



# Translation Office 3000 3D RTF Templates Guide

© Advanced International Translations

## Table of Contents

1	Templates.....	4
	Templates in TO3000 3D.....	4
	Saving a document for printing.....	4
	Template files.....	5
	Editing templates .....	5
	Template Variables.....	6
2	Logic and syntax of templates.....	7
	Variables .....	7
	Data Scan commands .....	7
	Condition checking .....	8
3	Common template variables.....	10
4	Locale settings and date format.....	10
5	Date and time functions .....	14
6	Project template variables (With Projects Mode only) .....	16
7	Contacts template variables.....	17
8	Quotes template variables.....	18
	Common Quote Variables.....	18
	Taxes.....	19
	Discounts .....	20
	Draft Client Job variables for Quotes to Clients.....	21
9	Client Jobs template variables.....	22
10	Invoice template variables.....	24
	Date and code .....	24
	Invoice totals.....	24
	Taxes.....	25
	Discounts .....	26
	Payment status variables.....	27
	Credit note status variables .....	27
	Jobs variables in invoice.....	28
	Linked payment variables .....	29
	Linked credit note variables.....	30
11	Credit note template variables.....	31

---

Date and Code.....	31
Linked invoice variables.....	31
Credit note totals .....	32
Taxes.....	32
Refund status variables .....	33
Linked refund variables .....	34
12 Payments template variables.....	35
Basic payments template variables.....	35
Linked invoice variables.....	36
13 Refund Template Variables .....	37
Linked credit note variables.....	37

# 1 Templates

## Templates in TO3000 3D

*Template* is an *RTF* ("Rich Text Format") file stored in your TO3000 3D setup folder and used as a template when saving work flow documents in *RTF* (rich text format) or *PDF* (portable document format) files. RTF files can be opened with most text editors, including *MS Word*, and easily converted to more common .doc format.

## Saving a document for printing

The following documents can be saved as printable RTF, PDF and DOC files using their own templates:

- Price lists
- Invoices
- Summaries of payments from clients
- Summaries of projects and client jobs
- Credit notes and refunds
- Quotes

Documents can be saved using windows **Edit Invoice**, **Edit Project** and so on, as well as **Prices** tabs of **Client/Prospect** window, and **General Prices for Clients/Prospects** window.

Any window which has RTF Templates section can be used to save its data in a printable document.

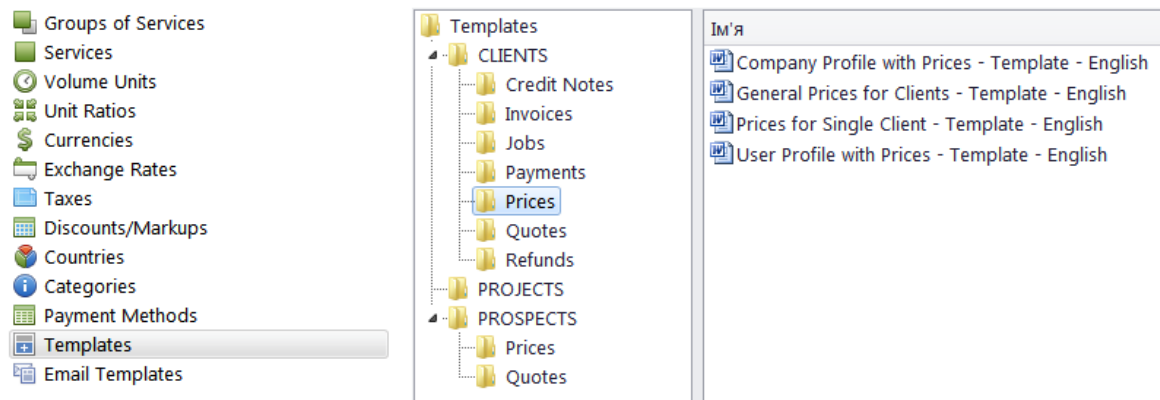
RTF template controls display a drop-down list, which can be used to select different templates.

- Use **RTF Templates** drop-down list to select the Template which must be used for saving the current document in an RTF file.
- Click **Save** button to save template in the respective folder.
- Click **Save&Open** button to open this document in your default text editor immediately after saving it.
- Click **Print** button to quickly send this document to printer.
- Click **Send Mail** to open the Mail sender with the template-based document already attached and ready to send.

**Note:** You can choose to save the output file in *RTF*, *PDF* or *DOC* format by selecting this format in **Save As** window with the help of **Save as type** drop-down list.

## Template files

Each document type (like *invoices*, *purchase orders*, *quotes*, and so on) uses templates stored in specific folder, named by the document type, which templates it stores:




By editing template layout and format, you edit the layout and format of all the documents which will be saved with it. There can be a number of templates to choose from for one document type. Each RTF file in the template folders is considered a template.

## Editing templates

Templates can be opened for editing using the **Templates** section of **TO3000 3D Personal settings**. To edit templates:

1. Click menu **Settings > Personal**.
2. Click **Templates** section of **Personal settings** window.
3. Locate the folder of the document, which templates you wish to edit.
4. Open required template file with text editor (**MS Word** or any other editor supporting RTF format).
5. After making all the necessary changes, save **Template** in the same or new RTF file of the same folder.

 **Note:** Template folders can also be opened with regular file browser, like **Windows Explorer**.


## Template Variables

Variable is a specific symbol combination which is entered in template files. When a document is saved using template, **TO3000 3D** recognizes the variable and inserts required data in the resulting document.

Variables are indicated with "\" (backslash) symbols in the beginning and in the end,

➔ **Example:** \CLIENT\_NAME\, \CLIENT\_STREET1\, \CLIENT\_PHONE1\ etc.

Each variable represents a portion of data which will be inserted in its place when the document is saved.

 **Note:** Template can be edited without changing the variables wording and outlay (fonts and colors can still be changed). Advanced users can learn about templates variables; add/delete them as necessary, customizing templates to the maximum.

## 2 Logic and syntax of templates

### Variables

All variables and commands can be identified by \ (backslash) symbol in the beginning and end. In example below you can see `\CLIENT_NAME\`, `\CLIENT_STREET1\` and `\CLIENT_PHONE1\` variables:

**To:** `\CLIENT_NAME\`

**Address:** `\CLIENT_STREET1\`


**Phone:** `\CLIENT_PHONE1\`

When you use this template by clicking on **Save** or **Print** in some dialog box, `\CLIENT_NAME\` variable will be replaced by name of currently selected *client*, `\CLIENT_STREET1\` — by *client's street address* and `\CLIENT_PHONE1\` — by *client's phone number*. The final output will look like this:

**To:** XYZ Company

**Address:** Elm Street, 1

**Phone:** +1 212 898 11 31

 **Note:** Numerical values sometimes are surrounded by `fnum` command. This command simply tells the parser to round up the value to specified number of digits after decimal point (2 digits in the example below):

**Job Total:**

`\fnum(dtLinkJobs:TOTAL, 2)\`


### Data Scan commands

Another of RTF templates is **Data Scan** commands. These are used to create tables in which number of rows is determined by the number of records in database.

#### ➔ How to:

1. Begin by entering `\scan(dtLinkJobs)\` command. This must not necessarily be `dtLinkJobs`, depending on the template, this can be:
  - `\scan(dtLinkJobs)\` — in *invoices*, templates, to display jobs included in invoice.
  - `\scan(dtLinkInvoices)\` — in *payment* templates to display invoices linked to the payment.
  - `\scan(dtLinkPayments)\` — in *invoice* templates to indicate linked sums.
2. Insert table header between `\scan(dtLinkJobs)\` and `\scanentry\`.
3. Insert one row of data-columns after `\scanentry\`. Every variable in table must be preceded with `dtlink` command. For instance if the table begins with `\scan(dtLinkJobs)\`, each variable must have this link included: `\dtLinkJobs:JOB_NAME\`, `\dtLinkJobs:COMPLETED\`, and so on.

4. Insert `\endscan\` after data-columns.

 **Note:** Use `noeof` to hide table's header and footer if the table body appears to be empty. For instance, if your *invoice* includes only *jobs* and no *expenses*, the header and footer for *expenses* will not be saved/printed.

**Example:**

<code>\scan(dtLinkJobs)\</code>			
<b>PO No.</b>	<b>Delivered</b>	<b>Job Name</b>	<b>Service</b>
<code>\scanentry\</code>			
<code>\dtLinkJobs:PO\</code>	<code>\dtLinkJobs:COMPLETE D\</code>	<code>\dtLinkJobs:JOB_NAM E\</code>	<code>\dtLinkJobs:SERV ICE\</code>
<code>\scanfooter\</code>			
<b>Jobs Total:</b> <code>\JOBS_TOTAL\</code>			

Command `\scan(dtLinkJobs)\` will make the parser scan (go from first record to the last one) all selected data in **Jobs** table (in this particular case all *jobs* included in the *invoice*) and output them to file or printer.

Text between `\scan(dtLinkJobs)\` command and `\scanentry\` command is table header.

Text between `\scanfooter\` and `\endscan\` commands will be considered table footer and will only be displayed once at the end of this table.

Text between `\scanentry\` and `\scanfooter\` commands is table "body". It includes variables from columns which must be listed in the table. In example above these are:

- `\dtLinkJobs:PO\` – Client PO Number.
- `\dtLinkJobs:COMPLETED\` – Date of job delivery.
- `\dtLinkJobs:JOB_NAME\` – Name of the job.
- `\dtLinkJobs:SERVICE\` – Service provided.

## Condition checking

This allows the template to react to certain varying conditions and produce output suitable to each of possible conditions. The logic is the following `\IF(condition)\ Reaction \ENDIF\`.

**Example:**

If *Discount 1* is applied, display *discount name*, *discount value* and *subtotal*. The code is as follows:

```
\IF(DISCOUNT1)\
\DISCOUNT1NAME:\DISCOUNT1VALUE\
Subtotal: \AFTERDISCOUNT1\endif\
```



---

This will make the template check if discount 1 is applied, and if so — display data between `\IF(...)` and `\ENDIF` commands.


- `DISCOUNT1` is a logical variable, i.e. it can have one of two values: either true or false. In this particular case, `TO3000 3D` sets `DISCOUNT1` into true if first discount is present and into false, if there is no first discount.
- When `\IF(DISCOUNT1)` is encountered in template, parser checks `DISCOUNT1` logical value, and if it is true, runs the code below this command, until `\endif` is encountered, which instructs parser to stop. If `DISCOUNT1` is false, everything until `\endif` command is skipped.
- In this particular case, without `\IF(DISCOUNT1)` command, the parser would output empty string with a colon symbol and an empty **Subtotal**: in cases when there would not be discount. But with `\IF(DISCOUNT1 )` command, in such cases above block is skipped in its entirety.

### 3 Locale settings and date format

There are two ways of configuring locale format:

1. In *TO3000 3D* (for each *Client* individually):

- open **Client/Prospect** window and click **Main** tab;
- click **Locale Format** button in client profile;
- select **Custom locale** in the drop-down list next to **Generation of RTF files:** string;
- configure the code in **Short Date Format** or **Long Date Format** to include 4-5 capital M letters (like *dd.MMMM.YYYY*).

 **Note:** Please note that **Long Date Format** for "long" date variables: *\DATE\_DUE\_LONG\*, *\INVOICE\_DATE\_LONG\*, and so on. Settings in **Short Date Format** will affect regular variables, like *\DATE\_DUE\*, *\INVOICE\_DATE\*, and so on.

2. In *Windows Control Panel* (these settings are used as default in *TO3000 3D*):

- open *Windows Control Panel*
- double-click **Regional and Language Options**
- click **Customize** button on **Regional Options** tab
- click **Date** tab
- configure long (or short) date format options to include 4-5 capital M letters (like *dd.MMMM.YYYY*). To remove day of the week from view, delete extra "d" letters, so that no more than 2 "d-s" are present in the code string.

## 4 Common template variables

There are four categories of common variables accessible in all templates:

- User information variables;
- Current date variables;
- Client information variables;
- Client currency variable.

### User information variables

VARIABLE:	DESCRIPTION:
\USER_NAME\	Registration name taken from your License Key. It cannot be modified within program.
\USER_CURRENCY\	Company's Base Currency

### Date variables

VARIABLE:	DESCRIPTION:
\DATE\	Date in short format (for example, 10/4/2006)
\DATE_LONG\ \LONGDATE\	Date in long format (for example, Monday, October 04, 2006)


### Client information variables

This set is available in all templates where single *client* is known (all templates except [Profile](#) templates when used in **Master Profile**):

VARIABLE:	DESCRIPTION:
\CLIENT_NAME\	Name of client.
\CLIENT_CODE\	Code of client.
\CLIENT_CURRENCY\	Currency of client.
\CLIENT_MINFEE\	Minimum fee of client.
\CLIENT_PAYMENT_TERMS\	Payment terms of client.

\CLIENT_PAYMENT_TERMS_NOTES\	Additional notes on payment terms of client.
\CLIENT_ADDRESS\	Client address. This variable has pre-defined order. If address format is different in your country, you can use separate address items to include address into your customized template.
\CLIENT_STREET1\	Street address of client.
\CLIENT_STREET2\	Street address 2 (if available) of client.
\CLIENT_CITY\	City from client's profile.
\CLIENT_STATE\	State from client's profile.
\CLIENT_COUNTRY\	Country from client's profile.
\CLIENT_ZIP\	ZIP from client's profile.
\CLIENT_EMAIL1\	E-mail of client.
\CLIENT_EMAIL2\	Alternative e-mail 2 (if available) from client's profile.
\CLIENT_PHONE1\	Phone number of client from client's profile.
\CLIENT_PHONE2\	Alternative phone number 2 (if available) from client's profile.
\CLIENT_PHONE3\	Alternative phone number 3 (if available) from client's profile.
\CLIENT_PHONE4\	Alternative phone number 4 (if available) from client's profile.
\CLIENT_FAX\	Fax number from client's profile.
\CLIENT_WEB\ \CLIENT_WWW\	Web-site address from client's profile.
\CLIENT_MINFO\	Application information of client document was produced for.
\CLIENT_MWEB\ \CLIENT_MURL\	URL (Web tab address) for application submission / information.

<code>\CLIENT_INFO\</code>	General Information about the client from client's profile.
<code>\CLIENT_VATNUM\</code>	VAT Number of client from client's profile.

 **Note:** `\CLIENT_ADDRESS\` variable has pre-defined order. If address format is different in your country, you can use separate address items to include address into your customized template.

Usage of special comma-terminated variables `\CLIENT_STREET1_C\`, `\CLIENT_STREET2_C\`, `\CLIENT_CITY_C\`, `\CLIENT_STATE_C\`, `\CLIENT_COUNTRY_C\`, `\CLIENT_ZIP_C\` is necessary if you would like parts of address to be separated by commas. You can as well insert commas directly into template but in this case unnecessary commas may appear even if some variable (like second line of street address) is left empty.

## 5 Date and time functions

In most cases database stores complete date and time. Certain commands can be used to customize the format of output date and time data (you may want to output only the *day* of the *week* or only the time etc.).

The following date and time functions can be used in all templates:

- fmdt
- wd
- date
- time

**Note:** These functions return value according to [Regional and Language Options](#) settings in your system. These options can be changed with the help of your [Windows Control Panel](#).

The same variable will be displayed in different way depending on the function used.

FUNCTION:	VARIABLE VALUE:	FUNCTION APPLIED:
fmdt	9/20/06 6:00 PM	Wednesday, September 20, 2006 6:00 PM
	9/20/06	Wednesday, September 20, 2006
wd	9/20/06 6:00 PM	Wednesday
	9/20/06	Wednesday
date	9/20/06 6:00 PM	09/20/06
	9/20/06	09/20/06
time	9/20/06 6:00 PM	6:00 PM
	9/20/06	(empty row)

Date and time functions are accessible in all templates and can be applied to:

- All variables from datasets which return date and time.
- All variables from the following table:

VARIABLE:	TYPE:	DESCRIPTION:
\PROJECT_DATE_STARTED\	Project template variable	Date when the <i>project</i> was started in following format: 9/20/2006.
\PROJECT_DATE_DEADLINE\	Project template variable	<i>Project</i> deadline in following format: 9/20/2006.
\PROJECT_DATE_COMPLETED\	Project template variable	Date of the <i>project</i> completion in following format: 9/20/2006.
\ASSIGNED\	Client Jobs template variable	Date when the <i>job</i> was assigned in following format: 9/20/2006.
\DEADLINE\	Client Jobs template variable	<i>Job</i> deadline in following format: 9/20/2006
\COMPLETED\	Client Jobs template variable	Completion date in following format: 9/20/2006.
\DONE\	Client Jobs template variable	Completion date in following format: 9/20/2006.
\START\ \ESTSTART\	Quotes template variable	Date assigned in following format: 9/20/2006.
\COMPLETION\ \ESTCOMPLETION\	Quotes template variable	Deadline date in following format: 9/20/2006.

### Syntax

Date and time functions are added to the variable in the following way:

\function(VARIABLE)\

#### ➔ Example

To add wd function to \ASSIGNED\ variable from the *client jobs* template, change the variable syntax in the following way: \wd(ASSIGNED)\

The result will be the day of the week, when the *job* was assigned (e.g. Wednesday).

## 6 Project template variables (With Projects Mode only)

These variables can be used in Project templates, as well as in any job or job assignment templates.

<b>VARIABLE:</b>	<b>DESCRIPTION:</b>
\PROJECT_NAME\	Name of <i>project</i> .
\PROJECT_CODE\	<i>Project</i> code.
\PROJECT_CLIENT_NAME\	<i>Client project</i> created for.
\PROJECT_CLIENT_CODE\	<i>Client</i> reference number.
\PROJECT_INFO\	Information about the <i>project</i> .
\CLIENT_PM_NAME\	Client <i>project manager</i>
\PROJECT_DATE_STARTED\	Date when <i>project</i> was started in following format: 10/4/2006.
\PROJECT_DATE_DEADLINE\	<i>Project</i> deadline in following format: 10/4/2006.
\PROJECT_DATE_COMPLETED\	Date of <i>project</i> completion in following format: 10/4/2006.
\PROJECT_DATE_STARTED_LONG\	Date when <i>project</i> was started in following format: Monday, October 04, 2006.
\PROJECT_DATE_DEADLINE_LONG\	<i>Project</i> deadline in following format: Monday, October 04, 2006.
\PROJECT_DATE_COMPLETED_LONG\	Date of <i>project</i> completion in following format: Monday, October 04, 2006.



## 7 Contacts template variables

These variables are used in any template mentioning client's details.

VARIABLE	DESCRIPTION
\SALUTATION\ \SAL\ \CONTACT_SALUTATION\ 	For example: "Mr.", "Ms.", "Mrs." etc.
\CONTACT_TITLE\ \TITLE\ 	<i>Contact</i> title.
\CONTACT_NAME\ \PM_NAME\ \ATTENTION\ 	<i>Contact</i> name.
\CONTACT_EMAIL1\ 	<i>Contact</i> email address.
\CONTACT_EMAIL2\ 	<i>Contact</i> email address 2 (if available).
\CONTACT_PHONE1\ 	<i>Contact</i> phone number.
\CONTACT_PHONE2\ 	<i>Contact</i> phone number 2 (if available).
\CONTACT_FAX\ 	<i>Contact</i> fax number.
\CONTACT_NOTES\ 	<i>Contact</i> notes.

## 8 Quotes template variables

Using variables of quote templates you can construct templates either for your reference or for sending to client by email or fax.

### Common Quote Variables

VARIABLE	DESCRIPTION
\QUOTE_NAME\ \	Possible quote name.
\QUOTE_CODE\ \CODE\ \	Quote code.
\SERVICE\ \	Service name.
\REQUEST\ \	Request for quote.
\ANSWER\ \	Answer to request for quote.
\VOLUME\ \	Quote volume.
\PRICING\ \TYPE\ \	Quote type.
\PRICE\ \	Quote price.
\UNITS\ \	Quote units.
\TOTAL\ \	Quote total.
\DATE_SENT\ \SENT\ \	Date sent in following format: 10/4/2006.
\DATE_SENT_LONG\ \LONGSENT\ \	Date sent in following format: Monday, October 04, 2006.
\START\ \ESTSTART\ \	Date assigned in following format: 10/4/2006.
\START_LONG\ \LONGESTSTART\ \	Date assigned in following format: Monday, October 04, 2006.

\COMPLETION\ \ESTCOMPLETION\	Deadline date in following format: 10/4/2006.
\LONG_COMPLETION\ \LONGESTCOMPLETION\	Deadline date in following format: Monday, October 04, 2006.
\COUNT_NOTES\	CATCount notes.
\STATUS\	Status of quote (unknown, accepted, r ejected)

### Taxes

VARIABLE	DESCRIPTION
\TAX1\	Indicates if the tax 1 has been added; used in algorithms (If tax 1 exists = True, if it doesn't = False)
\TAX1_NAME\	Tax 1 name
\TAX1_PERCENTS\	Tax 1 value in percents
\TAX1_VALUE\	Tax 1 value in client's currency
\TAX1_VALUE_BASE\	Tax 1 value in base currency
\TAX2\	Used in algorithms (If tax 2 exists = True, if it doesn't = False)
\TAX2_NAME\	Tax 2 name
\TAX2_PERCENTS\	Tax 2 value in percents
\TAX2_VALUE\	Tax 2 value in client's currency
\TAX2_VALUE_BASE\	Tax 2 value in base currency
\TAXES\	Indicates whether the taxes in Tax fields are set; used in algorithms (If any tax is set = True, if they are not = False)
\AFTER_TAX1\	Total in client's currency after Tax 1 applied
\AFTER_TAX1_BASE\	Total in base currency after Tax 1 applied

## Discounts

VARIABLE	DESCRIPTION
\DISCOUNT1\	Indicates if the discount/markup in the Discount/Markup field is set; used in algorithms (If discount 1 exists = True, if it does not = False)
\DISCOUNT1_NAME\	Discount 1 name
\DISCOUNT1_PERCENTS\	Discount 1 value in percents
\DISCOUNT1_VALUE\	Discount 1 value in client's currency
\DISCOUNT1_VALUE_BASE\	Discount 1 value in base currency
\DISCOUNT2\	Indicates if the discount/markup in the <b>Discount/Markup</b> field is set; used in algorithms (If discount 2 exists = True, if it does not = False)
\DISCOUNT2_NAME\	Discount 2 name
\DISCOUNT2_PERCENTS\	Discount 2 value in percents
\DISCOUNT2_VALUE\	Discount 2 value in client's currency
\DISCOUNT2_VALUE_BASE\	Discount 2 value in base currency
\DISCOUNTS\	Indicates if the discounts/markups in the <b>Discount/Markup</b> fields are set; used in algorithms (If any discount is set = True, if there are no = False)
\AFTER_DISCOUNT1\	Total in client's currency after Discount 1 applied
\AFTER_DISCOUNT1_BASE\	Total in base currency after Discount 1 applied

## Draft Client Job variables for Quotes to Clients

The following variables refer to draft client jobs, added to quote to client.

VARIABLE	DESCRIPTION
\DTMULTIQUOTEITEMS:QI_NAME\	Name of draft client job.
\DTMULTIQUOTEITEMS:GROUP_NAME\	Group of services of draft client job.
\DTMULTIQUOTEITEMS:SERV_NAME\	Service name of draft client job.
\DTMULTIQUOTEITEMS:UNIT_NAME\	Units in which draft client job volume is specified.
\DTMULTIQUOTEITEMS:QI_PRICE\	Price of draft client job.
\DTMULTIQUOTEITEMS:QI_VOLUME\	Volume of draft client job.
\DTMULTIQUOTEITEMS:QI_FEE_KIND\	Type of price set for client job (can be <i>per unit</i> , <i>flat fee</i> , or <i>free</i> ).
\DTMULTIQUOTEITEMS:QI_TOTAL\	Draft client job total.
\DTMULTIQUOTEITEMS:QI_COUNTNOTES\	CATCount notes of draft client job (if job volume had been entered with the help of CATCount).

## 9 Client Jobs template variables

Variables for templates from **New/Edit Job** dialog box can be used to construct document templates either for your reference during work process or for confirming job details to client.

VARIABLE	DESCRIPTION
\JOB_NAME\	<i>Job</i> name.
\JOB_CODE\ \CODE\	<i>Job</i> code.
\PO_CODE\ \PO\	Purchase order <i>client</i> issued for this <i>job</i> .
\CLIENT_REF\	<i>Client</i> reference number in accounting system of <i>client</i> .
\SERVICE\	<i>Service</i> name.
\GROUP_NAME\	<i>Group of Services</i> name.
\INSTRUCTIONS\	<i>Job</i> instructions.
\WORK_NOTES\	Work notes.
\VOLUME\	<i>Job</i> volume.
\TYPE\	<i>Job</i> type (for example: per unit, flat fee, free)
\PRICE\	<i>Job</i> price.
\UNITS\	<i>Job</i> units.
\TOTAL\	<i>Job</i> total.
\ASSIGNED\	Date when <i>job</i> was assigned in following format: 10/4/2006.

\ASSIGNED_LONG\ \LONGASSIGNED\ 	Date when <i>job</i> was assigned in following format: Monday, October 04, 2006.
\DEADLINE\ 	<i>Job</i> deadline in following format: 10/4/2006.
\DEADLINE_LONG\ 	<i>Job</i> deadline in following format:
\LONGDEADLINE\ 	Monday, October 04, 2006.
\COMPLETED\ \DONE\ 	Completion date in following format: 10/4/2006.
\COMPLETED_LONG\ \LONGCOMPLETED\ 	Completion date in following format: Monday, October 04, 2006.
\COUNT_NOTES\ 	CATCount or AnyCount notes.
\INVOICE_CODE\ \INVOICE\ 	<i>Invoice</i> code.
\INVOICE_GLOBAL_CODE\ \INV_GLOBAL\ 	<i>Invoice</i> global code.

## 10 Invoice template variables

### Date and code

VARIABLE	DESCRIPTION
\STATUS\ \DATE_DUE\ \SETTLEMENT_DATE\ \DATE_DUE_LONG\ \SETTLEMENT_LONGDATE\ \DUE_DATELONG\ \INVOICE_DATE\ \INV_DATE\ \INVOICE_DATE_LONG\ \INV_LONGDATE\ \INVOICE_CODE\ \INV_CODE\ \INVOICE_GLOBAL_CODE\ \INV_GLOBAL\ \INV_GLOBALLONG	Invoice status (For example: "Expected within 30 days", "Settled 5 days earlier" and so on.) Date when invoice is due in following format: 10/4/2006 Date when invoice is due in following format: Monday, October 04, 2006 Date invoice sent in following format: 10/4/2006 Date invoice sent in following format: Monday, October 04, 2006 Invoice code Invoice global code

### Invoice totals

VARIABLE	DESCRIPTION
\INVOICE_TOTAL\ \INVOICE_TOTAL_BASE\ \JOBS_TOTAL\ \JOBS_TOTAL_BASE	Invoice total <i>in client's currency</i> Invoice total <i>in base currency</i> Jobs total <i>in client's currency</i> Jobs total <i>in base currency</i>



\NET_JOBS_TOTAL\	Jobs total with discounts <i>in client's currency</i>
\NET_JOBS_TOTAL_BASE\	Jobs total with discounts <i>in base currency</i>
\VOLUME_BASE\	Jobs total volume in base units

## Taxes

VARIABLE	DESCRIPTION
\TAX1\	Indicates if the tax 1 has been added; used in algorithms (If tax 1 exists = True, if it doesn't = False)
\TAX1_NAME\	Tax 1 name
\TAX1_PERCENTS\	Tax 1 value in percents
\TAX1_VALUE\	Tax 1 value <i>in client's currency</i>
\TAX1_VALUE_BASE\	Tax 1 value <i>in base currency</i>
\TAX2\	Used in algorithms (If tax 2 exists = True, if it doesn't = False)
\TAX2_NAME\	Tax 2 name
\TAX2_PERCENTS\	Tax 2 value in percents
\TAX2_VALUE\	Tax 2 value <i>in client's currency</i>
\TAX2_VALUE_BASE\	Tax 2 value <i>in base currency</i>
\TAXES\	Indicates if the taxes in the Tax fields are set; used in algorithms (If any tax is set = True, if it they are not = False)
\AFTER_TAX1\	Total <i>in client's currency</i> after Tax 1 applied
\AFTER_TAX1_BASE\	Total <i>in base currency</i> after Tax 1 applied

## Discounts

VARIABLE	DESCRIPTION
\DISCOUNT1\	Indicates if the discount/markup in the Discount/Markup field is set; used in algorithms (If discount 1 exists = True, if it does not = False)
\DISCOUNT1_NAME\	Discount 1 name
\DISCOUNT1_PERCENTS\	Discount 1 value in percents
\DISCOUNT1_VALUE\	Discount 1 value <i>in client's currency</i>
\DISCOUNT1_VALUE_BASE\	Discount 1 value <i>in base currency</i>
\DISCOUNT2\	Indicates if the discount/markup in the Discount/Markup field is set; used in algorithms (If discount 2 exists = True, if it does not = False)
\DISCOUNT2_NAME\	Discount 2 name
\DISCOUNT2_PERCENTS\	Discount 2 value in percents
\DISCOUNT2_VALUE\	Discount 2 value <i>in client's currency</i>
\DISCOUNT2_VALUE_BASE\	Discount 2 value <i>in base currency</i>
\DISCOUNTS\	Indicates if the discounts/markups in the Discount/Markup fields are set; used in algorithms (If any discount is set = True, if there are no = False)
\AFTER_DISCOUNT1\	Total <i>in client's currency</i> after Discount 1 applied
\AFTER_DISCOUNT1_BASE\	Total <i>in base currency</i> after Discount 1 applied
\NET_JOBS_TOTAL\	Jobs total with discounts <i>in client's currency</i>
\NET_JOBS_TOTAL_BASE\	Jobs total with discounts <i>in base currency</i>

**Payment status variables**

VARIABLE	DESCRIPTION
\INVOICE_DUE\	Balance due <i>in client's currency</i>
\INVOICE_DUE_BASE\	Balance due <i>in base currency</i>
\INVOICE_PAID\	Total payments linked to this invoice <i>in client's currency</i>
\INVOICE_PAID_BASE\	Total payments linked to this invoice <i>in base currency</i>
\INV_IS_PAID\	Used in algorithms (If the invoice is paid = True, if it's not = False)
\BEFORE_ADJUSTMENTS\	Invoice total <i>in client's currency</i> , excluding adjustments
\BEFORE_ADJUSTMENTS_BASE\	Invoice total <i>in base currency</i> , excluding adjustments
\ADJUSTMENTS_VALUE\	Value of the adjustments <i>in client's currency</i>
\ADJUSTMENTS_VALUE_BASE\	Value of the adjustments <i>in base currency</i>
\ADJUSTMENTS_DESCR\	Description of the adjustments
\INVOICE_PAYMETHOD\	Invoice payment method. See Payment Methods topic
\INVOICE_PAYMETHOD_DESCR\	Payment method description. See Payment Methods topic

**Credit note status variables**

VARIABLE	DESCRIPTION
\CN_TOTAL\	The total of credit notes linked to this invoice, in the clients currency
\CN_TOTAL_BASE\	The total of credit notes linked to this invoice, in the base currency

## Jobs variables in invoice

The following variables refer to jobs added to invoice

VARIABLE	DESCRIPTION
\DTLINKJOBS:CJOB_PONUMB\	Client PO of the job
\DTLINKJOBS:CJOB_NAME\	Client job name
\DTLINKJOBS:CJOB_SERV_NAME\	<i>Service</i> name.
\DTLINKJOBS:CJOB_GROUP_NAME\	<i>Group of Services</i> name.
\DTLINKJOBS:CJOB_ASSIGNED\	Date the client job was assigned
\DTLINKJOBS:CJOB_DEADLINE\	Deadline of client job
\DTLINKJOBS:CJOB_ISCOMPLETED\	Completed (Boolean: True/False)
\DTLINKJOBS:CJOB_COMPLETED\	Date the client job was completed
\DTLINKJOBS:CJOB_PRICE\	Price of client job
\DTLINKJOBS:CJOB_VOLUME\	Client job volume
\DTLINKJOBS:CJOB_FEE_KIND\	Pricing (per unit, flat fee)
\DTLINKJOBS:CJOB_RATE\	Exchange rate
\DTLINKJOBS:CJOB_TOTAL\	Job total <i>in client's currency</i>
\DTLINKJOBS:CJOB_TOTAL_BASE\	Job total <i>in base currency</i>
\DTLINKJOBS:CJOB_INSTRUCTION\	Instructions of client job
\DTLINKJOBS:CJOB_WORKNOTES\	Work notes of client job
\DTLINKJOBS:CJOB_COUNTNOTES\	CATCount notes of client job
\DTLINKJOBS:SERV_NAME\	Service name of client job
\DTLINKJOBS:UNIT_NAME\	Volume units of client job
\DTLINKJOBS:PROJ_CODE\	Project code of client job

\DTLINKJOBS:CJOB_CODE\	Client job code
\DTLINKJOBS:CJOB_CLCODE\	Client Ref. of the job
\DTLINKJOBS:CCON_NAME\	Client PM of the client job

### Linked payment variables

VARIABLE	DESCRIPTION
\DTLINKPAYMENTS:CPAYM_CODE\	Linked payment's code
\DTLINKPAYMENTS:CPAYM_DATE\	Linked payment's creation date
\DTLINKPAYMENTS:CPAYM_TOTOAL\	Linked payment's total value
\DTLINKPAYMENTS:LINK_SUM\	Part of the payment's value linked to this invoice

**Linked credit note variables**

<b>VARIABLE</b>	<b>DESCRIPTION</b>
\DTLINKCREDITNOTES:CN_CODE\	Credit Note's code
\DTLINKCREDITNOTES:CN_DATE\	Credit Note's creation date
\DTLINKCREDITNOTES:CN_TOTAL\	Credit Note's total value
\DTLINKCREDITNOTES:CN_ASSIGN_TOTAL\	Credit note's value covered by refunds
\DTLINKCREDITNOTES:BALANCE\	Credit note's value not covered by refunds
\DTLINKCREDITNOTES:CN_STATUS\	Credit Note's status. Has the following values: 0 - "Closed": the CN's total is subtracted from the invoice's balance. 1 - "Awaiting payment": the Invoice is paid and the CN requires a refund. 2 - "Partially paid": the CN is partially covered by a refund. 3 - "Fully paid": the CN is fully covered by a refund.

## 11 Credit note template variables

### Date and Code

VARIABLE	DESCRIPTION
\CN_STATUS\	Credit note status <b>Example:</b> Awaiting payment, Fully paid
\CN_DATE\	Date credit note was sent, in the following format: 10/4/2006
\CN_DATE_LONG\	Date credit note was sent, in the following format: Monday, October 04, 2006
\CN_CODE\	Credit note code

### Linked invoice variables

VARIABLE	DESCRIPTION
\INV_DATE\	Date linked invoice was sent, in the following format: 10/4/2006
\INV_CODE\	Linked invoice code
\INV_TOTAL\	Linked invoice total in client's currency

### Credit note totals

VARIABLE	DESCRIPTION
\CN_TOTAL\	Credit note total in client's currency
\CN_SUB_TOTAL\	Credit note total with taxes
\CN_RATE\	Exchange rate into base currency
\CN_TOTAL_BASE\	Credit note total in base currency

### Taxes

VARIABLE	DESCRIPTION
\TAX1\	Indicates if the tax 1 has been added; used in algorithms (If tax 1 exists = True, if it doesn't = False)
\TAX1_NAME\	Tax 1 name
\TAX1_PERCENTS\	Tax 1 value in percents
\TAX1_VALUE\	Tax 1 value in client's currency
\TAX1_VALUE_BASE\	Tax 1 value in base currency
\TAX2\	Used in algorithms (If tax 2 exists = True, if it doesn't = False)



\TAX2_NAME\	Tax 2 name
\TAX2_PERCENTS\	Tax 2 value in percents
\TAX2_VALUE\	Tax 2 value in client's currency
\TAX2_VALUE_BASE\	Tax 2 value in base currency
\TAXES\	Indicates whether the taxes in Tax fields are set; used in algorithms (If any tax is set = True, if it they are not = False)

### Refund status variables

VARIABLE	DESCRIPTION
\CN_DUE\	Balance due in client's currency
\CN_DUE_BASE\	Balance due in base currency
\CN_PAID\	Total refunds linked to this credit note in client's currency
\CN_PAID_BASE\	Total refunds linked to this credit note in base currency
\CN_IS_PAID\	Used in algorithms (If the credit note is paid = True, if it's not = False)

**Linked refund variables**

<b>VARIABLE</b>	<b>DESCRIPTION</b>
\DTLINKREFUNDS:CREF_CODE\	Linked refund's Code
\DTLINKREFUNDS:CREF_DATE\	Linked refund's creation date
\DTLINKREFUNDS:CREF_TOTAL\	Linked refund's total value
\DTLINKREFUNDS:LINK_SUM\	Part of the linked refund's value linked to this credit note

## 12 Payments template variables

### Basic payments template variables

The following variables can be used to create payment templates

VARIABLE	DESCRIPTION
\PAYMENT_CODE\ \CODE\ 	<i>Payment</i> code.
\PAYMENT_DATE\ \PDATE\ 	Date <i>payment</i> received in following format: 10/4/2006.
\PAYMENT_DATE_LONG\ \PLONGDATE\ 	Date <i>payment</i> received in following format: Monday, October 04, 2006.
\PAYMENT_NOTES\ 	Notes about <i>payment</i> .
\TOTAL_PAID\ \TOTAL\ 	Total paid.
\NOT_LINKED\ 	Amount not linked with <i>invoices</i> .
\LINKED\ 	Amount linked with <i>invoices</i> .
\IS_LINKED\ 	Used in algorithms (If <i>payment</i> is linked with <i>invoice</i> = True, if it's not = False)
\PAYMENT_NOTES\ \NOTES\ 	<i>Payment notes</i>

### Linked invoice variables

These variables can add information from linked invoices to payment template

VARIABLE	DESCRIPTION
\DTLINKINVOICES:IDATE\	Linked <i>invoice</i> date.
\DTLINKINVOICES:ICODE\	Linked <i>invoice code</i> .
\DTLINKINVOICES:GNUMB\	Linked <i>invoice global code</i> .
\DTLINKINVOICES:TOTAL\	Linked <i>invoice</i> total.
\DTLINKINVOICES:OTHER\	The part of the linked <i>invoice</i> total covered by other <i>payments</i> .
\DTLINKINVOICES:ADJUST\	The sum of the <i>phantom payment</i> of the linked <i>invoice</i> .
\DTLINKINVOICES:BALANCE\	Balance Due of the linked <i>invoice</i> .
\DTLINKINVOICES:DATEDUE\	Linked <i>invoice</i> due date.

## 13 Refund Template Variables

VARIABLE	DESCRIPTION
\REF_CODE\	Refund code
\REF_DATE\	Date refund was received, in the following format: 10/4/2006
\REF_DATE_LONG\	Date refund was received, in the following format: Monday, October 04, 2006
\REF_NOTES\	Notes about refund.
\TOTAL_PAID\	Total paid (in client's currency).
\TOTAL_PAID_BASE\	Total paid (in base currency).
\REF_RATE\	Exchange rate.
\NOT_LINKED\	Amount not linked with credit notes.
\LINKED\	Amount linked with credit notes.
\IS_LINKED\	Used in algorithms (If refund is linked with credit note = True, if it's not = False).

### Linked credit note variables

VARIABLE	DESCRIPTION
\dtLinkCNs:ICODE\	The credit note's code
\dtLinkCNs:IDATE\	The credit note's creation date
\dtLinkCNs:TOTAL\	The credit note's total value
\dtLinkCNs:LINKED\	The part of the credit note's total that is linked to this refund
\dtLinkCNs:OTHER\	The part of the credit note's total that is linked to other refunds
\dtLinkCNs:BALANCE\	The part of the credit note's total that is not linked to any refund