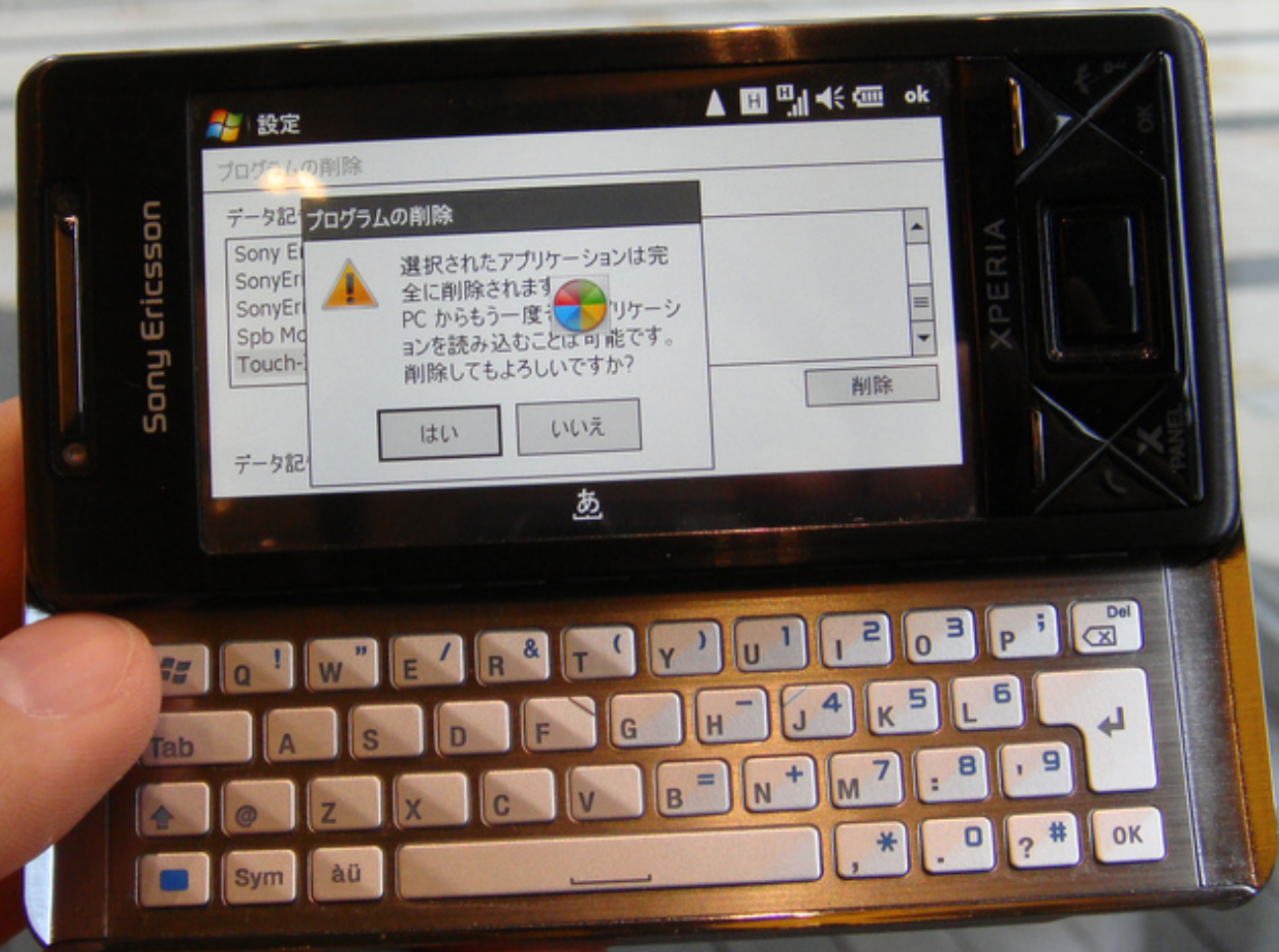




LUNATECH
RESEARCH

Language localisation in Java, JSF and Seam

2009-05-22/PeterHilton@www.lunatech.com



/localisation/outline

Language localisation

Java application localisation

Seam web application localisation

Extending localisation support

Localisation tools



Language localisation

From Wikipedia, the free encyclopedia



This article **may require [copy-editing](#) for grammar, style, cohesion, tone or spelling**. You can assist by [editing it](#) now. A how-to [guide](#) is available. *(November 2008)*



This article's **[external links](#) may not follow Wikipedia's [content policies](#) or [guidelines](#)**. Please [improve this article](#) by removing excessive or inappropriate external links.


Language localisation^[1] (from the English term "[locale](#)" and abbreviated in the [numeronym](#) "L10N", the 10 being a way of replacing the middle 10 letters of the word) can be defined as the second phase of a larger process ([Internationalisation](#)) of product translation and cultural adaptation (for specific countries, regions, groups) to account for differences in distinct markets. Thus, it is important not to reduce it to a mere [translation activity](#) because it involves a comprehensive study of the target culture in order to correctly adapt the product.

The localisation process is most generally related to cultural adaptation and translation of [software](#), video games and [websites](#), and less frequently to any written translation (although these also involve cultural adaptation processes). The process of localising can be done for regions or countries where people speak different languages, or where the same language is spoken. Just recall the language differences in countries where Spanish is natively spoken (for instance in South America), or where English is the official language (e.g. in the United States, the United Kingdom and the Philippines).

Contents [\[hide\]](#)

- 1 The overall process: internationalisation, globalisation and localisation
 - 1.1 Translation versus localisation
 - 1.2 Globalization versus localisation
- 2 Language tags and codes
- 3 Notes and references
- 4 See also
- 5 External links

The overall process: internationalisation, globalisation and localisation [\[edit\]](#)

As the [LISA association](#)  explains, globalization "can best be thought of as a cycle rather than a single process". To *globalise* is to plan in advance the way the product or the website should be designed and developed in order to avoid costs from going up and quality problems from emerging, to save time, and eventually to smooth the localising effort for each region/country. Localisation is one phase, but an integral part, of the overall process called **globalisation**.

In this view, there are two primary technical processes that comprise globalization, **internationalisation** and **localisation**, which make up a double-phase process.



WIKIPEDIA
The Free Encyclopedia

- navigation
- [Main page](#)
 - [Contents](#)
 - [Featured content](#)
 - [Current events](#)
 - [Random article](#)

search

- interaction
- [About Wikipedia](#)
 - [Community portal](#)
 - [Recent changes](#)
 - [Contact Wikipedia](#)
 - [Donate to Wikipedia](#)
 - [Help](#)

- toolbox
- [What links here](#)
 - [Related changes](#)
 - [Upload file](#)
 - [Special pages](#)
 - [Printable version](#)
 - [Permanent link](#)
 - [Cite this page](#)

- languages
- [한국어](#)

/localisation/introduction

Internationalisation

I012345678901234567N

I 18 N

I18N removes locale-specific code

Localisation

L0123456789N

L 10 N

L10N adds locale-specific versions

I18N/L10N is *mostly* about text



java.util.Locale

- >An identifier that represents a world region
- >Locale(String language)
- >Locale(String language, String country)
- >Locale(String language, String country, String variant)

java.util.ResourceBundle

- >Contains Locale-specific objects, e.g. text
- >Use ResourceBundle or implement your own class



/localisation/java/messages

```
# phrasebook.properties (English)
hello = Hi, a beer please
```

```
# phrasebook_ca.properties (Catalan)
hello = Hola, una cervesa si us plau
```

```
// Get Locale-specific resource bundles, using a country- and language-specific Locale.
```

```
final Locale englishUK = Locale.UK;
final ResourceBundle englishResources = ResourceBundle.getBundle("phrasebook", englishUK);

final Locale catalanSpain = new Locale("ca", "ES");
final ResourceBundle catalanResources = ResourceBundle.getBundle("phrasebook", catalanSpain);
```

```
// Hi, a beer please
System.out.println(englishResources.getString("hello"));
```

```
// Hola, una cervesa si us plau
System.out.println(catalanResources.getString("hello"));
```



/localisation/java/formats

java.util.Properties

>Implements the line-oriented properties file format

java.text.MessageFormat

>Provides language-neutral construction of text with placeholders

java.text.ChoiceFormat

>Is a NumberFormat that formats number ranges as text

>e.g. format '42' as '42' or 'The answer to life, the...'



/localisation/java/java.util.Properties

```
# Properties file, as parsed by java.util.Properties.load(Reader)

! Comments have a hash or exclamation mark as the first non-white space; blank lines are ignored

# It is conventional, but not required, to use dots to organise keys
key.1=value one

# In keys, leading and trailing white space is ignored
key.2 = value two

# The separator may also be a colon
key.3 : value three
key.4:value four

# You can use either separator in the key if you escape it.
key\:with\=separators.5 = value five

# If there is only a key, the value is the empty string ""
key.6

# Long lines may be broken with a backslash; leading white space on the next line is ignored
first.hundred = 3141592653589793238462643383279502884197169399375105820974944592307816406286\
208998628034825342117067
```

/localisation/java/java.text.MessageFormat

```
value.plain = Beer please

# {0} is a plain placeholder that specifies a parameter index

# {0} (String) name of a beer
value.MessageFormat.none = A bottle of {0} please

# {1,number} is a placeholder that specifies the MessageFormat 'number' format type

# {1} (Number) the number of bottles required
value.MessageFormat.number = {1,number} bottles of {0} please

# {1,choice,formatStyle} specifies a ChoiceFormat for {1} (Number)
value.MessageFormat.choice = \
    {1,choice,1#One bottle please|1<{1,number} bottles please|6<Just keep them coming}

# ... which is formatted as (for numbers 1, 2, 10):
# One bottle please
# 2 bottles please
# Just keep them coming

# {2,time,short} specifies the 'time' format type and the 'short' format style

# {2} (Date) when happy hour starts
value.MessageFormat.time = Happy hour starts at {2,time,short}

# {3} (Date) when happy hour starts
value.MessageFormat.date = There is live music on {3,date,full}
```

String behaviour depends on Locale and char

> " 漆 ".length() == 2 // Unicode code point > \uFFFF

> String.toUpperCase(Locale) needed for i18n, e.g.

"i".toUpperCase(new Locale("tr")).equals("İ") // Dotted I

"I".toLowerCase(new Locale("tr")).equals("ı") // Dotless i

JDK classes are not localised in all languages

> TimeZone, date formats, number formats, Locale

> Use SPIs to plug-in your own localisations

Use a java.text.Collator for sorting



/localisation/java/problems

Properties files use ISO-8859-1 encoding

>Use Unicode escapes, e.g. \u00e9

>UTF-8 encoding supported in Java 1.6

Use a tool to generate non-English files

>native2ascii generates Unicode escapes

>Write your own tool



JDK MessageFormat API documentation humour:

- > 'The rules for using quotes within message format patterns unfortunately have shown to be somewhat confusing. In particular, it isn't always obvious to localizers whether single quotes need to be doubled or not. Make sure to inform localizers about the rules, and tell them (for example, by using comments in resource bundle source files) which strings will be processed by MessageFormat'
- > We just use 'curly' quotes instead of 'single' quotes



javax.servlet.ServletRequest.getLocale()

- >The Servlet API constructs a Locale object from the Accept-Language HTTP headers in an HTTP request.
- >Users can set language preferences in the web browser or operating system
 - ... but sometimes do not, in which case you add a language preference to your web application.



`/localisation/java/conclusions`

`# Localisation API has not changed for years`

`# The basic functionality works well`

`>... even if it isn't pretty`

`# Beware of the details:`

`>many APIs are Locale-dependent`

`>message file text encoding`

`>MessageFormat syntax rules`



/localisation/seam

Seam 'locale' component

- > java.util.Locale instance
- > Set from HTTP request, faces-config.xml or default
- > Set programmatically using 'localeSelector'

Seam 'messages' component

- > Seam resource bundle loads from multiple bundles
- > 'messages' component provides Map interface to resources



/localisation/seam/configuration

```
<!-- faces-config.xml - JSF configuration -->
<faces-config>
  <application>
    <message-bundle>messages</message-bundle>
    <!-- Configure Locales that will be available to Seam's localeSelector --->
    <locale-config>
      <default-locale>en</default-locale>
      <supported-locale>en</supported-locale>
      <supported-locale>fr</supported-locale>
    </locale-config>
    <!-- ... --->
  </application>
</faces-config>
```

```
<!-- components.xml - Seam components configuration -->
<core:resource-loader>
  <core:bundle-names>
    <!-- Note that Seam can combine multiple properties files in its single resource bundle -->
    <value>api</value>
    <value>external</value>
    <value>messages</value>
    <value>ui</value>
  </core:bundle-names>
</core:resource-loader>
```



/localisation/seam/org/jboss/seam/international/Messages.java

```
/**
 * Factory for a Map that contains interpolated messages defined in the Seam ResourceBundle.
 */
@Scope(ScopeType.STATELESS)
@BypassInterceptors
@Name("org.jboss.seam.international.messagesFactory")
@Install(precedence=BUILT_IN)
public class Messages {
    /**
     * Create the Map and cache it in the EVENT scope. No need to cache
     * it in the SESSION scope, since it is inexpensive to create.
     *
     * @return a Map that interpolates messages in the Seam ResourceBundle
     */
    @Factory(value="org.jorg.jboss.seam.international.messages", autoCreate=true, scope=EVENT)
    public Map<String, String> getMessages()
    {
        return createMap();
    }

    /**
     * @return the message Map instance
     */
    public static Map<String, String> instance() {
        if ( !Contexts.isSessionContextActive() ) {
            throw new IllegalStateException("no event context active");
        }
        return (Map<String, String>) Component.getInstance(
            "org.jboss.seam.international.messages", true);
    }
}
```



```

protected Map createMap()
{
    // AbstractMap uses the implementation of entrySet to perform all its
    // operations - for a resource bundle this is very inefficient for keys
    return new AbstractMap<String, String>()
    {
        private java.util.ResourceBundle bundle = SeamResourceBundle.getBundle();

        @Override
        public String get(Object key)
        {
            if (key instanceof String)
            {
                String resourceKey = (String) key;
                String resource=null;
                if (bundle!=null)
                {
                    try
                    {
                        resource = bundle.getString(resourceKey);
                    }
                    catch (MissingResourceException mre)
                    {
                        //Just swallow
                    }
                }
                return resource==null ? resourceKey : resource;
            }
            else
            {
                return null;
            }
        }
    }
}

```



/localisation/seam/facelets

```
# messages.properties - Java properties file, localised text for messages used in Java classes.  
orders = Orders  
command.orders.edit = Edit order details
```

```
# messages_fr - French localisation  
orders = Commandes  
# \u00C9 'E' with acute accent  
# \u00E9 'e' with acute accent  
command.orders.edit = \u00C9diter les d\u00E9tails de la commande
```

```
<!-- facelets-view.xhtml -->
```

```
<!-- The messages component has a java.util.Map interface, so you can deference by dot -->  
<h1>#{messages.orders}</h1>
```

```
<!-- Wrong! Causes javax.el.ELException: /facelets-view.xhtml:  
Property 'orders' not found on type java.lang.String -->  
<h2>#{messages.command.orders.edit}</h2>
```

```
<!-- Right! Message key that contains dots -->  
<h2>#{messages['command.orders.edit']}</h2>
```



Using MessageFormat in attribute values

>Problem: JSF only provides an h:outputFormat tag

Using JSF converters

>Problem: MessageFormat does not work with converters

JSF components embedded in messages

>Problem: you can interpolate a String, not a Component



/localisation/seam/facelets/placeholders in messages

```
# messages.properties (English) - message containing a placeholder
# {0} is the user name
notify.user = Notify {0}
```

```
# messages.properties (Dutch)
notify.user = {0} attenderen
```

```
<!-- Format the message using placeholders -->
<h:outputFormat value="#{messages['notify.user']}">
  <f:param value="#{user.name}"/>
</h:outputFormat>
```

```
<!-- Problem: we cannot use h:outputFormat to format an attribute value. -->
<h:commandLink action="#{action.notify}" value="Notify Peter"/>
```

```
<!-- Too verbose to be a solution - put the value as the tag body: -->
<h:commandLink action="#{action.notify}">
  <h:outputFormat value="#{messages['notify.user']}">
    <f:param value="#{user.name}"/>
  </h:outputFormat>
</h:commandLink>
```

```
<!-- Solution: an Expression Language function that does message look-up. -->
<h:commandLink action="#{action.notify}" value="#{eg:message('user.notify', user.name)}/>
```

```
<!-- Alternative solution: use the 'interpolator' component -->
#{interpolator.interpolate(messages['user.notify'], user.name)}
```



/localisation/facelets/defining a facelets tag library function

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE facelet-taglib PUBLIC "-//Sun Microsystems, Inc.//DTD Facelet Taglib 1.0//EN"
    "http://java.sun.com/dtd/facelet-taglib_1_0.dtd">

<!-- Facelets tag library descriptor that defines tags for use in Facelets views -->

<facelet-taglib>
  <namespace>http://example.com/taglib</namespace>

  <!-- Message function that calls a static method in a Functions utility class. -->
  <function>
    <function-name>message</function-name>
    <function-class>com.example.jsf.Functions</function-class>
    <!-- Params: String message key, Object[] varargs parameters -->
    <function-signature>
      java.lang.String getLocalisedMessage(java.lang.String, java.lang.Object[])
    </function-signature>
  </function>
</facelet-taglib>
```



/localisation/seam/expression language function

```
package com.example.jsf;

import java.text.MessageFormat;
import java.util.MissingResourceException;

import org.jboss.seam.core.ResourceBundle;

public class Functions {

    /**
     * Formats a localised message from the resource bundle for the Seam locale.
     *
     * @param messageKey the key to the formatted message
     * @param params replacement parameters for placeholders
     * @return localised formatted message, or the messageKey
     */
    public static String getLocalisedMessage(
        final String messageKey, final Object... params) {

        try {
            // Get the message from the Seam resource bundle, using the current Seam locale.
            final String message = ResourceBundle.instance().getString(messageKey);
            return MessageFormat.format(message, params);
        }
        catch (final MissingResourceException e) {
            return messageKey;
        }
    }
}
```



/localisation/seam/date-formatting

Next problem: `java.text.MessageFormat` date formatting should be Locale-specific.

{0} user name

```
comment.added1 = Comment posted {0,date}
comment.added2 = Comment posted {0,date,short}
comment.added3 = Comment posted {0,date,dd-MM-yyyy}
```

```
/**
 * Solution: modify the EL function to get the session locale from Seam
 * Use this to construct a locale-specific java.text.MessageFormat
 */
public static Locale getSeamLocale() {
    final org.jboss.seam.Component localeComponent = (org.jboss.seam.Component) Contexts
        .lookupInStatefulContexts("org.jboss.seam.core.locale.component");

    if (localeComponent != null) {
        final org.jboss.seam.core.Locale seamLocale =
            (org.jboss.seam.core.Locale) localeComponent.newInstance();

        if (seamLocale != null) {
            return seamLocale.getLocale();
        }
    }
    return Locale.ENGLISH;
}
```



/localisation/seam/JSF converters

```
# Problem MessageFormat date formatting does not use JSF converters
# e.g. our custom converter that formats dates as 'yesterday', 'today' and 'tomorrow'
comment.added1 = Comment posted {0,date}

# Solution: implement pluggable message format types
# This requires a custom MessageFormat implementation
comment.added1 = Comment posted {0,relativeDate}
```



/localisation/seam/messages containing JSF components

```
<!-- Problem: how do you internationalise a message that contains a JSF component? -->

<!-- See the related items. -->
<p>See the <h:commandLink action="#{action.showRelated()}">related items</h:commandLink>.</p>

<!-- Option 1: vote (and wait) for JSF 2.0 specification issue 517:
      https://javaserverfaces-spec-public.dev.java.net/issues/show_bug.cgi?id=517 -->

<!-- Option 2: implement a custom JSF component whose tag supports nested facets,
      to replace h:outputFormat -->
<eg:componentMessage key="items.showRelated">
  <f:facet name="link">
    <!-- Don't forget that a f:facet may only have one immediate child JSF component -->
    <h:commandLink action="#{action.showRelated()}">
      #{messages['items.showRelated.link']}
    </h:commandLink>
  </f:facet>
</eg:componentMessage>
```

```
# {link} the link text
items.showRelated = See the {link}.
items.showRelated.link = related items
```



/localisation/seam/conclusions

Seam Locale/ResourceBundle support is good

MessageFormat deserves JSP's fate

JSF-oriented MessageFormat replacement needed



Translators are not programmers

- > You could try explaining properties files, Unicode escapes, ISO-8859-1 encoding, version control, and MessageFormat's quote syntax.
- > Good luck with that

You need a collaboration tool

- > Parse and generate Java properties files
- > Multi-user versioning and collaboration features



/localisation/tools/solution

Translators want a spreadsheet user-interface

Wikis have transformed online collaboration

- >enable updates from many people

- >versioning and public edit history

- >discussion and comments in-line

- >no excuse for sending unversioned files by e-mail

We obviously need another Seam application...



User account actions

accountAction.close.success

Account {0} closed

Created 01-12-2008 16:11

Spanish translations (1/3)

no translation

Updated 01-12-2008 16:11

accountAction.timeZone.howToSet

To set your display time zone, select Account > Edit account, choose a Display time zone and Save Changes

Created 01-12-2008 16:11

no translation

Updated 01-12-2008 16:11

accountAction.timeZone.notSet

You have not chosen a time zone to use for dates and times. Dates and times are displayed in the `#{userTimeZone.displayName}` time zone, unless otherwise specified.

Created 01-12-2008 16:11

no translation

Usted no ha elegido una zona horaria para utilizar por fechas y épocas

Updated 01-12-2008 16:11

or [Cancel](#)



```
# curl -X POST --user client:xxxxxxx -H "Content-Type: text/plain" \  
> -T src-static/visiblelogistics.war/WEB-INF/classes/ui.properties \  
> http://api.tolkie.com/product/VisibleLogistics  
INFO Updated command.resource.placementParent.select.help  
INFO Updated command.sort.c  
INFO Updated mail.createOrder.p2  
INFO Updated mail.createOrder.p4  
INFO Updated mail.createOrder.p6  
INFO Updated mail.createOrder.p9  
INFO Updated model.Contact.key.help  
INFO Updated model.Contact.key.help.short  
INFO Updated model.Contact.organisationUrl.help  
INFO Updated model.Order.consignments.help  
INFO Updated model.Order.reference.help  
INFO Updated model.Person.organisationUrl.help  
INFO Updated model.Placement.reference.help  
INFO Added sms.order.status 'Order {0} status update: {1} - {2} {3}'  
INFO Updated ui.account.close.p3  
INFO Updated ui.catalogue.confirmDelete  
INFO Updated ui.catalogue.items.help  
INFO Updated ui.component.selectContact.help  
INFO Updated ui.contacts.delete.confirm  
INFO Updated ui.contacts.existing  
INFO Updated ui.contacts.existing.key  
INFO Updated ui.contacts.existing.organisationName  
INFO Updated ui.incoterm.deq.help  
INFO Updated ui.incoterm.exw.help  
INFO Updated ui.incoterm.fca.help  
INFO Updated ui.order.delete.confirm  
INFO Updated ui.order.feed.help  
INFO Updated ui.order.list.help  
INFO Updated ui.resource.exportDetails.help  
INFO Updated ui.resource.ownership  
INFO Updated ui.resource.ownership.everyone
```































































```
# curl --user lunatech:xxxxxxx http://api.tolkie.com/product/VisibleLogistics/nl/action.contact  
# Message file generated by Tolkie.com - 31 of 33 messages translated.
```

```
tolkie.groups=action.contact
```

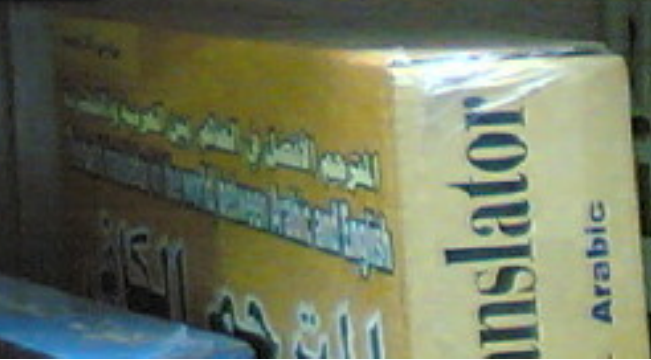
```
action.contact.account.close.success=Account {0} gesloten  
action.contact.added={0} is toegevoegd aan uw contactlijst  
action.contact.added.existing={0} is toegevoegd aan uw contactlijst ({1} heeft reeds een account)  
action.contact.added.invitation={0} is toegevoegd aan uw contactlijst (ter vervanging van {1})  
action.contact.none=U heeft geen contactpersonen.  
action.contact.hide.success={0} is verborgen  
action.contact.show.success={0} is weer zichtbaar in uw contactlijst  
action.contact.confirm.account.updated=Uw accountgegevens zijn met succes bijgewerkt  
action.contact.confirm.account.newTimeZone=Tijdstippen worden nu in de {0} tijdzone weergegeven  
action.contact.confirm.codeInvalid=Activatielink is niet meer geldig  
action.contact.confirm.codeMissing=Autorisatiecode niet opgegeven  
action.contact.confirm.creator={0} heeft uw e-mail adres reeds opgegeven  
action.contact.confirm.eMailAddress.updated=E-mail adres succesvol gewijzigd naar {0}  
action.contact.confirm.invalidConfirmation=The gebruikte link is niet meer geldig  
action.contact.confirm.logoutFirst=Logt u eerst uit voor u een nieuw account activeert.  
action.contact.confirm.password.noMatch=The ingevoerde wachtwoorden komen niet overeen  
action.contact.confirm.signup=Uw e-mail adres is met succes geverifieerd  
action.contact.delete.replacementNotFound=Vervanging {0} bestaat niet  
action.contact.delete.replacementRequired=Vervanging is vereist  
action.contact.mail.sent=Bevestiging e-mail verstuurd naar {0}  
action.contact.mail.not.sent=Kan geen e-mail sturen naar {0}. Probeer u het later nog eens.  
action.contact.mail.exists=Het e-mail adres \u2018{0}\u2019 heeft al een account  
action.contact.mail.change.notSent=Kan geen bevestigings e-mail sturen naar {0}.  
action.contact.invite.addedContact={0} ({1}) toegevoegd aan contactlijst
```



Groups		Spanish	Portuguese	French	Dutch
All groups	1173	 0 of 1173 messages	 0 of 1173 messages	 99% 1057 of 1072 messages	 20% 229 of 1170 messages
action	11	 0 of 11 messages	 0 of 11 messages	 100% 11 of 11 messages	 100% 11 of 11 messages
action.catalogue	18	 0 of 18 messages	 0 of 18 messages	 100% 18 of 18 messages	 100% 18 of 18 messages
action.contact	33	 0 of 33 messages	 0 of 33 messages	 100% 33 of 33 messages	 100% 33 of 33 messages
action.jira	10	 0 of 10 messages	 0 of 10 messages	 100% 10 of 10 messages	 100% 10 of 10 messages
action.order	41	 0 of 41 messages	 0 of 41 messages	 100% 41 of 41 messages	 100% 41 of 41 messages
action.resource	34	 0 of 34 messages	 0 of 34 messages	 100% 34 of 34 messages	 0 of 34 messages
authenticator	2	 0 of 2 messages	 0 of 2 messages	 100% 2 of 2 messages	 100% 2 of 2 messages
command	113	 0 of 113 messages	 0 of 113 messages	 100% 112 of 112 messages	 6% 7 of 113 messages
com.visiblelogistics.action	7	 0 of 7 messages	 0 of 7 messages	 100% 7 of 7 messages	 100% 7 of 7 messages
field	2	 0 of 2 messages	 0 of 2 messages	 100% 2 of 2 messages	 100% 1 of 1 messages
historyChangeListener	27	 0 of 27 messages	 0 of 27 messages	 100% 25 of 25 messages	 0 of 27 messages
javax.faces	1	 0 of 1 messages	 0 of 1 messages	 100% 1 of 1 messages	 0 of 1 messages
kml	15	 0 of 15 messages	 0 of 15 messages	 100% 15 of 15 messages	 0 of 15 messages
mail	38	 0 of 38 messages	 0 of 38 messages	 100% 38 of 38 messages	 8% 3 of 38 messages



<http://www.flickr.com/photos/forzaq8/19397654/>



peter.hilton@lunatech.com

www.lunatech.com

employment@lunatech.com



LUNATECH
RESEARCH