

# Localization Rules for HP Service Manager

Preface.....	2
Localization Rules for Software.....	2
Properties files for format, message and help on field.....	2
Properties files for the clients.....	4
JS and HTML files.....	4
How to find strings in the application.....	5
How to find Help on Field messages in the application.....	5
Strings that must remain in English.....	6

# Preface

This document describes the localization rules for HP Service Manager. Be sure to adhere to these rules and localize language and words only when allowed.

The product name “Service Manager” should always be in English in all cases. Module names such as Incident Management, and Problem Management should be localized according to ITIL standards or left in English, depending on market requirements.

## Localization Rules for Software

### Properties files for format, message and help on field

The strings extracted from format, message and help on field are placed in the `format.properties`, `message.properties`, and `help.properties` files, respectively. When you translate strings from format, message, and help on field, you must follow these rules:

- *Files that return from translation and need to be imported must be encoded in ANSI (not Unicode)*

For example, the English string “Properties” should appear as: “Propri\u00e9t\u00e9s” (which is encoded in ANSI) rather than the native string in French: “Propriétés” (which is encoded in Unicode).

**Note:** To convert a file with native strings to ASCII encoding (in the `/u####` representation) and vice versa, you can use the “native2ascii” tool provided in the Java JDK. More information on this tool is available at:

**<http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/native2ascii.html>**

- *Use hyphens instead of n-dashes*

Use hyphens instead of n-dashes when localizing strings such as: `FORMAT.STR/AlertDef.g\13\0\0\0\Caption=$L.slo.expiration - SLO expiration date/time.`

If you use an n-dash (i.e. “`$L.slo.expiration - SLO expiration date/time`”), this character will display as a symbol (something like a vertical rectangle).

- *Use a backslash before double quotes*

If you need to insert double quotes within a string to identify GUI terms, use an escaping backslash before the double quotes.

**Warning:** *If we do not have backslashes preceding the double quotes, the string will be truncated, giving us error while re-importing the files into Translation Editor.*

The following is an example:

If we write:

```
<span class=FormTitle>Recherche $$(_DocumentLabel)</span><br> Entrez  
autant de valeurs que vous le souhaitez pour affiner la recherche, puis  
appuyez sur le bouton "Rechercher".
```

The string will be truncated and will display as:

```
<span class=FormTitle>Recherche $$(_DocumentLabel)</span><br> Entrez  
autant de valeurs que vous le souhaitez pour affiner la recherche, puis  
appuyez sur le bouton
```

In order to avoid this, we need to write the string like this:

```
<span class=FormTitle>Recherche $$ (_DocumentLabel)</span><br> Entrez  
autant de valeurs que vous le souhaitez pour affiner la recherche, puis  
appuyez sur le bouton \"Rechercher\".
```

**Important:** *If double quotes are already included in the strings, please do not delete them.*

- *If the strings contain logic or variables, the logic should not be broken and the variables should not be translated*

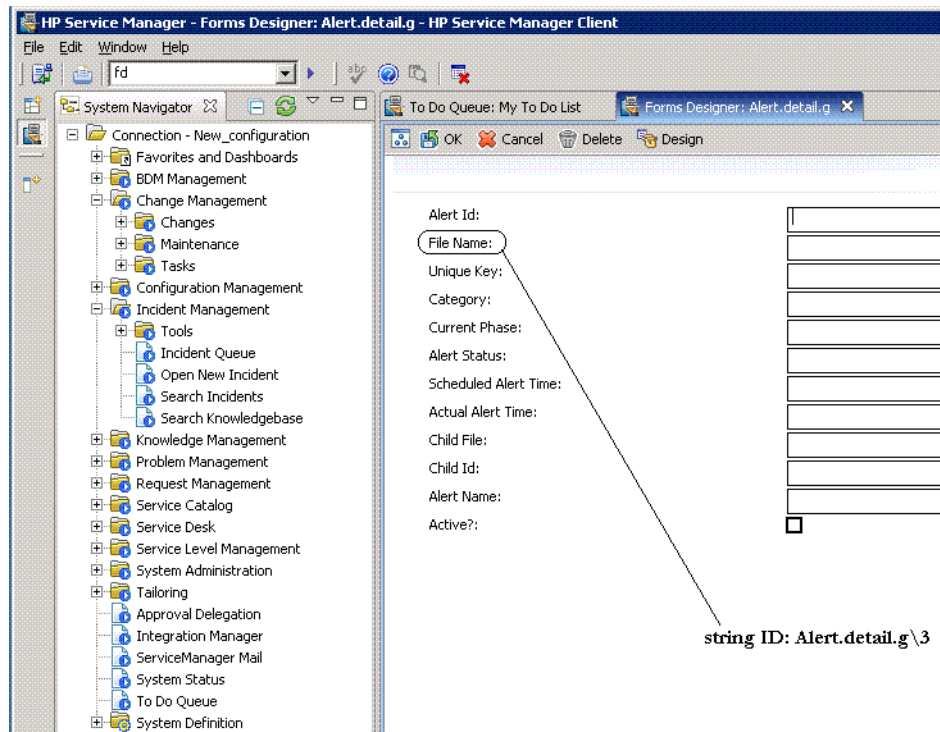
The following are some examples:

- “<word>;<word>:\:<word>” should be translated as “<trans>;<trans>:\:<trans>”.
- “&<char>” or “\$<word>” should remain untranslated.
- Strings that contain parameters like {0} can be translated, but the variable representation must remain unchanged - {0}.

For example: “{0} files were found in folder {1}” will be translated to Chinese as “{1} 文件夹中发现了 {0} 个文件”.

- *Do NOT add special characters unless they appear in the English strings*
- *Translate the output and caption strings for each label exactly the same way*

For each label in a form, we have two strings: output and caption. The two strings must be translated exactly the same way. The following screenshots illustrate an example string ID: Alert.detail.g\3. The first part of the string corresponds to the actual form, the numbers correspond to the position within the form (x and y coordinates).



```

"Alert.detail.g\1\1\0\0\output", "Alert Id"
"Alert.detail.g\1\1\0\0\Caption", "Alert Id"
"Alert.detail.g\3\1\0\0\output", "File Name"
"Alert.detail.g\3\1\0\0\Caption", "File Name"
"Alert.detail.g\5\1\0\0\output", "Unique Key"
"Alert.detail.g\5\1\0\0\Caption", "Unique Key"
"Alert.detail.g\7\1\0\0\output", "Category"
"Alert.detail.g\7\1\0\0\Caption", "Category"
"Alert.detail.g\9\1\0\0\output", "Current Phase"
"Alert.detail.g\9\1\0\0\Caption", "Current Phase"
"Alert.detail.g\11\1\0\0\output", "Alert Status"
"Alert.detail.g\11\1\0\0\Caption", "Alert Status"
"Alert.detail.g\13\1\0\0\output", "Scheduled Alert Time"
"Alert.detail.g\13\1\0\0\Caption", "Scheduled Alert Time"
"Alert.detail.g\15\1\0\0\output", "Actual Alert Time"
"Alert.detail.g\15\1\0\0\Caption", "Actual Alert Time"
"Alert.detail.g\17\1\0\0\output", "Child File"
"Alert.detail.g\17\1\0\0\Caption", "Child File"
"Alert.detail.g\19\1\0\0\output", "Child Id"
"Alert.detail.g\19\1\0\0\Caption", "Child Id"
"Alert.detail.g\21\1\0\0\output", "Alert Name"
"Alert.detail.g\21\1\0\0\Caption", "Alert Name"
"Alert.detail.g\23\1\0\0\output", "Active?"
"Alert.detail.g\23\1\0\0\Caption", "Active?"
"Alert.g\1\0\0\0\output", "Alert"
"Alert.g\1\0\0\0\Caption", "Alert"
"Alert.g\2\1\0\0\output", "Alert Name"
"Alert.g\2\1\0\0\Caption", "Alert Name"
"Alert.g\4\1\0\0\output", "Alert Status"
"Alert.g\4\1\0\0\Caption", "Alert Status"
"Alert.g\6\1\0\0\output", "Scheduled Alert Time"
"Alert.g\6\1\0\0\Caption", "Scheduled Alert Time"
"Alert.g\8\1\0\0\output", "Actual Alert Time"
"Alert.g\8\1\0\0\Caption", "Actual Alert Time"
"Alert.g\10\1\0\0\output", "Active?"
"Alert.g\10\1\0\0\Caption", "Active?"
"Alert.qbe\1\0\0\0\output", "Alert Name"

```

## Properties files for the clients

Do not to remove \n where in use, otherwise the string will be truncated because sentences do not wrap automatically with the Eclipse client.

## JS and HTML files

Follow the standard rules in the localization industry.

## How to find strings in the application



The string ID consists of the format name and a numbering scheme for the position within the format. The format name can be entered in Forms Designer, a module of Service Manager. In order to do that, follow these steps:

- 1 Start the server using the RUN command and type: **sm**.
- 2 Start the Service Manager Client (do not close the Command Prompt window).
- 3 Select your login language.
- 4 Use **FALCON** for the user name.
- 5 In the command line, type **fd**. The Forms Designer window opens.
- 6 Enter the full format name and select **OK** to search for a specific format.
- 7 Enter **\*.g** to search for all formats with the ending ".g" or enter any letter to search for all formats starting with that letter.

## How to find Help on Field messages in the application

Field Help enables the user to learn more about an individual field in a form and the value to specify in the field.

To view Field Help, perform the following steps:

- 1 Navigate to any Service Manager form.
- 2 Insert the cursor in any field or table cell.
- 3 Do one of the following:
  - Press **CTRL+H**.
  - Click **Help > Help on Field** .
  - Click **Help** on the toolbar .

## Strings that must remain in English

See the following table.

Type	Examples
Event names	rmqin rmlin rmoin
Predefined system events	alerts recal alerts reset approval approval reeval
boolean expressions	UNKNOWN TRUE FALSE
parameter names	analyzelog lockanalysis lockslog
RAD debugger commands	d (display) t (trace) ta tar tt b (breakpoint) ba (ba menu.manager) bt (bt rinit) c (continue) s (step) gl (globals) v (variables) l (locals) sta (stack) re (relations) m (memory) h (help)
Display list variables	\$G.init.impact.disp \$G.urgency.disp

© 2010 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

May 2010

