

Translation **TDS-Gaia**

Also available in *Inglês*, *Português*

The **TDS-Gaia** uses English as the default language and we need collaboration for translation and revision of the translated material, which can be:

- Text files, identified with the extension **MD**;
- Messages and texts used in the source files.

The review can be orthographic, grammatical and procedural. Make the one you feel most comfortable with.

Translation or review of texts (text files)

Procedures

- Clone the **TDS-Gaia** project, preferably the **Dev** branch. Optionally, transfer it to your workstation;
- Duplicate the file to be translated according to the environment (*web* or *local*), adding the extension **.<locale>.MD**. Where, **<locale>** is the language code of [supported by VSCode] (https://code.visualstudio.com/docs/getstarted/locales#_available-locales);
- Open the file for editing according to the environment;
- Do the translation or review;
- At the end of the process:
 - If the environment is local, confirm the changes made;
 - Request the reintegration of your clone to the original project.

Recommendations to translators and reviewers

- Avoid using foreign terms. Use it only when there is no translation or to improve understanding, italicizing or translating and placing the original in parentheses. For example, "mouse click" is "acione o *mouse*" or "acione o rato (*mouse*)";
- Before starting the translation, read the texts already translated by other collaborators. That way, you get used to the terms used and maintain a language standard;
- In case of links (*links*), check if there is a file with the translation for the language being work and adjust it, otherwise, keep the original;
- Do not translate commands or codes. Usually these will be in the formatted as code, which is indicated by the mark ``;
- Do not translate names of products, users, brands and the like, unless directed to do so.
- The names of files and folders must always be in English.
- When using acronyms that are repeated in the text, in the first occurrence of this, put it in full and the acronym in parentheses. For example, "The Protheus Object Repositories (RPO, *Repository Protheus Objects*) is used for ...".
- Whenever possible, adjust the text to the learned standards of the language.
- In the case of numerals, use the rules/recommendations of the working language. In the case of Portuguese, numbers (cardinal or ordinal) up to ten, one hundred, one thousand, at the beginning

of sentences and fractional numbers (two thirds, one quarter), must be in full.

- In messages for translation, a number can appear between braces, e.g. `{0}` **attribute required**.. These keys indicate that it is an argument that will be used in the final presentation and should be considered in the translation and placed in the correct position of the translated text.

Translation or review of messages and texts (source files)

Recommendations to the developer

- The default language is English, so any *string* must be written in English, regardless of whether it will be translated or not;
- All *strings* to be translated must be used with the `vscode.l10n.t`;
- Try to use generic text and if necessary with arguments. For example, instead of writing:

```
typescript ... if (productCode === '') { msgErro.push (locate ("productCode",  
"Required product code.)) ) } if (productName === '') { msgErro.push (find  
("productName", "Required product name.)) ) console.log (`Validation with $  
{msgErro.length} errors`); ...
```

Write:

```
typescript ... if (productCode === '') { msgErro.push (vscode.l10.t("Required  
product code.)) ) } if (productName === '') { msgErro.push  
(vscode.l10.t("Required product name. Product: {0}", productName)) } console.log  
(vscode.l10.t("Validation with {0} errors", msgErro.length)); ... ..
```

- In longer texts, you must use the `vscode.l10.t` per paragraph;
- Do not use variables of any type or scope, instead of the *string* to be translated. For example:

```
typescript const ATT_REQ = "{0} attribute required."; ... if (productCode ===  
'') { msgErro.push (locate ("ATTRIBUTE_REQUIRED", ATT_REQ, locate  
("productCode", "Product Code"))); } if (productName === '') { msgErro.push  
(locate ("ATTRIBUTE_REQUIRED", ATT_REQ, locate ("productName", "Product  
Name"))); } ...
```

The code snippet above is wrong and will not work in the translation process. The correct is:

```
typescript const ATT_REQ = (attribute) => vscode.l10.t ("{0} attribute  
required.", attribute); ... if (productCode === '') { msgErro.push (ATT_REQ  
(vscode.l10.t("Product Code"))); } if (productName === '') { msgErro.push  
(ATT_REQ (vscode.l10.t("Product Name"))); } ...
```

Support VSCode extensions (local only)

These extensions can help you in the process:

- [Code Spell Checker] (<https://marketplace.visualstudio.com/items?itemName=streetsidesoftware.code-spell-checker>) and the dictionary extension for the language being worked on.
- [VSCode Google Translate] (<https://marketplace.visualstudio.com/items?itemName=funkyremi.vscode-google-translate>).