Translation Office 3000, Version 10
RTF Templates Guide

Table of Contents

1 Templates

Templates in TO3000, Version 10

Template is an *RTF* ("Rich Text Format") file stored in your TO3000, Version 10 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:

- Invoices to clients
- Payments from clients summaries
- Project, and client job summaries
- Quotes to clients

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

Any window which has RTF Templates section can be used to save its data in a printable document. Until the data which you are editing is saved in your database, RTF Templates section will display message that the templates will become available once the new document record has been saved in database.

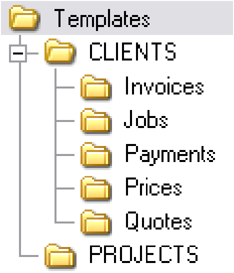
To save a document, first apply changes made to this document by clicking **Apply** button. Once the document has been saved in database, RTF template controls for it will become active, and display a drop-down list, which can be used to select different templates.

- 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.

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 overlay and format, you edit overlay 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, Version 10 **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 certain symbol combination which is entered in template files. When a document is saved using template, TO3000, Version 10 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 in the 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.

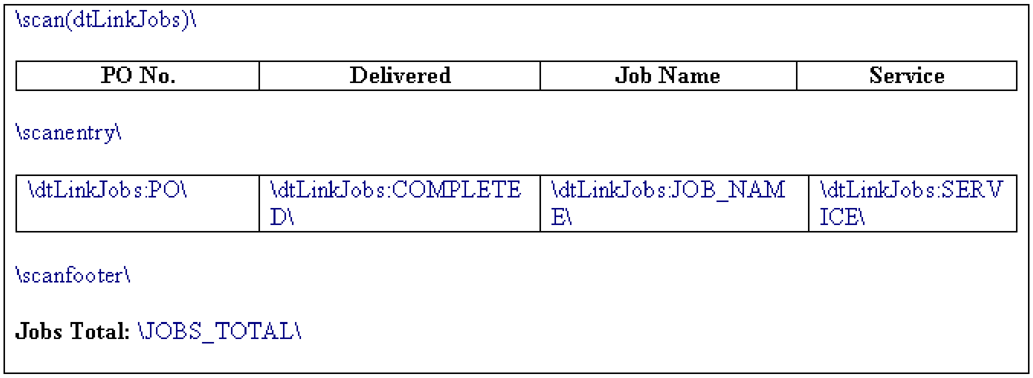
2. Insert table header between \scan(dtLinkJobs)\ and \scanentry\.

3. Insert one row of data-columns after \scanentry\. Every variable in table must be preceeded with dtlink commfnd. 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:



\scan(dtLinkJobs)\

PO No.	Delivered	Job Name	Service

\scanentry\

\dtLinkJobs:PO\	\dtLinkJobs:COMPLETED\	\dtLinkJobs:JOB_NAME\	\dtLinkJobs:SERVICE\

\scanfooter\

Jobs Total: \JOBS_TOTAL\

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, Version 10 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: and 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 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\ or \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\ or \CLIENT_WWW\	Web-site address from client's profile.
\CLIENT_MINFO\	Application information of client document was produced for.
\CLIENT_MWEB\ or \CLIENT_MURL\	URL (Web tab address) for application submission / information.
\CLIENT_INFO\	General Information about the client from client's profile.
\CLIENT_VATNUM\	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.

4 Locale settings and date format

There are two ways of configuring locale format:

1. In TO3000, Version 10 (for each *Client* individually):

- open **Client** 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, Version 10):

- 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.

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 wit 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 *project* was started in following format: 9/20/2006.
\PROJECT_DATE_DEADLINE\	Project template variable	*Project* deadline in following format: 9/20/2006.
\PROJECT_DATE_COMPLETED\	Project template variable	Date of the *project* completion in following format: 9/20/2006.
\ASSIGNED\	Client Jobs template variable	Date when the *job* was assigned in following format: 9/20/2006.
\DEADLINE\	Client Jobs template variable	*Job* 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.

VARIABLE:	DESCRIPTION:
\PROJECT_NAME\	Name of *project*.
\PROJECT_CODE\	*Project* code.
\PROJECT_CLIENT_NAME\	*Client project* created for.
\PROJECT_CLIENT_CODE\	*Client* reference number.
\PROJECT_INFO\	Information about the *project*.
\CLIENT_PM_NAME\	Client *project manager*
\PROJECT_DATE_STARTED\	Date when *project* was started in following format: 10/4/2006.
\PROJECT_DATE_DEADLINE\	*Project* deadline in following format: 10/4/2006.
\PROJECT_DATE_COMPLETED\	Date of *project* completion in following format: 10/4/2006.
\PROJECT_DATE_STARTED_LONG\	Date when *project* was started in following format: Monday, October 04, 2006.
\PROJECT_DATE_DEADLINE_LONG\	*Project* deadline in following format: Monday, October 04, 2006.
\PROJECT_DATE_COMPLETED_LONG\	Date of *project* 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\	*Contact* title.
\CONTACT_NAME\ \PM_NAME\ \ATTENTION\	*Contact* name.
\CONTACT_EMAIL1\	*Contact* email address.
\CONTACT_EMAIL2\	*Contact* email address 2 (if available).
\CONTACT_PHONE1\	*Contact* phone number.
\CONTACT_PHONE2\	*Contact* phone number 2 (if available).
\CONTACT_FAX\	*Contact* fax number.
\CONTACT_NOTES\	*Contact* 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)

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 *per unit*, *flat fee*, or *free*).
\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\	*Job* name.
\JOB_CODE\ \CODE\	*Job* code.
\PO_CODE\ \PO\	Purchase order *client* issued for this *job*.
\CLIENT_REF\	*Client* reference number in accounting system of *client*.
\SERVICE\	*Service* name.
\GROUP_NAME\	*Group of Services* name.
\INSTRUCTIONS\	*Job* instructions.
\WORK_NOTES\	Work notes.
\VOLUME\	*Job* volume.
\TYPE\	*Job* type (for example: per unit, flat fee, free)
\PRICE\	*Job* price.
\UNITS\	*Job* units.
\TOTAL\	*Job* total.
\ASSIGNED\	Date when *job* was assigned in following format: 10/4/2006.
\ASSIGNED_LONG\ \LONGASSIGNED\	Date when *job* was assigned in following format: Monday, October 04, 2006.
\DEADLINE\	*Job* deadline in following format: 10/4/2006.
\DEADLINE_LONG\	*Job* 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\	*Invoice* code.
\INVOICE_GLOBAL_CODE\ \INV_GLOBAL\	*Invoice* global code.

10 Invoice template variables

Date and code

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

Invoice totals

VARIABLE	DESCRIPTION
\INVOICE_TOTAL\	Invoice total *in client's currency*
\INVOICE_TOTAL_BASE\	Invoice total *in base currency*
\JOBS_TOTAL\	Jobs total *in client's currency*
\JOBS_TOTAL_B\	Jobs total *in base currency*
\NET_JOBS_TOTAL\	Jobs total with discounts *in client's currency*
\NET_JOBS_TOTAL_B\	Jobs total with discounts *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_VALUE\	Tax 1 value *in client's currency*
\TAX1_VALUE_B\	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_VALUE\	Tax 2 value *in client's currency*
\TAX2_VALUE_B\	Tax 2 value *in base currency*
\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 *in client's currency* after Tax 1 applied
\AFTER_TAX1_B\	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_VALUE\	Discount 1 value *in client's currency*
\DISCOUNT1_VALUE_B\	Discount 1 value *in base currency*
\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_VALUE\	Discount 2 value *in client's currency*
\DISCOUNT2_VALUE_B\	Discount 2 value *in base currency*
\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 *in client's currency* after Discount 1 applied
\AFTER_DISCOUNT1_B\	Total *in base currency* after Discount 1 applied
\NET_JOBS_TOTAL\	Jobs total with discounts *in client's currency*
\NET_JOBS_TOTAL_B\	Jobs total with discounts *in base currency*

Payment status variables

VARIABLE	DESCRIPTION
\INVOICE_DUE\	Balance due *in client's currency*
\INVOICE_DUE_BASE\	Balance due *in base currency*
\INVOICE_PAID\	Total payments linked to this invoice *in client's currency*
\INVOICE_PAID_BASE\	Total payments linked to this invoice *in base currency*
\INV_IS_PAID\	Used in algorithms (If the invoice is paid = True, if it's not = False)
\BEFORE_ADJUSTMENTS\	Invoice total *in client's currency*, excluding adjustments
\BEFORE_ADJUSTMENTS_B\	Invoice total *in base currency*, excluding adjustments
\ADJUSTMENTS_VALUE\	Value of the adjustments *in client's currency*
\ADJUSTMENTS_VALUE_B\	Value of the adjustments *in base currency*
\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

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\	*Service* name.
\DTLINKJOBS:CJOB_GROUP_NAME\	*Group of Services* 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 *in client's currency*
\DTLINKJOBS:CJOB_TOTAL_B\	Job total *in base currency*
\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

11 Payments template variables

Basic payments template variables

The following variables can be used to create payment templates

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

Linked invoice variables

These variables can add information from linked invoices to payment template

DATASET WITH COLUMN NAME	DESCRIPTION
\DTLINKINVOICES:IDATE\	Linked *invoice* date.
\DTLINKINVOICES:ICODE\	Linked *invoice code*.
\DTLINKINVOICES:GNUMB\	Linked *invoice global code*.
\DTLINKINVOICES:TOTAL\	Linked *invoice* total.
\DTLINKINVOICES:OTHER\	The part of the linked *invoice* total covered by other *payments*.
\DTLINKINVOICES:ADJUST\	The sum of the *phantom payment* of the linked *invoice*.

\DTLINKINVOICES:LINKED\	The part of the current *payment* total linked to the *invoice*.
\DTLINKINVOICES:BALANCE\	Balance Due of the linked *invoice*.
\DTLINKINVOICES:DATEDUE\	Linked *invoice* due date.