

Interchange I18N Features

Table of Contents

<u>1. Internationalization</u>	1
<u>1.1. Configuring the Locale</u>	1
<u>1.2. Setting the Locale</u>	1
<u>1.3. Interchange Locale Settings</u>	2
<u>1.4. Special Locale Keys for Price Representation</u>	3
<u>1.5. Dynamic Locale Directive Changes</u>	4
<u>1.6. Sorting Based on Locale</u>	6
<u>1.7. Placing Locale Information in a Database</u>	7

Interchange I18N Features

1. Internationalization

Interchange has a rich set of internationalization (I18N) features that allow conditional message display, differing price formats, different currency definitions, price factoring, sorting, and other settings. The definitions are maintained in the `catalog.cfg` file through the use of built-in POSIX support and Interchange's `Locale` directive. All settings are independent for each catalog and each user visiting that catalog, since customers can access the same catalog in an unlimited number of languages and currencies.

1.1. Configuring the Locale

It is recommended to use the `ScratchDefault` directive for setting the catalog's default locale:

```
ScratchDefault mv_locale de_DE
```

1.2. Setting the Locale

The locale could be set to `fr_FR` (French for France) in one of two ways:

[setlocale locale=locale* currency=locale* persist=1*]

Immediately sets the locale to `locale`, and will cause it to persist in future user pages if the `persist` is set to a non-zero, non-blank value. If the `currency` attribute is set, the pricing and currency-specific locale keys and Interchange configuration directives are modified to that locale. If there are no arguments, it sets it back to the user's default locale as defined in the scratch variables `mv_locale` and `mv_currency`.

This allows:

```
Dollar Pricing:

[setlocale en_US]
[item-list]
[item-code]: [item-price]<BR>
[/item-list]

Franc Pricing:

[setlocale fr_FR]
[item-list]
[item-code]: [item-price]<BR>
[/item-list]

[comment] Return to the user's default locale [/comment]
[setlocale]
```

[page process/locale/fr_FR/page/catalog]

This is the same as `[page catalog]`, except when the link is followed it will set the locale to `fr_FR` before displaying the page. This is persistent.

[page process/locale/fr_FR/currency/en_US/page/catalog]

This is the same as `[page catalog]`, except when the link is followed it will set the locale to `fr_FR` and the pricing/number display to the locale `en_US` before displaying the page. This is persistent.

Once the locale is persistently set for a user, it is in effect for the duration of their session.

1.3. Interchange Locale Settings

The `Locale` directive has many possible settings that allow complete internationalization of page sets and currencies. The `Locale` directive is defined in a series of key/value pairs with a key that contains word characters only being followed by a value. The value must be enclosed in double quotes if it contains whitespace. In this example, the key is `Value setting`.

```
Locale fr_FR "Value setting" "Configuration de valeur"
Locale de_DE "Value setting" "Werteinstellung"
```

When accessed using the special tag `[L]Value setting[/L]`, the value `Configuration de valeur` will be displayed **only** if the locale is set to `fr_FR`. If the locale is set to `de_DE`, the string `Werteinstellung` will be displayed. If it is neither, the default value of `Value setting` will be displayed.

The `[L]` and `[/L]` must be capitalized. This is done for speed of processing as well as easy differentiation in text.

Another way to do this is right in the page. The `[LC] ... [/LC]` pragma pair permits specification of locale-dependent text.

```
[LC]
    This is the default text.
[fr_FR] Text for the fr_FR locale. [/fr_FR]
[de_DE] Text for the de_DE locale. [/de_DE]
[/LC]
```

You can also place an entirely new page in place of the default one if the locale key is defined. When a locale is in force, and a key named `HTMLsuffix` is set to that locale, Interchange first looks for a page with a suffix corresponding to the locale. For example:

```
<A HREF="[area index]">Catalog home page</A>
```

If a page `index.html` exists, it will be the default. If the current locale is `fr_FR`, a page `index.fr_FR` exists, and `Locale` looks like this:

```
Locale fr_FR HTMLsuffix .fr_FR
```

Then, the `.fr_FR` page will be used instead of the `.html` page. For a longer series of strings, the configuration file recognizes:

```
Locale fr_FR <<EOF
{
    "Value setting",
    "Configuration de valeur",

    "Search",
    "Recherche"
}
EOF
```

This example sets two string substitutions. As long as this is a valid Perl syntax describing a series of settings, the text will be matched. It can contain any arbitrary set of characters that don't contain [L] and [/L]. If using double quotes, string literals like \n and \t are recognized.

A database can also be used to set locale information. Locale information can be added to any database in the `catalog.cfg` file, and the values in it will overwrite previous settings. For more information, see `LocaleDatabase`. The [L]default text[/L] is set before any other page processing takes place. It is equivalent to the characters "default text" or the appropriate `Locale` translation for all intents and purposes. Interchange tags and `Variable` values can be embedded.

Because the [L] message [/L] substitution is done before any tag processing, the command [L][item-data table field][[/L] will fail. There is an additional [loc] message [/loc] `UserTag` supplied with the distribution. It does the same thing as [L] [/L] except it is programmed after all tag substitution is done. See the `interchange.cfg.dist` file for the definition.

Note: Be careful when editing pages containing localization information. Even changing one character of the message can change the key value and invalidate the message for other languages. To prevent this, use:

```
[L key]The default.[/L]
```

The key `msg_key` will then be used to index the message. This may be preferable for many applications.

A `localize` script is included with Interchange. It will parse files included on the command line and produce output that can be easily edited to produce localized information. Given an existing file, it will merge new information where appropriate.

1.4. Special Locale Keys for Price Representation

Interchange honors the standard POSIX keys:

```
mon_decimal_point    or    decimal_point
mon_thousands_sep   or    thousands_sep
currency_symbol      or    int_currency_symbol
frac_digits          or    p_cs_precedes
```

See the POSIX `setlocale(3)` man page for more information. These keys will be used for formatting prices and approximates the number format used in most countries. To set a custom price format, use these special keys:

price_picture

Interchange will format a currency number based on a "picture" given to it. The basic form is:

```
Locale en_US price_picture "$ ###,###,###.##"
```

The `en_US` locale, for the United States, would display `4452.3` as `$ 4,452.30`. The same display can be achieved with:

```
Locale en_US mon_thousands_sep ,
Locale en_US mon_decimal_point .
Locale en_US p_cs_precedes 1
Locale en_US currency_symbol $
```

Interchange I18N Features

A common `price_picture` for European countries would be `###.###.###,##`, which would display that same number as 4.452,30. To add a franc notation at the end for the locale `fr_FR`, use the setting:

```
Locale fr_FR price_picture "##.###,## fr"
```

IMPORTANT NOTE: The decimal point in use, set by `mon_decimal_point`, and the thousands separator, set by `mon_thousands_sep` must match the settings in the `price_picture`. The `frac_digits` setting is not used in this case. It is derived from the location of the decimal (if any).

The same setting for `fr_FR` above can be achieved with:

```
Locale fr_FR mon_thousands_sep .
Locale fr_FR mon_decimal_point ,
Locale fr_FR p_cs_precedes 0
Locale fr_FR currency_symbol fr
```

If the number of digits is greater than the # locations in the `price_picture`, the digits will be changed to asterisks. An overflow number above would show as `**.***,** fr`.

picture

Same as `price_picture`, but sets the value returned if the `[currency]` tag is not used. If the number of digits is greater than the # locations in the `picture`, the digits will be changed to asterisks, displaying something like `** ,***.**`.

1.5. Dynamic Locale Directive Changes

If a Locale key is set to correspond to an Interchange `catalog.cfg` directive, that value will be set when the locale is set.

PageDir

To use a different page directory for different locales, set the `PageDir` key. For example, to have two separate language page sets, French and English, set:

```
# Establish the default at startup
PageDir english
Locale fr_FR PageDir francais
Locale en_US PageDir english
```

ImageDir

To use a different image directory for different locales, set the `ImageDir` key. To have two separate language button sets, French and English, set:

```
# Establish the default at startup
ImageDir /images/english/
Locale fr_FR ImageDir /images/francais/
Locale en_US ImageDir /images/english/
```

ImageDirSecure

See ImageDir.

PriceField

To use a different field in the products database for pricing based on locale, set the PriceField locale setting. For example:

```
# Establish the default at startup
PriceField price
Locale fr_FR PriceField prix
```

The default will always be price, but if the locale fr_FR is set, the PriceField directive will change to prix to give prices in francs instead of dollars.

If PriceBreaks is enabled, the prix field from the pricing database will be used to develop the quantity pricing.

Note: If no Locale settings are present, the display will always be price, regardless of what was set in PriceField. Otherwise, it will match PriceField.

PriceDivide

Normally used to enable penny pricing with a setting of 100, PriceField can be used to do an automatic conversion calculation factor based on locale.

```
# Default at startup is 1 if not set
# Franc is strong these days!
Locale fr_FR PriceDivide .20
```

The price will now be divided by .20, making the franc price five times higher than the dollar.

PriceCommas

This controls whether the mon_thousands_sep will be used for standard currency formatting. This setting will be ignored if you are using price_picture. Set the value to 1 or 0, to enable or disable it. Do not use yes or no.

```
# Default at startup is Yes if not set
PriceCommas Yes
Locale fr_FR PriceCommas 0
Locale en_US PriceCommas 1
```

UseModifier

Changes the fields from the set shopping cart options.

```
# Default at startup is 1 if not set
# Franc is strong these days!
UseModifier format
Locale fr_FR UseModifier formats
```

If a previous setting was made for an item based on another locale, it will be maintained.

PriceAdjustment

Interchange I18N Features

Changes the fields set by `UseModifier` that will be used to adjust pricing for an automatic conversion factor based on locale. For example:

```
# Default at startup
PriceAdjustment format
Locale fr_FR PriceAdjustment formats
```

TaxShipping,SalesTax

Same as the standard directives.

DescriptionField

This changes the field accessed by default with the `[item-description]` and `[description code]` tags. For example

```
# Establish the default at startup
DescriptionField description
Locale fr_FR DescriptionField desc_fr
```

The [locale] tag

Standard error messages can be set based on Locale settings. Make sure not to use any of the predefined keys. It is safest to begin a key with `msg_`. The default message is set between the `[locale key]` and `[/locale]` tags. See the example above.

1.6. Sorting Based on Locale

The Interchange `[sort database:field]` keys will use the `LC_COLLATE` setting for a locale provided that:

- The operating system and C compiler support locales for POSIX, and have the locale definitions set.
- The locale setting matches any configured locales.

If this arbitrary database named `letters`:

```
code      letter
00-0011   f
99-102    é
19-202    a
```

and this loop:

```
[loop 19-202 00-0011 99-102]
[sort letters:letter]
[loop-data letters letter]  [loop-code]
[/loop]
```

used the default C setting for `LC_COLLATE`, the following would be displayed:

```
a 19-202
f 00-0011
é 99-102
```

If the proper LC_COLLATE settings for locale fr_FR were in effect, then the above would become:

```
a 19-202
é 99-102
f 00-0011
```

1.7. Placing Locale Information in a Database

Interchange has the capability to read its locale information from a database, named with the `LocaleDatabase` directive. The database can be of any valid Interchange type. The locales are in columns, and the keys are in rows. For example, to set up price information:

key	en_US	fr_FR	de_DE
PriceDivide	1	.1590	.58
mon_decimal_point	.	,	,
mon_thousands_sep	,	.	.
currency_symbol	\$	frs	DM
ps_cs_precedes	1	0	0

This would translate to the following:

```
Locale en_US PriceDivide 1
Locale en_US mon_decimal_point .
Locale en_US mon_thousands_sep ,
Locale en_US currency_symbol $
Locale en_US ps_cs_precedes 1

Locale fr_FR PriceDivide .1590
Locale fr_FR mon_decimal_point ,
Locale fr_FR mon_thousands_sep .
Locale fr_FR currency_symbol " frs"
Locale fr_FR ps_cs_precedes 0

Locale de_DE PriceDivide .58
Locale de_DE mon_decimal_point ,
Locale de_DE mon_thousands_sep " "
Locale de_DE currency_symbol "DM "
Locale de_DE ps_cs_precedes 1
```

These settings append and overwrite any that are set in the catalog configuration files, including any include files.

Important note: This information is only read during catalog configuration. It is not reasonable to access a database for translation or currency conversion in the normal course of events.

Copyright 2002–2004 Interchange Development Group. Copyright 2001–2002 Red Hat, Inc. Freely redistributable under terms of the GNU General Public License.

