

Interchange Upgrade Guide

Table of Contents

| | |
|---|------------------|
| <u>1. Introduction</u> | <u>1</u> |
| <u>2. Interchange 4.8 Deprecated Features</u> | <u>3</u> |
| <u>2.1. Deprecated Features Previous to Interchange 4</u> | <u>3</u> |
| <u>2.2. Interchange 4 Deprecated Features</u> | <u>5</u> |
| <u>3. Upgrading from Minivend 4.0 to Interchange 4.6</u> | <u>7</u> |
| <u>3.1. minivend.cfg</u> | <u>7</u> |
| <u>3.2. Access Manager</u> | <u>7</u> |
| <u>3.3. Database Editing</u> | <u>7</u> |
| <u>3.4. Order Manager</u> | <u>7</u> |
| <u>3.5. Affiliates</u> | <u>8</u> |
| <u>3.6. Page Editor</u> | <u>8</u> |
| <u>3.7. Item Editor</u> | <u>8</u> |
| <u>3.8. Preferences Editor (KNAR)</u> | <u>9</u> |
| <u>3.9. Route Editor</u> | <u>9</u> |
| <u>3.10. Transactions database</u> | <u>10</u> |
| <u>4. Upgrading from Minivend 3 to Minivend 4</u> | <u>11</u> |
| <u>4.1. Nested [loops]</u> | <u>11</u> |
| <u>4.2. All frame features removed</u> | <u>11</u> |
| <u>4.3. Tags removed</u> | <u>11</u> |
| <u>4.4. Directives removed</u> | <u>13</u> |
| <u>4.5. Minor operations removed</u> | <u>13</u> |
| <u>4.6. Search lists</u> | <u>14</u> |
| <u>4.7. Search conditionals</u> | <u>14</u> |
| <u>4.8. Form data updates</u> | <u>14</u> |
| <u>4.9. Checkout changes</u> | <u>15</u> |
| <u>4.10. [if-field] etc.</u> | <u>15</u> |
| <u>4.11. [search-list]</u> | <u>15</u> |
| <u>4.12. Global subs</u> | <u>15</u> |
| <u>A. Minivend 3 compatibility usertags and globalsubs</u> | <u>17</u> |
| <u>A.1. body</u> | <u>17</u> |
| <u>A.2. buttonbar</u> | <u>18</u> |
| <u>A.3. form_mail.cfg</u> | <u>18</u> |
| <u>A.4. help</u> | <u>19</u> |
| <u>A.5. random_rotate</u> | <u>20</u> |
| <u>A.6. AsciiBackend</u> | <u>21</u> |
| <u>5. History of Interchange</u> | <u>23</u> |

1. Introduction

This document contains, in rough form, notes on upgrading from Minivend 3 to Minivend 4, and Minivend 4 to Interchange.

2. Interchange 4.8 Deprecated Features

This document describes features of Interchange 4.8 that have been deprecated. Any use of these features should be discontinued. In most cases we have provided an alternative mechanism to accomplish the same results. These deprecated features may be removed at some point in the future. You should change to the new mechanism to avoid breakage.

2.1. Deprecated Features Previous to Interchange 4

This section needs some serious work.

cart/page from path

interchange.PL 308,313

```
if($path =~ s/(.*)::) {
    $cart = $1;
    if($cart =~ s/(.*)::) {
        $page = $1;
    }
}
```

mv_orderpage

interchange.PL 321,323

```
$CGI::values{mv_nextpage} = $CGI::values{mv_orderpage}
                                || find_special_page('order')
if ! $CGI::values{mv_nextpage};
```

\$decode

interchange.PL 493

```
HTML::Entities::decode($value) if $decode;
```

mv_orderpage

interchange.PL 854,855

```
$CGI::values{mv_nextpage} = $CGI::values{mv_orderpage}
    if $CGI::values{mv_orderpage};
```

ROUTINES and LANG

intechange.PL 1552,1579

```
ROUTINES: {
    last ROUTINES unless index($Vend::FinalPath, '/process/') == 0;
    while ($Vend::FinalPath =~ s:/process/(locale|language|currency)/([^\/]*)/:pr
        $::Scratch->{"mv_$1"} = $2;
    }
    $Vend::FinalPath =~ s:/process/page/::;
```

Interchange Upgrade Guide

```
}
my $locale;
if($locale = $::Scratch->{mv_language}) {
    $Global::Variable->{LANG}
        = $::Variable->{LANG} = $locale;
}

if ($Vend::Cfg->{Locale}
    $locale = $::Scratch->{mv_locale}      and
    defined $Vend::Cfg->{Locale_repository}->{$locale}
)
{
    $Global::Variable->{LANG}
        = $::Variable->{LANG}
        = $::Scratch->{mv_language}
        = $locale
        if ! $::Scratch->{mv_language};
    Vend::Util::setlocale( $locale,
                           ($::Scratch->{mv_currency} || undef)
                           { persist => 1 }
    )
}
```

list_compat

lib/Vend/Interpolate.pm 2808

```
list_compat($opt->{prefix}, \$text);
```

lib/Vend/Interpolate.pm 3538

```
list_compat($opt->{prefix}, \$text);
```

lib/Vend/Interpolate.pm 3874

```
list_compat($opt->{prefix}, \$page);
```

find_sort

lib/Vend/Interpolate.pm 3270,3271

```
$text =~ /^s*\[sort\s+.*\/si
    and $opt->{sort} = find_sort(\$text);
```

mv_order_report

lib/Vend/Order.pm 867,868

```
$body = readin($::Values->{mv_order_report})
    if $::Values->{mv_order_report};
```

mv_error_\$var

lib/Vend/Order.pm 1030

```
$::Values->{"mv_error_$var"} = $message;
```


2.2. Interchange 4 Deprecated Features

Vend::Util::send_mail Vend::Order::send_mail send_mail

The send_mail routine has been replaced by the Vend::Mail::send routine.

3. Upgrading from Minivend 4.0 to Interchange 4.6

if [item-price] suddenly turns 0, check PriceField in the catalog.cfg

3.1. minivend.cfg

- Remove references to MiniMate.
- Add this line to minivend.cfg:
#include lib/UI/ui.cfg
Make sure the files catalog_before.cfg and catalog_after.cfg are there, or add their contents to etc/your_cat_name.before and etc/your_cat_name.after to it only for some catalogs.

3.2. Access Manager

You need to get the minimate.asc file renamed to access.asc and add the following fields to the first line:

```
groups
last_login
name
password
```

Remove these catalog.cfg lines:

| | | | |
|----------|----------------|--------------|-----|
| Variable | MINIMATE_META | mv_metadata | |
| Variable | MINIMATE_TABLE | minimate | |
| Database | minimate | minimate.asc | TAB |

Add this one:

| | | | |
|----------|-----------|---------------|-----|
| Database | affiliate | affiliate.txt | TAB |
|----------|-----------|---------------|-----|

Authentication for admin users is now done from a separate table than customers, and passwords are encrypted.

3.3. Database Editing

Update the mv_metadata.asc file as appropriate.

3.4. Order Manager

Some things that are needed for the order manager:

- Add these fields to transactions:

```
affiliate  approx. char(32)
archived   char(1)
campaign   approx. char(32)
comments   blob/text
complete   char(1)
deleted     char(1)
order_wday char(10)
order_ymd   char(8)
```

```
po_number approx. char(32)
```

- Add these fields to transactions:

```
affiliate approx. char(32)
campaign approx. char(32)
```

- Remove this field from userdb:

```
mv_credit_card_info
```

- Add these fields to userdb:

```
inactive char(1)
credit_limit char(14) or decimal(12,2)
dealer char(3)
```

- Create the directory 'logs'.
- Create the directory 'orders' if it doesn't already exist.
- Update your order routes to those in the Interchange distribution. Note that the route log_entry is necessary if you want to enter orders from the Interchange UI.
- Update the etc/log_transaction file.
- Add the etc/log_entry file.
- Add this to catalog.cfg:

```
## Don't want people setting their credit_limit directly
UserDB default scratch "credit_limit dealer"
```

3.5. Affiliates

Add a tab-delimited affiliate table:

```
affiliate name    campaigns    join_date    url timeout active password
```

You can find a recommended database configuration in foundation/dbconf/*/affiliate.*.

3.6. Page Editor

Add the directories 'templates' and 'backup'. Copy the contents of the Interchange simple/templates to templates.

3.7. Item Editor

Add a merchandising table with the following fields:

```
Database merchandising merchandising.txt __SQLDSN__
Database merchandising DEFAULT_TYPE text

sku char(32)
featured char(32)
banner_text
banner_image
```

```

blurb_begin
blurb_end
timed_promotion      char(16)
start_date           char(24)
finish_date          char(24)
upsell_to
cross_sell
cross_category       char(64)
others_bought
times_ordered

```

Index the fields with char(*) types. You can find the recommended database configuration in `foundation/dbconf/*/merchandising.*`

3.8. Preferences Editor (KNAR)

Create the tab-delimited file `variable.txt` with these fields:

```
code  Variable  pref_group
```

Add this as the **first** line of `catalog.cfg`:

```
VariableDatabase variable
```

3.9. Route Editor

Create the file `route.txt` with these fields:

```

code
report
receipt
encrypt_program
encrypt
pgp_key
pgp_cc_key
cyber_mode
credit_card
profile
inline_profile
email
attach
counter
increment
continue
partial
supplant
track
errors_to

```

Add this line in `catalog.cfg`:

```
RouteDatabase route
```

3.10. Transactions database

The back office UI should work fine for editing database tables. Obviously the things which are specific to the order transaction setup will break unless you have the right fields, but even these can be controlled by configuring the UI.

Add a new field to transaction.txt called 'archived'.

4. Upgrading from Minivend 3 to Minivend 4

There were big changes from Minivend 3 to Minivend 4, some of which were incompatible. Many things were removed as redundant, deprecated, or just plain crufty:

4.1. Nested [loop]s

MV3 used a different scheme for creating nested loop lists:

`[loop with="-a"* arg="item item item" search="se=whatever"]`

allowed you to refer to the nested values with a `[loop-code-a]` construct. In Minivend 4, the form is:

```
[loop prefix=size list="Small Medium Large"]
  [loop prefix=color list="Red White Blue"]
    [color-code]-[size-code]<BR>
  [/loop]
<P>
[/loop]
```

4.2. All frame features removed

Frames are now managed by the user in HTML.

4.3. Tags removed

4.3.1. buttonbar

Replace with Variable defined in catalog.cfg. buttonbar was previously used as an SSI-like command for catalog-wide standardized features like navigation bars. In the 3.x catalog.cfg the ButtonBars parameter defines a list of html snippets, like

```
ButtonBars header.html footer.html copyright.html
```

So `[buttonbar 0]` substitutes 'header.html', `[buttonbar 1]` substitutes 'footer.html', etc.

In 4.x catalog.cfg, define variables, like

```
Variable HEADER      <pages/header
Variable FOOTER      <pages/footer
Variable COPYRIGHT    <pages/copyright
```

Then replace all occurrences of `[buttonbar 0]` with `__HEADER__`, `[buttonbar 1]` with `__FOOTER__`, etc. Note that the old header.html, footer.html, etc. contained html code, but were not actually html pages with `<html><body>` etc, tags. Thus the current practice is to use filenames with no extension or perhaps '.txt' to differentiate them from pages.

4.3.2. random

Replace with `[ad random=1]` or custom code. See the `[ad]` tag docs. Random and rotate were used to place random or rotating regions on pages, such as banner ads.

The Random directive in catalog.cfg defines the numbered HTML snippet files, similar to buttonbars above.

4.3.3. rotate

Replace with [ad ...]. See [random] above.

4.3.4. help

No replacement. Use data functions or variables.

4.3.5. body

Replace with templates. Again the body tag [body 1] etc. defines numbered body definitions that could be applied site-wide. However, in this case minivend actually built up the <body> substitution using the Mv_* directives in catalog.cfg.

4.3.6. finish_order

[finish_order] was a conditional tag; if the basket contained anything a 'checkout' graphic would be displayed. No replacement; use [if items]Message[/if].

4.3.7. last_page

No replacement – this can be emulated by setting a scratch variable on one page, then using it to build the return URL.

4.3.8. item-link

No replacement, just use [page [item-code]].

4.3.9. loop-link

No replacement, just use [page [loop-code]].

4.3.10. sql-link

No replacement, just use [page [sql-code]].

4.3.11. accessories

Replace with normal data functions.

4.3.12. Compatibility routines

Compatibility routines for many popular tags like [random], [rotate], etc. are provided in the appendix of this document. To use one, copy it to a file and put it in your usertag directory. (Tags in the usertag directory are read in by interchange.cfg by default).

4.4. Directives removed

ActionMap
 AdminDatabase
 AdminPage
 AsciiBackend
 BackendOrder
 ButtonBars
 CheckoutFrame
 CheckoutPage
 CollectData
 DataDir
 Delimiter
 DescriptionTrim
 FieldDelimiter
 FrameFlyPage
 FrameLinkDir
 FrameOrderPage
 FrameSearchPage
 ItemLinkDir
 ItemLinkValue
 MsqlDB
 MsqlProducts
 Mv_AlinkColor
 Mv_Background
 Mv_BgColor
 Mv_LinkColor
 Mv_TextColor
 Mv_VlinkColor
 NewReport
 NewTags
 OldShipping
 OrderFrame
 PageCache
 PriceDatabase
 Random
 ReceiptPage
 RecordDelimiter
 ReportIgnore
 Rotate
 SearchFrame
 SearchOverMsg
 SecureOrderMsg
 SpecialFile
 SubArgs
 Tracking

4.5. Minor operations removed

- auto-substitution of mp= on [loop search=profile], [search-region arg=profile]
- [tag scan]...
- [tag sql]...

Many of these are related to one of:

 - Removal of frames logic
 - Removed tags
 - Obsolete methods
 - Old routines for 2.0x compatibility

4.6. Search lists

Search tags must now be surrounded by [search-region] [/search-region]. This is because multiple searches can be done in a page, with multiple [more-list] entries, multiple [no-match] areas, etc. It was not really possible to avoid this and add the feature.

To find all files containing the search list, do:

```
find pages -type f | xargs grep -l '\[search.list'
```

That will yield a set of files that need to be updated. You should surround all parts of the search area, i.e.:

```
[search-region]

[search-list]
    your search iteration stuff, [item-code], etc.
[/search-list]

[more-list]
    [more]
[/more-list]

[/search-region]
```

4.7. Search conditionals

Search conditionals should now say [if-item-field field] [/if-item-field] and [if-item-data table column] [/if-item-data]. This allows mixing and nesting of lists. You may find that the old works in some situations, but it will not work in all situations.

4.8. Form data updates

Added Scratch variable mv_data_enable to gate the update_data function. You must set it before doing a form update. Prior to this it was possible to update a SQL database willy-nilly.

A quick fix like this will allow the update on a single page:

```
[set update_database]
[set mv_data_enable]1[/set]
[/set]
<INPUT TYPE=hidden NAME=mv_click VALUE=update_database>
```

It will ensure at least that the user loads one form from you for each update. For best security, gate with a userdb entry like this:

```
[set update_database]
[if type=data term="userdb::trusted::[data session username]" ]
    [set mv_data_enable]1[/set]
[else]
    [set mv_data_enable]0[/set]
[/else]
[/if]
[/set]
```

4.9. Checkout changes

Minivend 4 uses in-page error-flagging on the checkout page. Simplest way to convert is probably to use the checkout.html from the simple demo as a start, and move in any customization from the existing site's catalog.html (headers, footers, logos, etc.) A line-by-line comparison of the data fields in the checkout page should be performed, adding any custom fields as needed. Custom error checking in etc/order.profiles may have to be re-worked, or can be added into checkout.html using the in-page order profile capability. Remember to update receipt.html and report/report.html with any custom fields, as well.

4.10. [if-field] etc.

The least-compatible things in the tag area are [if-field] (needs to be [if-PREFIX-field], where prefix might be item|loop by default depending on the tag. Likewise:

```
[if-data table col] --> [if-PREFIX-data table col]
[on-change mark]    --> [PREFIX-change mark]
[if-param param]    --> [if-PREFIX-param param]
[PREFIX-param N]     --> [PREFIX-pos N] (where N is a digit)
```

4.11. [search-list]

You must always surround [search-list] with [search-region] [/search-region].

Embedded Perl changes quite a bit. While there are the \$Safe{values} and other variable settings, they are automatically shared (no arg="values") and move to:

```
$Safe{values}  --> $Values
$Safe{cgi}     --> $CGI
$Safe{carts}   --> $Carts
$Safe{items}   --> $Items
$Safe{config}  --> $Config
$Safe{scratch} --> $Scratch
```

There are a number of other objects, see the docs.

Most other issues have more to do with the catalog skeleton (i.e. simple or barry or basic or art) than they do the core. For instance, the "basic" catalog produced for MV3 ran unchanged except for the issues discussed above.

4.12. Global subs

Accessing globalsubs from [perl] tags is done slightly differently.

Minivend 3 method:

```
[perl sub]
myfunsub();
[/perl]
```

Minivend 4/IC method:

```
[perl subs=1]
myfunsub();
[/perl]
```

Interchange Upgrade Guide

If you do this wrong, you'll get an error that looks like this:

```
115.202.115.237 H8gbq6oK:115.202.115.237 - [28/February/2001:18:58:50 -0500] testcat /cgi-bin/testca
```

A. Minivend 3 compatibility usertags and globalsubs

These files were originally distributed with Minivend 4 in the compat/ directory. They replace Minivend 3 functionality that was removed or greatly altered in Minivend 4.

A.1. body

```
UserTag body PosNumber 2
UserTag body Order type extra
UserTag body Routine <<EOR
use vars qw($C);
sub parse_color {
    my ($var, $value) = @_ ;
    return '' unless $value;
    $var = lc $var;
    $C->{Color}->{$var} = [];
    @{$C->{'Color'}->{$var}} = split /\s+/, $value;
    return $value;
}

sub {
    my ($scheme, $extra) = @_ ;
    my $r = '<BODY';
    my ($var, $tag);
    #return '<BODY>' unless (int($scheme) < 16 and int($scheme) > 1);

    my %color = qw( mv_bgcolor BGCOLOR mv_textcolor TEXT
                    mv_linkcolor LINK mv_vlinkcolor VLINK
                    mv_alinkcolor ALINK mv_background BACKGROUND );

    if (defined $::Values->{mv_resetcolors}
        and $::Values->{mv_resetcolors}) {
        delete $::Values->{mv_customcolors};
        undef $::Values->{mv_resetcolors};
    }
    if (defined $::Values->{mv_customcolors}) {
        foreach $var (keys %color) {
            $r .= qq| $color{$var}= "| . $::Values->{$var} . ' ' '
            if $::Values->{$var};
        }
    }
    else {
        foreach $var (keys %color) {
            $r .= qq| $color{$var}= "| . ${ $Vend::Cfg->{Color}->{$var} }[$scheme] . ' ' '
            if defined ${ $Vend::Cfg->{Color}->{$var} }[$scheme]
                && ${ $Vend::Cfg->{Color}->{$var} }[$scheme] !~ /\bnone\b/ ;
        }
    }
    $r =~ s#(BACKGROUND="(?!http:))([^\]|)#$1$Vend::Cfg->{ImageDir}$2#;
    $r .= " $extra" if defined $extra;
    $r .= '>';
}
EOR

AddDirective Mv_Background    color
AddDirective Mv_BgColor      color
AddDirective Mv_TextColor     color
AddDirective Mv_LinkColor     color
```

```
AddDirective Mv_AlinkColor    color
AddDirective Mv_VlinkColor    color
```

A.2. buttonbar

```
# Returns a buttonbar by number
UserTag buttonbar Order type
UserTag buttonbar PosNumber 1
UserTag buttonbar Interpolate 1
UserTag buttonbar Routine <<EOR
sub get_files {
    my($dir, @files) = @_;
    my(@out);
    my($file, $contents);
    foreach $file (@files) {
        config_error(
            "No leading ../.. allowed if NoAbsolute set. Contact administrator.\n")
        if $file =~ m#^\./\.\./*\.\.\.# and $Global::NoAbsolute;
        push(@out, "\n") unless
            push(@out, readfile("$dir/$file.html"));
    }

    @out;
}

sub parse_buttonbar {
    my ($var, $value) = @_;
    return [] unless $value;
    my @c;
    my @vals = grep /\S/, split /\s+/, $value;
    for(@vals) {
        push @c, Vend::Util::readfile("pages/$_ .html");
    }
    return \@c;
}

sub {
    my($buttonbar) = @_;
    if (defined $Vend::Cfg->{'ButtonBars'}->[$buttonbar]) {
        return $Vend::Cfg->{'ButtonBars'}->[$buttonbar];
    }
    else {
        return '';
    }
}
EOR

AddDirective ButtonBars    buttonbar
```

A.3. form_mail.cfg

```
GlobalSub <<EndOfSub
sub form_mail {
    my($to, $subject, $reply, $body) = @_;
    my($ok);

    $subject = '<no subject>' unless defined $subject && $subject;

    $reply = '' unless defined $reply;
```

```

$reply = "Reply-to: $reply\n" if $reply;

$ok = 0;
SEND: {
    open(Vend::MAIL,"|$Vend::Cfg->{'SendMailProgram'} -t") or last SEND;
    print Vend::MAIL
        "To: $to\n",
        $reply,
        "Subject: $subject\n",
        "Errors-To: $Vend::Cfg->{MailOrderTo}\n\n",
        $body
        or last SEND;
    close Vend::MAIL or last SEND;
    $ok = ($? == 0);
}

if (!$ok) {
    logError("Unable to send mail using $Vend::Cfg->{'SendMailProgram'}\n" .
        "To '$to'\n" .
        "With subject '$subject'\n" .
        "With reply-to '$reply'\n" .
        "And body:\n$body");
}
$ok;
}
EndOfSub

```

A.4. help

```

UserTag help PosNumber 1
UserTag help Order name
UserTag help Routine <<EOR
sub parse_help {
    my ($var, $value) = @_;
    my (@files);
    my (@items);
    my ($c, $chunk, $item, $help, $key);
    unless (defined $value && $value) {
        $c = {};
        return $c;
    }
    $c = $C->{'Help'};
    $var = lc $var;
    $C->{'Source'}->{'Help'} = $value;
    @files = get_files($C->{'PageDir'}, split /\s+/, $value);
    foreach $chunk (@files) {
        @items = split /\r?\n\r?\n/, $chunk;
        foreach $item (@items) {
            ($key,$help) = split /\s*\n/, $item, 2;
            if(defined $c->{$key}) {
                $c->{$key} .= $help;
            }
            else {
                $c->{$key} = $help;
            }
        }
    }
    return $c;
}

```

```

sub {
  my($help) = shift;
  # Move this to control section?
  if ($::Values->{mv_helpon}) {
    delete $::Values->{mv_helpoff};
    undef $::Values->{mv_helpon};
  }
  return '' if defined $::Values->{'mv_helpoff'};
  if (defined $Vend::Cfg->{'Help'}{$help}) {
    return $Vend::Cfg->{'Help'}{$help};
  }
  else {
    return '';
  }
}
EOR

```

AddDirective Help help

A.5. random_rotate

```

UserTag random PosNumber 0
UserTag random Interpolate 1
UserTag random Routine <<EOR
package Vend::Config;
sub parse_random {
  my ($var, $value) = @_;
  return '' unless (defined $value && $value);
  my $c = [];
  $var = lc $var;
  my @files = grep /\S/, split /\s+/, $value;
  local ($Vend::Cfg) = $C;
  for (@files) { push @$c, Vend::Util::readin($_) }
  return $c;
}

package Vend::Interpolate;
sub {
  my $random = int rand(scalar(@{$Vend::Cfg->{'Random'}}));
  if (defined $Vend::Cfg->{'Random'}->[$random]) {
    return $Vend::Cfg->{'Random'}->[$random];
  }
  else {
    return '';
  }
}
EOR

UserTag rotate PosNumber 2
UserTag rotate Order ceiling floor
UserTag rotate Interpolate 1
UserTag rotate Routine <<EOR
sub {
  return '' unless $Vend::Cfg->{Rotate};
  my $ceiling = $_[0] || @{$Vend::Cfg->{'Rotate'}} || return '';
  my $floor = $_[1] || 1;

  $ceiling--;
  $floor--;

  my $marker = "rotate$floor$ceiling";

```



```

if($ceiling < 0 or $floor < 0) {
    $floor = 0;
    $ceiling = scalar @{$Vend::Cfg->{'Rotate'}} - 1;
    logError "Bad ceiling or floor for rotate";
}

my $rotate;
$rotate = $Vend::Session->{$marker} || $floor;

if($rotate > $ceiling or $rotate < $floor ) {
    $rotate = $floor;
}

$Vend::Session->{$marker} = $rotate + 1;
return $Vend::Cfg->{'Rotate'}->[$rotate];
}
EOR

```

```

AddDirective Random random
AddDirective Rotate random

```

A.6. AsciiBackend

```

GlobalSub <<EOS
sub AsciiBackend {
    package Vend::Order;
    $Vend::Order::override_track_order = \&track_order;
    sub track_order_backend {
        my ($order_no,$order_report) = @_;
        my ($c,$i);
        my (@backend);

        @backend = split /\s*,\s*/, $Vend::Cfg->{BackendOrder};

        if(@backend and $Vend::Cfg->{AsciiBackend}) {
            my(@ary);
            push @ary, $order_no;
            for(@backend) {
                push @ary, $::Values->{$_};
            }
            foreach $i (0 .. $#Vend::Items) {
                push @ary, $Vend::Items->[$i]{'code'};
                push @ary, $Vend::Items->[$i]{'quantity'};
                if ($Vend::Cfg->{UseModifier}) {
                    foreach $j (@{$Vend::Cfg->{UseModifier}}) {
                        push @ary, $Vend::Items->[$i]->{$j}
                    }
                }
            }
            logData ($Vend::Cfg->{AsciiBackend}, @ary);
        }
        $Vend::Order::override_track_order->($order_no, $order_report);
    }
    *track_order = \&Vend::Order::override_track_order;
}
EOS

AddDirective BackendOrder
AddDirective AsciiBackend

```


5. History of Interchange

Interchange is a descendent of Vend, an e-commerce solution originally developed by Andrew Wilcox in early 1995. Mike Heins took the first publicly-released version, Vend 0.2, and added searching and DBM catalog storage to create MiniVend. Mike released MiniVend 0.2m7 on December 28, 1995. Subsequent versions of MiniVend took parts from Vend 0.3, especially the vlink and Server.pm modules, which were adapted to run with MiniVend. In the four years that followed, Mike Heins expanded and enhanced MiniVend, creating a powerful and versatile e-commerce development platform. MiniVend grew to support thousands of businesses and their e-commerce sites.

Separately, an experienced e-commerce development team founded Akopia. Their goal was to create a sophisticated open source e-commerce platform that was both feature-rich and easy to use. Their product, Tallyman, was intuitive, and had great content-management features, but lacked many of MiniVend's capabilities.

Akopia acquired MiniVend in June 2000. Mike Heins and the Tallyman developers combined MiniVend with Tallyman's features to create Interchange. Interchange replaces both MiniVend and Tallyman. In order to preserve compatibility, the name "minivend" and prefixes like "mv_" and "MVC_" will still appear in source code and configuration files.

In January 2001, Red Hat acquired Akopia and created its new E-Business Solutions Division. Interchange development is going forward and the user community continues to grow.

