Contents

[**Arabic languages - RTL Install** 1](#_Toc192443938)

[Tcash.ini / osf.ini file 1](#_Toc192443939)

[Language file Label ID 3738 1](#_Toc192443940)

[Arabic languages - List of RTL (Right to Left) languages 1](#_Toc192443941)

[The 12 right-to-left languages: 2](#_Toc192443942)

[**Previous development - Jordan** 3](#_Toc192443943)

[Fixed Missing Translations 3](#_Toc192443944)

[Jordan Accounting resources 3](#_Toc192443945)

[Jordan Tax rates 4](#_Toc192443946)

[LTR - Interface (Default) 4](#_Toc192443947)

[RTL - Switch 5](#_Toc192443948)

[RTL - Interface 5](#_Toc192443949)

[Arabic screens 6](#_Toc192443950)

[Widestring Errors 6](#_Toc192443951)

[Let osFinancials help to create a Set of Books (Default) 6](#_Toc192443952)

[Fixed - Export (Setup ribbon) 7](#_Toc192443953)

[Unicode context menu options of forms 8](#_Toc192443954)

# Reportman - https://reportman.sourceforge.io/ can it handle RTL languages and what should be done to display properly in reports?

ReportMan is a reporting tool that supports multiple languages, including right-to-left (RTL) languages like Arabic, Hebrew, and Persian. However, proper display of RTL languages in reports requires specific configuration and considerations. Here's what you need to do to ensure RTL languages are displayed correctly in ReportMan:

**1. Set the Report Language and Alignment**

* Ensure the report is configured to recognize RTL text. This includes setting the text alignment to right-aligned for RTL languages.
* In ReportMan, you can set the alignment property of text elements (e.g., labels, fields) to Right or RTL mode.

**2. Use a Compatible Font**

* Choose a font that supports RTL scripts. Common fonts for RTL languages include:
  + Arabic: Arial, Times New Roman, Tahoma
  + Hebrew: Arial, David, Times New Roman
  + Persian: Nazanin, Mitra, Arial
* Ensure the font is installed on the system where the report is generated.

**3. Enable RTL Layout in the Report**

* Some reporting tools require explicit enabling of RTL layout for the entire report or specific sections. Check if ReportMan has an option to set the report or section direction to RTL.
* If ReportMan does not have a built-in RTL setting, you may need to manually adjust the layout and alignment of elements.

**4. Handle Text Encoding**

* Ensure the text encoding is set to UTF-8 or another encoding that supports RTL characters. This ensures that characters are displayed correctly and not as garbled text.

**5. Test with Sample Data**

* Create a sample report with RTL text and test it thoroughly. Check for issues such as:
  + Incorrect text alignment
  + Misplaced punctuation or numbers
  + Broken text flow
* Adjust the layout and settings as needed.

**6. Consider Bidi (Bi-directional) Text**

* If your report contains mixed LTR (left-to-right) and RTL text (e.g., English and Arabic), ensure the tool handles bi-directional text correctly. This may require additional configuration or manual adjustments.

**7. Check for RTL Support in Export Formats**

* If you export the report to formats like PDF, HTML, or Excel, verify that RTL text is preserved and displayed correctly in the exported file.

**8. Consult ReportMan Documentation or Community**

* If you encounter issues, refer to the [ReportMan documentation](https://reportman.sourceforge.io/) or seek help from the community. There may be specific settings or workarounds for RTL support.

By following these steps, you should be able to handle RTL languages in ReportMan and ensure they display properly in your reports. If ReportMan lacks native RTL support, you may need to explore alternative reporting tools with better RTL handling capabilities.

# In Arabic should the layout of objects and expressions not be changed for example code description etc. not be displayed or printed as description, code, etc.

When working with **Arabic** or other **right-to-left (RTL)** languages, the layout of objects and expressions in reports must be carefully adjusted to ensure proper readability and alignment. In Arabic, the natural reading order is from right to left, and this affects not only the text but also the layout of elements like **code**, **description**, and other fields.

Here’s how you should handle the layout of objects and expressions in Arabic reports:

**1. Text Alignment**

* **Arabic text** should be **right-aligned** within its container (e.g., text boxes, labels, or fields).
* If you have mixed content (e.g., Arabic and English), ensure the alignment respects the dominant language (RTL for Arabic).

**2. Order of Fields**

* In Arabic, the logical order of fields should follow the RTL reading direction. For example:
  + **Incorrect (LTR order):** Code | Description
  + **Correct (RTL order):** Description | Code
* This means you may need to **rearrange the layout** of objects in your report to match the RTL flow.

**3. Expressions and Concatenation**

* When concatenating fields or expressions, ensure the order respects the RTL logic. For example:
  + If you have a field like Description + " - " + Code, it should be reversed for Arabic to Code + " - " + Description.
* Use conditional logic in your reporting tool to handle this dynamically based on the language.

**4. Numbers and Dates**

* Numbers and dates in Arabic are often written from left to right (e.g., 123 or 2023-10-05), even within RTL text. Ensure these elements are displayed correctly and do not disrupt the RTL flow.
* Some systems support **Arabic-Indic numerals** (e.g., ١٢٣ instead of 123). If needed, configure your report to use the appropriate numeral format.

**5. Punctuation and Special Characters**

* Punctuation marks (e.g., commas, periods, question marks) should appear in their correct position relative to the RTL text. For example:
  + In Arabic, a question mark (؟) is used instead of the standard ?, and it should appear at the **end** of the sentence (left side in RTL layout).
* Ensure your reporting tool handles these characters correctly.

**6. Testing and Validation**

* Test your report thoroughly with Arabic data to ensure:
  + Text is properly aligned and readable.
  + The order of fields (e.g., Description, Code) follows the RTL logic.
  + Numbers, dates, and punctuation are displayed correctly.
* If possible, have a native Arabic speaker review the report for accuracy and readability.

**7. Dynamic Layout Adjustment**

* If your report needs to support both LTR (e.g., English) and RTL (e.g., Arabic) languages, consider implementing **dynamic layout adjustments**:
  + Use conditional formatting or scripting to switch the layout and field order based on the language.
  + For example, in ReportMan, you might use expressions or scripts to rearrange fields or change alignment dynamically.

**Example in ReportMan**

If you are using ReportMan, you can:

1. Set the **alignment** of text elements to Right for Arabic.
2. Rearrange the **order of fields** in the report designer to match the RTL flow (e.g., Description | Code instead of Code | Description).
3. Use **expressions** to handle dynamic text concatenation or formatting based on the language.

**Summary**

In Arabic reports, the layout of objects and expressions should follow the RTL logic. This means:

* **Text alignment** should be right-aligned.
* **Field order** should be reversed (e.g., Description | Code instead of Code | Description).
* **Numbers, dates, and punctuation** should be displayed correctly within the RTL context.

By carefully adjusting the layout and testing with Arabic data, you can ensure that your reports are properly formatted and readable for RTL audiences.

osFinancials/TurboCASH has mostly LTR languages for the Firebird and MSSQL databases. Is it recommended that a separate install be done for RTL language?

When dealing with **osFinancials/TurboCASH** and supporting both **LTR (left-to-right)** and **RTL (right-to-left)** languages, the decision to use a **separate installation** depends on several factors, including the complexity of the application, the database structure, and the level of customization required. Here's a detailed analysis to help you decide:

**1. Database Structure and Language Support**

* **Firebird and MSSQL databases** used by TurboCASH are typically language-agnostic, meaning they can store data in any language, including RTL languages like Arabic.
* However, the **application layer** (TurboCASH) may not natively support RTL layouts or may require significant customization to handle RTL languages properly.

**2. Challenges with Mixed LTR and RTL Support**

* **Layout Issues**: TurboCASH is primarily designed for LTR languages, so the user interface (UI) and reports may not automatically adjust for RTL languages. This can lead to misaligned text, incorrect field ordering, and poor readability.
* **Data Entry**: Input fields for RTL languages may not behave correctly (e.g., cursor movement, text alignment).
* **Reports**: Reports generated by TurboCASH may not handle RTL text properly unless explicitly configured.

**3. Recommendation: Separate Installation for RTL Languages**

Given the challenges, it is **recommended to use a separate installation** of TurboCASH for RTL languages like Arabic. Here's why:

* **Simplified Customization**: A separate installation allows you to customize the UI, reports, and database settings specifically for RTL languages without affecting the LTR setup.
* **Avoid Conflicts**: Mixing LTR and RTL languages in the same installation can lead to layout and formatting conflicts, especially in reports and printed documents.
* **Easier Maintenance**: Maintaining separate installations simplifies updates, backups, and troubleshooting for each language group.

**4. Steps to Set Up a Separate Installation for RTL Languages**

If you decide to use a separate installation, follow these steps:

**a. Install TurboCASH for RTL Languages**

* Install a fresh copy of TurboCASH specifically for RTL languages.
* Configure the application to use the appropriate language settings (e.g., Arabic).

**b. Customize the UI and Reports**

* Adjust the UI layout to support RTL text alignment and field ordering.
* Modify reports to ensure proper RTL formatting (e.g., right-aligned text, correct field order).

**c. Configure the Database**

* Ensure the database (Firebird or MSSQL) is set up to store and retrieve RTL text correctly.
* Use UTF-8 or another Unicode encoding to support RTL characters.

**d. Test Thoroughly**

* Test the application with RTL data to ensure proper functionality, including data entry, reporting, and printing.
* Validate the layout and readability of reports and documents.

**5. Alternative: Single Installation with Conditional Customization**

If a separate installation is not feasible, you can use a **single installation** with conditional customization:

* **Dynamic Layout Adjustment**: Use scripts or conditional formatting to adjust the UI and reports based on the selected language.
* **Language-Specific Templates**: Create separate report templates for LTR and RTL languages.
* **Database Configuration**: Ensure the database supports both LTR and RTL text.

However, this approach requires significant development effort and may still result in inconsistencies.

**6. Best Practices**

* **Consult Documentation**: Review the TurboCASH documentation or community forums for guidance on RTL language support.
* **Engage Developers**: If customization is required, consider hiring developers familiar with TurboCASH and RTL language handling.
* **Test Extensively**: Test the application with real-world RTL data to identify and resolve issues.

**Conclusion**

For **osFinancials/TurboCASH**, a **separate installation** is recommended for RTL languages like Arabic to ensure proper functionality, readability, and maintainability. This approach simplifies customization and avoids conflicts between LTR and RTL layouts. If a separate installation is not possible, consider implementing conditional customization, but be prepared for additional complexity and testing.

**Previous development - Jordan**

one guy i have for 4 hours a day that is someone i think could be good in implementing osf/tc for his county. Thats the one doing the conversion now.

## Fixed Missing Translations

Its a arabic country and we are missing some translations.

Also we need a books for Jordan we need to see if we can find data on that

Also see Missing labels added from English - Language file records recon Added

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Language*** | ***Number of Records (build (292)*** | ***Labels added*** | ***New Number of records*** | ***Duplicate Labels removed*** | ***Comments*** |
| *Arabic* | *3805* | *Added 198 labels 3127 to 3324 + others* | *4012* | *3267        Selling price / Supplier stock code* |  |

Current labels 4033

Missing records should be fixed - only outstanding untranslated labels should need to be done +-1620 Labels to be translated

## Jordan Accounting resources

**Keywords**

Accounting regulation International Financial Reporting Standards Corporate governance Accounting practices Jordan

**Jordan accounting and tax, Jordan legal and compliance (healyconsultants.com)**

- <https://www.healyconsultants.com/jordan-company-registration/accounting-legal/> -

**Accounting Education and Accountancy Profession in Jordan: The Current Status and the Processes of Improvement**

- <https://iiste.org/Journals/index.php/RJFA/article/viewFile/7167/7380> -

**Evaluation of practical accounting education in Jordan**

- <https://www.emerald.com/insight/content/doi/10.1108/HEED-04-2021-0034/full/html> -

**Accounting courses in Jordan**

- <https://courses.laimoon.com/jordan/accounting-finance-and-banking/accounting> -

**International Federation of Accountants**

- <https://www.ifac.org/about-ifac/membership/country/jordan> -

**Financial Reporting Framework in Jordan**

- <https://www.iasplus.com/en/jurisdictions/asia/jordan> -

## Jordan Tax rates

Sales tax Threshold

Annual thresholds:

Goods - JOD50,000

Services - JOD30,000

Sales tax rates

Standard rate = 16%

Reduced rates are 0%, 4%,5% and 10%

Languages

Label ID 3738 = LTR (Left to Right) Default for most countries.

Some countries and Languages that require RTL (Right to Left) Language and Interface support

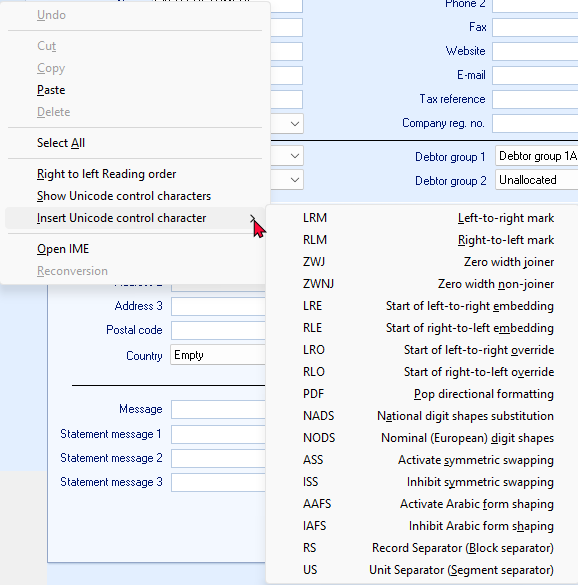
need to change this LTR Left To Right (Label ID 3738) change the LTR with RTL (Right To Left).

If that is RTL mode else not so we can switch language and get the right RTL for the language

Source - <https://lingohub.com/academy/best-practices/rtl-language-list> -

# Unicode context menu options of forms

In osFinancials5.1/TurboCASH5-3, the context menu on most fields in forms, such as [**Debtors**](hnd-topic://7ea26aa8109d4b06988600557f4bad1a), [Creditors](hnd-topic://e784edfabf87435991481c10dcee1e03), [Stock](hnd-topic://b63ce7792fbe4d239b1fe6a867815835) on the **Default** ribbon and [Accounts](hnd-topic://5fdf5282071941c289dc32354fd37707), [Groups](hnd-topic://1ac89184c1994c0a863dfdaed4e44fff) on the **Setup** menu, include the following context menu items related to Unicode (if Unicode is enabled).



These context menu items in osFinancials5/TurboCASH5, specifically related to Unicode, provide advanced text formatting and control options when working with characters beyond the standard ASCII set. Let's break down what each option means and its function:

**1. Right to Left Reading Order**

* **Function:** This option controls the reading direction of the text. When enabled, the text in the field will be displayed and processed from right to left, which is essential for languages like Arabic and Hebrew.

**2. Show Unicode Control Characters**

* **Function:** This option displays invisible Unicode control characters that might be present in the text. These characters can affect text formatting and rendering, so visualizing them can help troubleshoot display issues.

**3. Insert Unicode Control Character**

* **Function:** This allows you to manually insert specific Unicode control characters into the text. These characters are used to fine-tune text direction, shaping, and rendering:
  + **LRM (Left-to-right Mark):** Forces the following characters to be displayed left-to-right.
  + **RLM (Right-to-left Mark):** Forces the following characters to be displayed right-to-left.
  + **ZWJ (Zero Width Joiner):** Connects two characters that might otherwise be separated.
  + **ZWNJ (Zero Width Non-Joiner):** Prevents two characters from being connected.
  + **LRE (Start of Left-to-right Embedding):** Starts a region of left-to-right text within right-to-left text.
  + **RLE (Start of Right-to-left Embedding):** Starts a region of right-to-left text within left-to-right text.
  + **LRO (Start of Left-to-right Override):** Overrides the current text direction and sets it to left-to-right.
  + **RLO (Start of Right-to-left Override):** Overrides the current text direction and sets it to right-to-left.
  + **PDF (Pop Directional Formatting):** Ends a region of embedded text direction.
  + **NADS (National Digit Shapes Substitution):** Uses the digit shapes specific to the current locale.
  + **NODS (Nominal (European) Digit Shapes):** Uses the standard European digit shapes (0-9).
  + **ASS (Activate Symmetric Swapping):** Enables symmetric swapping of characters, which can affect the appearance of certain characters.
  + **ISS (Inhibit Symmetric Swapping):** Disables symmetric swapping of characters.
  + **AAFS (Activate Arabic Form Shaping):** Enables shaping of Arabic characters based on their position in the word.
  + **IAFS (Inhibit Arabic Form Shaping):** Disables shaping of Arabic characters.
  + **RS (Record Separator):** A block separator character.
  + **US (Unit Separator):** A segment separator character.

**4. Open IME**

* **Function:** This opens the Input Method Editor (IME), a tool that allows you to input characters from languages that don't have direct keyboard mappings.

**5. Reconversion**

* **Function:** This option is specific to IMEs. It re-analyzes the input text and may suggest alternative conversions or corrections.

**In Summary**

These Unicode-related context menu items in osFinancials5/TurboCASH5 provide powerful tools for handling text in various languages and scripts. They allow you to control text direction, insert special characters, and fine-tune text rendering to ensure accurate and professional-looking results.