually `ivend/configurations`.
- *iVend Data Directory*: This is usually `ivend/data`.
- *Mount Path*: this is where iVend will be located within your server's virtual filesystem. The default is `/ivend/`. You can change this to anything you'd like.

You can have multiple instances of iVend running on the same server by giving each instance it's own configuration directory. By doing this, you can have iVend running on multiple virtual servers without conflict.

To have iVend stores share the same configuration, such as having a store duplicated on an SSL3 virtual server for checkout, see the section below titled "Setting Up a New Store".

5. Assign values for any other configuration variables such as config user and password.

6. Save your changes.

Congratulations! You should now have a functional iVend installation. At this point, you'll probably want to start adding stores. See the section "Setting Up a New Store" below for more information.

## Setting Up a New Store

---

For each store you set up you'll need the following:

Database administrator username and password. The iVend setup wizard will automatically create the store database and setup an administrative user for you. For iVend to be able to do this, you must have a db account which has permissions to create databases and database users.

### IMPORTANT NOTE

You must have a good grasp of db administration and security techniques before you can properly set up iVend in a production environment. Since you will be dealing with sensitive information, please take this area of store setup very seriously. Helpful tips for new MySQL users are located in *Appendix E: Tips for New MySQL Users*.

Please note that iVend uses database access permissions to determine who has permission to administer your iVend store. Because of this, make sure that you don't grant access permission to this database to anyone that you would not allow store admin privileges. By default, MySQL installs global permissions to a passwordless user called root. This means that unless you change the default access configuration by adding a password for root, anyone can access any database (and therefore any store administration area just by logging in as "root" without a password. This feature also allows multiple logins with access to a store, as well as read-only access to store admin features. While this may seem more complex at first, it allows much more flexible setup of store administrators. Be sure you know who (or what) has access to your databases at all times.

Use the iVend Configuration interface to add a new store. There is a link to the iVend Configuration interface under the module in Roxen's Configuration interface. You will be prompted for the administrator username and password you provided when you added the module to Roxen. Click on "*New Configuration*." The process of adding a new store is wizard based and should be fairly straightforward.

- *Store Name*: A short (25 characters or less) description of your store.
- *Store ID*: This is the unique identifier for this store. This is one word, alphanumeric characters are allowed. If you have numerous stores, this store will be mounted on the directory /mountpoint/id.
- *Database Host*: Host your database is located on. Usually this is localhost.
- *Database User / Password*: Username and password which has administrative rights

for the database server.

- *Store Root*: This is the location where your store's data files will be housed. This directory does not need to exist, as the Wizard will create it.
- *Session Timeout*: This value determines how long a shopping cart will remain active before it is deleted. The timeout is a value in seconds.
- *Store Style*: iVend comes with a set of layouts to use as a starting point for your new store. Select one of these layouts and decide whether to copy the template files, which is recommended.

After you provide the information, the wizard will let you know what it's about to do. The wizard will create and populate the store directory and set up the proper tables in your database.

After the wizard completes this work, you will be given a store administration username and password, as well as the location of your store's data files.

You should be presented with a list of active stores. Click on "Save Configurations." Your new store should appear in the list.

If this is the first time you have entered the configuration interface, you should also review the settings under the "Global" tab. After you've done this, click on the "modify settings" button, then save the changes to your global configurations using the "save changes" link.

For more information about store and global settings, see *Appendix A*.

At this point, you can enter the new store's administration interface. The administration interface is located at `/mountpoint/storeid/admin`. You'll be prompted for a username and password which has access to the store's SQL database. If you have a properly set up database server, you should use the information given to you by the New Store Wizard.

From this interface, you can add, delete, and modify products and groups; set up shipping charges and rules; set up sales tax charges; as well as view and process incoming orders.

If you want to use another program to add or manage your products, you have a few options: ODBC, exported data, or other client interfaces. ODBC allows compliant programs to access data from other database systems. You can use ODBC to access your store's products and orders through programs such as MS Access, Excel and so forth. You can also find scripts in the scripts directory that will export and import tab delimited text files from iVend. If you decide to "roll your own" administration interface, be sure to send a copy to [hww3@riverweb.com](mailto:hww3@riverweb.com) so I can see what other ways are being used.



## ***Customizing your store***

---

### **Customizing the Database**

After you create the store, you can use standard SQL statements to fine tune the database files to your specific needs. These tables can contain as many and varied fields as you like, but at least the following must be present:

products and groups tables

1. image\* char
2. price float (products only)
3. taxable char(1)
4. primary key

The above fields are considered 'special' by iVend, and are expected to be present. The presence of additional fields is purely optional, however fields such as shipping wights and costs, product descriptions or inventory counts may also be useful. The tables "products" and "groups" must each have a primary key in order for iVend to operate. By default, this field is called "id" but you may change it to anything that suits you, such as "sku" or "partnumber". iVend will automatically detect the correct field to use at startup.

\* field names that begin with image are considered special by iVend. They indicate images to be stored with groups and products. You can have as many image fields as you like, such as thumbnails and fullsize images.

Additionally, there are a number of tables that are required by iVend to operate. These tables are:

- customer\_info: contains customer address and contact information.
- payment\_info: contains credit card payment information.
- orders: master list of orders.
- orderdata: list of products associated with each order.
- sessions: current shopping cart sessions.
- taxrates: tax rate calculation table.
- status: master list of order and status codes.
- shipments: as orders are shipped, their status is stored in this table.
- shipping\_\*: used internally by the various shipping modules.

You can customize the information requested in the customer\_info and payment\_info fields. Note that the field that is to contain your customer's credit card numbers must be of type char( 255 ) in order to hold the encrypted card number data.

## The Checkout Process

iVend has the capability to configure the store checkout process in an infinite number of ways. In order to configure checkout, we need to think of the process as a series of steps, where each step accomplishes a different task, such as collecting shipping information and so on.

The checkout process is controlled by a set of checkout templates which are located in the `$STOREDIR/html/checkout` directory. The template files in this directory are named `checkout_n.html`, where `n` is a number corresponding with the checkout step the file controls. For example, if you wanted to accomplish checkout in 4 steps, you would have files called `checkout_1.html`, `checkout_2.html`, `checkout_3.html` and `checkout_4.html`. By modifying the checkout templates, we should be able to quickly customize the checkout to match a vendor's needs.

iVend comes with a set of tasks that can be used to build store checkouts. Some of the tasks already available include:

- Shopping cart confirmation.
- Billing and Shipping address collection, including email address checking.
- Payment information collection, with simple credit card number verification.
- Shipping method selection and charge calculation.
- Sales tax calculation.
- Grand Total calculation.
- Final order confirmation.

You may elect to use some or all of these building blocks to accomplish order confirmation with iVend. You can also create new building blocks using iVend modules and procedures. For more information on how to extend iVend, please see the *iVend Programmer's Guide*.

## ***Appendix A: Configuration Interface Settings***

---

## **Appendix B: IVML Markup Tags**

---

The following tags are additions or modifications of existing HTML and RXML tags. They are only available within the iVend system. Please note that checkout tags are discussed in a separate section, as they are independent of the main iVend engine.

`<html></html>` the grand daddy of the tags. Surround all of your pages that are served by iVend with these tags, otherwise the following tags won't work. You may also use `<ivml></ivml>` instead.

`<a></a>` has been modified to provide session tracking capabilities. Additionally, the following attributes have been added:

- `cart`: generates a link to the store's shopping cart page.
- `groups`: generates a link to the store's group listing page.
- `checkout`: generates a link to the store's checkout page.
- `template`: optional attribute to specify a nonstandard template to use.
- `external`: optional attribute to cause iVend to drop sessionid on link.

All other attributes supplied to this container, such as `target`, will be passed through to the browser.

`<ivstatus>` when used on product pages, this tag will display information about additions to a user's shopping cart, and possibly other information from the iVend system.

`<ivmg>` is used to insert an image from a product or group stored in ivend's database. This tag takes the following attributes:

- `field`: indicates the field that contains the image you want to display.

`<icart></icart>` displays the store shopping cart for the current sessionid. This tag takes the following attributes:

- `fields`: a comma separated list of fields to include in the cart.

`<form></form>` has been modified to automatically handle session tracking.

`<listitems>` generates a table of products which fall in a particular group. The following attributes affect this tag:

- `fields`: a comma separated list of fields to include in the table.

- `names`: a comma separated list of alternate names for fields included in the table.

template: specify an alternate template to generate subpages from. The standard template is 'group\_template.html'

show: if this argument is supplied, all items will be displayed. If not, only items with a status of 'A' will be shown.

title: displays the value string as a headline for the following item table in <H2> style.

modulo: if this numeric argument is supplied, a fancy color banded table will be generated with bands appearing in groups of this argument.

tcolors: a comma separated list of four HTML-color values:  
tcolors="table-head-back, table-head-font, list-back, list-font"  
where the table-head-\* is the first line of the table.

headlinebgcolor  
headlinefontcolor  
listbgcolor  
listbgcolor2  
listfontcolor

if specified, these arguments will override the default values for colors in the listitems object.

<category\_output></category\_output> will repeat the contents the container once for each record found, replacing values of each field where instances of the fieldname are surrounded by #s. Similar to <formoutput></formoutput> in RXML. This container takes the following attributes:

type=products|groups: required. selects what type of records to select.  
showall: show all items, options other than random and order are ignored.  
restriction=restriction: optional restriction on selection. example:  
restriction="manufacturer='kodak' "

the restriction must follow SQL format rules.

random: optional, when used with formrotate, the template to be used will be chosen at random from those provided.  
show: show items with all status types: active, inactive, etc.  
order: sql order clause, ie ORDER="name, id asc".

<formrotate></formrotate> is used within the  
<category\_output></category\_output> container. It provides multiple templates to rotate through while processing records. If you provide 2 formrotate containers, every odd record will be processed using the contents of the first container and every even record will be processed using the second container as the template.

<itemoutput></itemoutput> is used on product pages. It surrounds text that will

be parsed for embedded fields, much the same as  
<category\_output></category\_output>. It takes the following attributes:  
extrafields="field1,field2...": additional fields such as calculations that are selected  
from the product table.

<additem> is used to add one or more items.  
noform: do not generate form tags surrounding the item.  
silent: do not show quantity and submit buttons.  
showquantity: show quantity box.  
item: item number.  
quantity: quantity to add.

## Checkout Tags

the subtotal is the sum of all taxable and nontaxable items (including shipping if taxable).  
discounts and salestax are calculated from the subtotal. All checkout tags function within  
the checkout container, <checkout></checkout>.

generate_form	generate a form from the database
table=tablename	this is the table to get data for.
hide=field1,field2	hide these fields in the list.
pulldown=field1,...	generate a pulldown from fields in the db/table_field.val file.
confirmemail	
field=fieldname	name of field containing email address to check.
salestax	added as a lineitem. taxrate is automatically calculated using values supplied in the
admin	interface.
discount	
percent=percentage	percent discount of subtotal, which is then removed from the subtotal.
grandtotal	sum of all lineitems
showorder	show all items in this order.
shippingcost	show the shipping charges
shippingtype	show the shipping type
shippingtypes	show all shipping types with a cost calculation in

a menu.

shippingcalc	show the charge for using a certain shipping method.
shippingadd	add shipping to the order.
addentry	add the data from the previous form.
encrypt=field1,field2..	encrypt these fields using rsa.
noflush	don't remove preexisting db records useful when table key is not unique.
cardcheck	check credit card number
cardnumber=field	use this field name for card number
cardtype=field	use this field name for card type (AMEX,VISA,etc)
expdate=field	use this field name for expiration date

## Other Stuff

- \* The database schema to use currently is located in examples/schema.mysql
- \* The form generator will look for sets of values in a file called storeroot/db/tablename\_fieldname.val ... to create dropdowns.
- \* iVend will send email messages following certain events. These messages are found in the notes directory of your store.

## ***Appendix C: iVend Store Files***

---

These files are considered special to iVend and are located in the `html/` directory of your iVend store directory.

`index.html`: the default page displayed upon arrival at the store.  
`groups.html`: this page is used to display all available store groups.  
`group_template.html`: template used to generate group listings.  
`product_template.html`: template used to generate individual product listings.  
`cart.ivml`: handles shopping cart functions.  
`error.html`: custom error message for iVend errors (optional).

`checkout/*`: these are the checkout steps.

Examples of these files are located in the `examples` directory.

### **Email Notification**

Automatic email messages are sent by iVend following certain events. iVend checks the "notes" directory of your store for the presence of text files. Here is a list of the text files that iVend looks for:

`notify.txt`: sent to the store administrator when a new order is confirmed.

These messages are sent to the individual who places an order:

`confirm.txt`: sent after confirmation of a new order.  
`rejpai.txt`: sent when payment information is rejected.  
`ship.txt`: sent when all or part of an order is shipped.

The messages are parsed through the RXML parser, so you can include the results of sql queries on the product db.

### **The iVend Main Index**

If you place a file called `index.html` in the iVend data directory (as specified in the roxen config interface), and change "CreateIndex" to yes in iVend's global variables, that file will be returned upon an access to the ivend root mountpoint (such as `/ivend/`). This is useful for providing a list of lists for all stores provided by a particular module.

To aid in the generation of a list of lists, a special container called `<ivindex></ivindex>` is available in this page. It works much like the `formoutput`



container in that field names from store configurations that are surrounded by #s like this: #name# will be replaced with that store's value. For a better example, see the file `data/index.html`.

## Actions and Events

During the course of its operation, events that iVend handles will trigger actions.

- + additem
- + deleteitem
- + newsessionid
- + confirmorder

## ***Appendix F: Notes about data security***

---

As with any application of this nature, one must be extremely sensitive when designing and implementing electronic commerce solutions. Improperly implemented designs and security can open holes through which data, such as account numbers or addresses might be compromised.

iVend was written with this in mind, and therefore will use, if available, the cryptography toolkit built into pike to secure sensitive data. Sensitive data is never stored in the clear in databases or transmitted in the clear over unsecure lines.

The encryption methods used by iVend include RSA public key encryption. This means that access to data encrypted with an entity's public key must be made through the corresponding private key. For this reason, it is highly recommended that the two keys be kept in separate directories, preferably on different systems. This way, if the system running iVend is compromised, it would not be possible to decrypt secured information because the required private key would not be present.

*NOTE:* You should make sure that your RSA keyfiles are readable only by trusted users.