
tryton
Release 6.8

unknown

Jul 29, 2023

CONTENTS

| | | |
|----------|-----------------------------|-----------|
| 1 | Contents | 3 |
| 1.1 | Installing tryton | 3 |
| 1.2 | Client Usage | 3 |
| 1.3 | Glossary | 12 |
| 1.4 | Release notes | 14 |
| | Bibliography | 31 |
| | Index | 33 |

Tryton is a Graphical User Interface to Tryton based on [GTK](#) and [Python](#).

CONTENTS

1.1 Installing tryton

1.1.1 Installation

There are three options to install Tryton:

- Install the version provided by your operating system distribution. This is the quickest and recommended option for those who has operating system that distributes Tryton.
- Install the published package. You first need to have `pip` installed. Then to install `tryton` run:

```
$ python -m pip install tryton
```

- Without installation, you need to make sure you have all the dependencies installed and then run:

```
$ python bin/tryton
```

1.2 Client Usage

This document is the reference about the concepts of the graphical user interface (also known as *Tryton client*) of the Tryton application framework.

1.2.1 Name

tryton - Graphical user client of the Tryton application framework

1.2.2 Synopsis

```
$ tryton [options] [url]
```

On startup the login dialog is displayed.

1.2.3 Options

- version** Show program version number and exit
- h, --help** Show help message and exit
- c FILE, --config=FILE** Specify alternate *configuration file*
- d, --dev** Enable development mode, which deactivates client side caching
- v, --verbose** Enable basic debugging
- l LOG_LEVEL, --log-level=LOG_LEVEL** Specify the log level: DEBUG, INFO, WARNING, ERROR, CRITICAL
- o LOG_OUTPUT, --log-ouput=LOG_OUTPUT** Specify the file used to output logging information
- u LOGIN, --user=LOGIN** Specify the login user
- s SERVER, --server=SERVER** Specify the server hostname:port

1.2.4 URL

When an url is passed, the client will try to find already running client that could handle it and send to this one to open the url. If it doesn't find one then it will start the GUI and open the url itself.

The url schemes are:

```
tryton://<hostname>[:<port>]/<database>/model/<model name>[/<id>][;parameters]
tryton://<hostname>[:<port>]/<database>/wizard/<wizard name>[;parameters]
tryton://<hostname>[:<port>]/<database>/report/<report name>[;parameters]
```

parameters

the corresponding fields of actions encoded in [JSON](#)

Note: model is for window

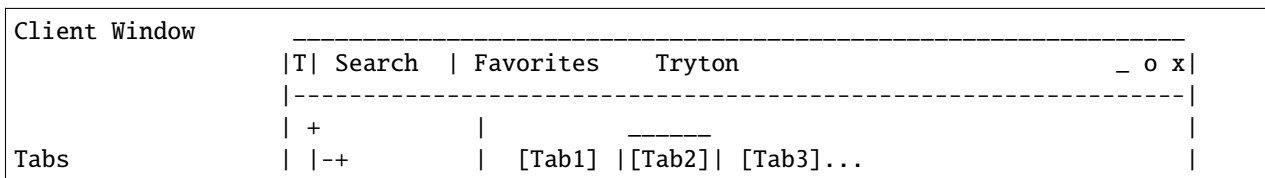
Note: report must have at least a data parameter with ids, id and model name

1.2.5 Overview

On startup the login dialog is displayed. It allows to select a existing profile (or to manage them) or to enter the host and database information.

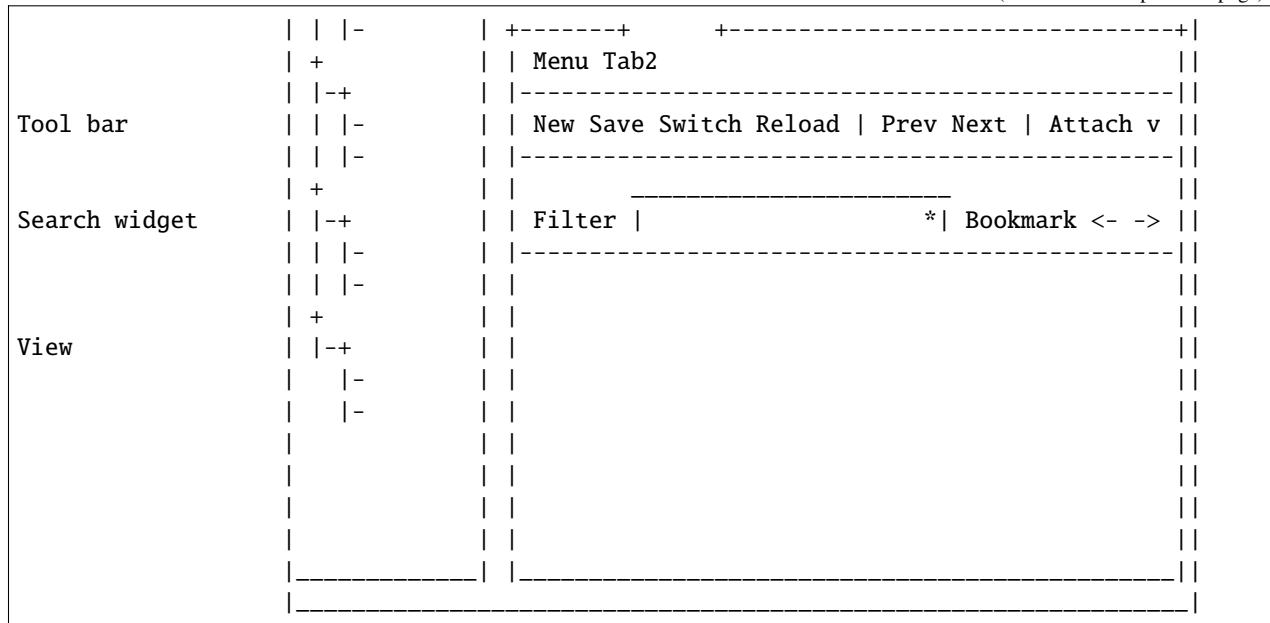
The following schematic illustration of the Tryton client shows the names of all important visual parts.

Figure: Tryton client application:



(continues on next page)

(continued from previous page)



Tabbed Main Frame

This part of the client contains all the related contents and functions provided by the *Tryton server modules*. All aspects inside the *main frame* depend at least on the individual set of activated modules.

The main frame provides a *tabbed document interface* to arrange different views side by side. New *tabs* are opened by special *actions*, like choosing a menu item or clicking some action buttons. All tabs include titles which show the name of the provided view.

Tabs can be arranged by Drag and Drop.

Note: Inside *views* there can be tabs, too.

Menu

The *menu* does not contain fixed menu items. All of them are dynamically provided by the actual set of the activated *modules* depending on the access rules of the current user. If a menu item is clicked, the appropriate action will open in a new tab.

A search field allows to quickly filter the menu items by name and to search in models for which the global search is enabled.

1.2.6 Application Menu

The following section describes the action of the application menu. A rule of thumb: All items of the menu bar that are suffixed by three dots (...) will open an intermediate *dialog* for setting up the provided menu action. Most dialog provide a *Cancel* button, used to stop the complete dialog process.

Preferences:

A preference dialog opens, where the actual user can show and edit his personal settings. All user preferences are stored server side. I.e. logging in with the same credentials from different computers always restores the same preferences.

Options

The Options menu sets up several visual and context depending preferences.

Toolbar:

- Default: Shows labels and icons as defaulted in the GTK configuration.
- Text and Icons: Shows labels and icons in the tool bar.
- Icons: Shows icons only in the tool bar.
- Text: Shows labels only in the tool bar.

Form:

- Save Column Width: Check box to enable saving of manually adjusted widths of columns in lists and trees.
- Save Tree Expanded State: Check box to enable saving of expanded and selected nodes in trees/lists.
- Spell Checking: Check box to enable spell checking in fields.

PDA Mode:

When activated, the client display in a condensed mode.

Search Limit:

Open a dialog to set up the maximum number of records displayed on a list.

Check Version:

Check box to enable the check of new bug-fix version.

Help

Keyboard Shortcuts...:

Shows the information dialog of the predefined keyboard shortcut map.

- Edition Widgets: Shows shortcuts working on text entries, relation entries and date/time entries.

About...:

License, Contributors, Authors of Tryton

1.2.7 Tool Bar

The tool bar contains the functionalities linked to the current tab. Some operations are working with one record or with a selection of *records*. In *form view* the actual record is selected for operations. In *tree view* all selected records are used for operations.

New:

Creates a new record.

Save:

Saves the actual record.

Switch View:

Switches the actual view aspect to:

- *Form view*
- *Tree view*
- *Graph view*

Not all views provide all aspects.

Reload/Undo:

Reloads the content of the actual tab. Undoes changes, if save request for the current record is denied.

Duplicate:

Duplicates the content of the actual record in a newly created record.

Delete:

Deletes the selected or actual record.

Previous:

Goes to the last record in a list (sequence).

Next:

Goes to the next record in a list (sequence).

Search:

Goes to the search widget.

View Logs...:

Shows generic information of the current record.

Show revisions...:

Reload the current view/record at a specific revision.

Close Tab:

Closes the current tab. A Request *Dialog* opens in case of unsaved changes.

Attachment:

The attachment item handles the document management system of Tryton which is able to attach files to any arbitrary *model*. On click it opens the attachments *dialog*. The default dialog shows a list view of the attached files and links.

Actions...:

Shows all actions for the actual view, model and record.

Relate...:

Shows all relate view for the actual view, model and record.

Report...:

Shows all reports for the actual view, model and record.

E-Mail...:

Open an editor to send an email related to the actual record.

Print...:

Shows all print actions for the actual view, model and record.

Copy URL:

Copy the URL of the form into the clipboard.

Export Data...:

Export of current/selected records into *CSV*-file or open it in Excel.

- Predefined exports
 - Choose preferences of already saved exports.
- All Fields: Fields available from the model.
- Fields to export: Defines the specific fields to export.
- Options:
 - Save: Save export as a CSV file.
 - Open: Open export in spread sheet application.
- Add field names: Add a header row with field names to the export data.
- Actions:
 - Add: Adds selected fields to *Fields to export*.
 - Remove: Removes selected fields from *Fields to export*.
 - Clear: Removes all fields from *Fields to export*.
 - Save Export: Saves field mapping to a *Predefined export* with a name.
 - Delete Export: Deletes a selected *Predefined export*.
 - OK: Exports the data (action depending on *Options*).
 - Cancel

Import Data...:

Import records from *CSV*-file.

- All Fields: Fields available in the model (required fields are marked up).
- Fields to Import: Exact sequence of all columns in the CSV file.
- File to Import: File *dialog* for choosing a CSV file to import.
- CSV Parameters: Setup specific parameters for chosen CSV file.
 - Field Separator: Character which separates CSV fields.
 - Text Delimiter: Character which encloses text in CSV.
 - Encoding: *Character encoding* of CSV file.
 - Lines to Skip: Count of lines to skip a headline or another offset.
- Actions:
 - Add: Adds fields to *Fields to Import*.
 - Remove: Deletes fields from *Fields to Import*.
 - Clear: Removes all fields from *Fields to Import*.

- Auto-Detect: Tries to auto detect fields in the *CSV File to Import*.
- OK: Proceeds the data import.
- Cancel

1.2.8 Widgets

There are a several widgets used on Tryton in client side. The follow sections will explains some of them.

Date/DateTime/Time Widgets

These widgets have several key shortcuts to quickly modify the value. Each key increases if lower case or decreases if upper case:

- y: by one year
- m: by one month
- w: by one week
- d: by one day
- h: by one hour
- i: by one minute
- s: by one second

The = key sets the widget value to the current date and time.

TimeDelta Widgets

This widget represent a duration using different symbol of time separated by space:

- Y: for years (default: 365 days)
- M: for months (default: 30 days)
- w: for weeks (default: 7 days)
- d: for days (default: 24 hours)
- h: for hours (default: 60 minutes)
- m: for minutes (default: 60 seconds)
- s: for seconds (default: 1 seconds)

The hours, minutes and seconds are also represented as H:M:s.

For example: 2w 3d 4:30 which represents: two weeks, three days and four and an half hours.

The value of each symbol may be changed by the context of the widget. For example, a day could be configured as 8 hours.

Search Widget

The search widget adds the ability to easily search for records on the current tab. This widget is visible only on *tree view*.

The Syntax

A query is composed of search clauses. A clause is composed of a field name (with `:` at the end), an operator and a value. The field name is optional and defaults to the record name. The operator is also optional and defaults to `like` or `equal` depending on the type of the field. The default operator is `=` except for fields of type `char`, `text` and `many2one` which is `ilike`.

Field Names

All field names shown in the *tree view* can be searched. Field names must be followed by a `:`.

For example: `Name:`

If the field name contains spaces, it is possible to escape it using double quotes.

For example: `"Receivable Today":`

Operators

The following operators can be used:

- `=`: equal to
- `<`: less then
- `<=`: less then or equal to
- `>`: greater then
- `>=`: greater then or equal to
- `!=`: not equal
- `!`: not equal or not like (depending of the type of field)

For example: `Name: != Dwight`

Note: The `ilike` operator is never explicit and `%` is appended to the value to make it behaves like `starts with`.

Values

The format of the value depends on the type of the field. A list of values can be set using `;` as separator.

For example: `Name: Michael; Pam`

It will find all records having the `Name` starting with `Michael` or `Pam`.

A range of number values can be set using `...`

For example: `Amount : 100..500`

It will find all records with `Amount` between `100` and `500` included.

There are two wildcards:

- `%`: matches any string of zero or more characters.
- `_`: matches any single character.

It is possible to escape special characters in values by using double quotes.

For example: `Name: "Michael:Scott"`

Here it will search with the value `Michael:Scott`.

Clause composition

The clauses can be composed using the two boolean operators `&` (for **logical conjunction**) and `|` (for **logical disjunction**). By default, there is an implicit `&` between each clause if no operator is specified.

For example: `Name: Michael Amount: 100`

is the same as `Name: Michael & Amount: 100`

The `&` operator has a highest precedence than `|` but you can change it by using parenthesis.

For example: `(Name: Michael | Name: Pam) & Amount: 100`

is different than `Name: Michael | Name: Pam & Amount: 100`

which is evaluated as `Name: Michael | (Name: Pam & Amount: 100)`

RichText Editor

This feature create a rich text editor with various features that allow for text formatting. The features are:

- **Bold**: On/off style of bold text
- **Italic**: On/off style of italic text
- **Underline**: On/off style of underline text
- **Choose font family**: Choice from a combo box the desired font family
- **Choose font size**: Choice from a combo box the desired size font
- **Text justify**: Choice between four options for alignment of the line (left, right, center, fill)
- **Background color**: Choose the background color of text from a color palette
- **Foreground color**: Choose the foreground color of text from a color palette

Besides these features, it can change and edit text markup. The text markup feature has a similar HTML tags and is used to describe the format specified by the user and is a way of storing this format for future opening of a correct formatted text. The tags are explain follows:

- **Bold**: Tag `b` is used, i.e. `text`
- **Italic**: Tag `i` is used, i.e. `<i>text</i>`
- **Underline**: Tag `u` is used, i.e. `<u>text</u>`
- **Font family**: It is a attribute `font-family` for `span` tag, i.e. `text`
- **Font size**: It is a attribute `size` for `span` tag, i.e. ` text`

- Text Justify: For justification text is used paragraph tag `p`. The paragraph tag is used to create new lines and the alignment is applied across the board. Example: `<p align='center'>some text</p>`
- Background color: It is a attribute `background` for `span` tag, i.e. `text`
- Foreground color: It is a attribute `foreground` for `span` tag, i.e. `text`

1.2.9 CSS

The client can be styled using the file `theme.css`.

Here are the list of custom selectors:

- `.readonly`: read only widget or label
- `.required`: widget or label of required field
- `.invalid`: widget for which the field value is not valid
- `headerbar.profile-<name>`: the name of the connection profile is set on the main window

For more information about style option see [GTK+ CSS](#)

1.2.10 Appendix

Configuration File

```
:file:`~/.config/tryton/x.y/tryton.conf` # General configuration
:file::~~/.config/tryton/x.y/accel.map` # Accelerators configuration
:file::~~/.config/tryton/x.y/known_hosts` # Fingerprints
:file::~~/.config/tryton/x.y/ca_certs` # Certification Authority (http://docs.python.
↪org/library/ssl.html#ssl-certificates)
:file::~~/.config/tryton/x.y/profiles.cfg` # Profile configuration
:file::~~/.config/tryton/x.y/plugins` # Local user plugins directory
:file::~~/.config.tryton/x.y/theme.css` # Custom CSS theme
```

Note: `~` means the home directory of the user. But on Windows system it is the `APPDATA` directory.

1.3 Glossary

Actions

An *action* is a function which is triggered by a user intervention. *Actions* are called from activating menu items or pushing buttons. Actions often provide *wizards*.

Board

The *board* is a type of *views* able to handle other views. This view type is not documented or not used for now.

Character Encoding

See [\[WP-ENCOD\]](#)

CSV

File format for Comma Separated Values. See [\[WP-CSV\]](#)

Data

Data means information content produced by users.

Dialog

A *dialog* is a *popup* window, which overlays other windows and request user interaction. *Dialogs* are used to set up special *actions*.

Fields

Fields are attributes of a *data object*. *Fields* are represented as table fields in relational databases.

Form

The *form* is the general type of *views* used in Tryton. The *form* provides several modes for presenting *data*:

- *Form View*
- *Tree View*
- *Graph View*

Form View

The *form* is a mode of *views*, which displays single *records* of data.

Graph View

Graph view is a mode of *views* to show sets of data in a diagram. *Graph views* can be pie-charts or bar-charts.

Main Frame

The *main frame* is a huge part arranged in the center of the *Tryton client*. Using the *Tryton client* means mainly using the *main frame* part. It contains *tabs* to organize and to show different *views*.

Model

A *model* describes how data is represented and accessed. Models formally define records and relationships for a certain domain of interest.

Modules

Modules are enclosed file packages for the *Tryton server*. A *Module* defines the *Model*, the presentation of the information (*views*), functions, *actions* and default presets. Additionally *modules* may provide standardized data like ISO names for countries. *Modules* in Tryton are build up generically. That is, they are constructed as simple as possible to provide the desired functionality.

Plugins

A *plugin* is an add-on module for the *Tryton client*.

Popup

A small window which pops up the main window.

Records

A *record* is a singular dataset in a *Model*. *Records* are represented as lines or *records* in a relational database table.

Tabs

Tabs are *widgets* to arrange different contents side by side. They are used to switch quickly between different domains of interest. Tryton uses *tabs* in two layer:

- A tabbed *Main Frame*.
- Tabs inside *Views*.

The main frame consists of *tabs* that embed the main menu and all views to an appropriate *model*. The other type of *tabs* is used inside of *views* to split them into visual domains of the same model. These *tabs* are used for structuring contents of one model to different sub headings.

Three-Tiers

A *three-tiers* application framework like Tryton, is build up of three different software components:

1. The storage or data tier.
2. The logic or application tier.
3. The presentation tier.

The storage tier in the Tryton framework is provided by the PostgreSQL database engine. The application logic tier is provided by *Tryton server* and its *modules*. The presentation tier is mainly provided by the *Tryton client*. In a *three tiers* framework, the presentation tier (client) never connects directly to the storage tier. All communication is controlled by the application tier.

Tree View

Tree view is a mode of *views* showing sets of *data*. *Tree views* can be flat lists or tables as well as tree-like nested lists.

Tryton Client

The *Tryton client* application is the graphical user interface (GUI) of the *Tryton server*.

Tryton Server

The *Tryton server* is the application or logic tier in the *three-tiers* application platform *Tryton*. The *Tryton server* connects the underlying application logic of the different *modules* with corresponding database records. The *Tryton server* provides different interfaces to present the generated information:

- *Tryton client*: (graphical user interface GUI)
- XMLRPC see [WP-XMLRPC]
- WebDAV see [WP-WebDAV]
- OpenOffice

Views

A *view* is the visual presentation of *data*. *Views* resides inside *tabs* in the *main frame* of the *Tryton client*. There are two general types of *views* in Tryton:

1. *Form*
2. *Board*

Each of the view types has different modes to show data. *Views* are built of several *widgets* and provide often additional *actions*. It is also possible to present the same data in different view modes alternately.

Widgets

A *Widget* is a visual element of a graphical user interface (GUI). Some *Widgets* solely show information, others allow manipulation from user side. Example *Widgets* are buttons, check-boxes, entry-boxes, selection lists, tables, lists, trees, ...

Wizards

Wizards define stateful sequences of interaction to proceed complex *actions*. A *wizard* divides the complexity of some actions into several user guided steps.

1.3.1 References

1.4 Release notes

1.4.1 Version 6.8.1 - 2023-05-17

- Bug fixes (see mercurial logs for details)

1.4.2 Version 6.8.0 - 2023-05-01

- Bug fixes (see mercurial logs for details)
- Include events to logs
- Add option to disable thread
- Remove support for Python 3.7
- Add support for Python 3.11
- Save and Restore tree state on List Form
- Do record (pre)validation and saving on selection change for list-form views
- Eagerly read string value of selection fields
- Do not select first record by default
- Use & and | as boolean operator in search widget
- Use toggle button and label on translate dialog
- Manage domain on id in single value list as unique
- Do not validate domain nor enforce uniqueness for empty fields unless they're required or invisible
- Display the number of selected records on xxx2Many

1.4.3 Version 6.6.0 - 2022-10-31

- Bug fixes (see mercurial logs for details)
- Support authentication services

1.4.4 Version 6.4.0 - 2022-05-02

- Bug fixes (see mercurial logs for details)
- Use unittest discover
- Support notification message
- Add option to define logging output location
- Remove TreeViewControl
- Manage optional column
- Default CSV encoding to UTF8 with BOM
- Display the number of selected records
- Humanize the count result
- Add limit to search_count
- Call view_get for board view
- Limit board action domain to active id and ids
- Add support for Python 3.10
- Remove support for Python 3.6

- Manage creatable attribute of view

1.4.5 Version 6.2.0 - 2021-11-01

- Bug fixes (see mercurial logs for details)
- Support grouping attribute
- Support monetary and symbol field attributes
- Support digits from relation field
- Allow PYSON Expression as key for PYSON In with dict object
- Support Binary field in CSV import/export

1.4.6 Version 6.0.0 - 2021-05-03

- Bug fixes (see mercurial logs for details)
- Support printing zip archive
- Add direct print to UNIX
- Add attachment preview
- Support document widget
- Support icon of type URL
- Allow sending email with existing attachments
- Add paths and model context to action data
- Add indentation in CSV export of tree
- Manage which records to use for actions
- Handle device cookie
- Add breadcrumb as title of window form
- Display revision on dialog
- Execute report asynchronously
- Add support for Python 3.9
- Support empty value for timedelta converter
- Add interactive search on tree view
- Unify PYSON string format

1.4.7 Version 5.8.0 - 2020-11-02

- Bug fixes (see mercurial logs for details)
- Remove support for Python 3.5
- Format timedelta in CSV export
- Manage symbol widget
- Fallback to model name as title
- Manage deletable and writable state
- Support e-mail template
- Send e-mail via the server
- Support PYSON comparison of date and datetime
- Position copied records based on order
- Allow configuration of default colors for graph and calendar
- Use existing context for get_preferences
- Add context to export URL

1.4.8 Version 5.6.0 - 2020-05-04

- Bug fixes (see mercurial logs for details)
- Support link button on form
- Add URL for the current export
- Add option to export listed records
- Retire App Menu
- Add support for Python 3.8
- Manage readonly dates in calendar view
- Manage editable on calendar view
- Manage model access on calendar view
- Set client title at the end of header bar
- Manage named separator as label
- Add MultiSelection entry to Dict field
- Position new record based on order
- Allow relation field to be selected in CSV export
- Support validate attribute of wizard's Button

1.4.9 Version 5.4.0 - 2019-11-04

- Bug fixes (see mercurial logs for details)
- Support visual context on tree view
- Support multiselection field
- Support dot notation on PYSON Eval
- Order Dict keys by sequence
- Support relation fields in domain parser
- Add Dict widget for tree view
- Add handle to reordable line of tree views

1.4.10 Version 5.2.0 - 2019-05-06

- Bug fixes (see mercurial logs for details)
- Add domain inversion for reference fields
- Add list-form view
- Remove support for Python 3.4
- Replace ColorSelectionDialog by ColorChooserDialog
- Support HTML widget
- Use locale format for data import and export
- Add day view on calendar
- Add CSV exports into print toolbar
- Display unread and total of notes
- Support badge on icon
- Use resources method
- Replace Arrow by Image
- Remove Fast Tabbing option
- Remove usage of focus_chain
- Add shortcuts for switch and remove on One2Many
- Rename roundup.url configuration to bug.url
- Remove send error to bug tracker
- Replace Table by Grid
- Use separate XML parser in Views
- Use between numbers in filter box
- Rename email and calendar file
- Use markup for styling
- Replace ImageMenuItem by MenuItem

- Replace ComboBoxEntry by ComboBox or ComboBoxText
- Replace GObject.idle_add by GLib.idle_add
- Remove positional arguments for PyGObject
- Remove pygtkcompat
- Implement Reference cell with Selection and Many2One cells
- Replace CellRendererBinary with clickable pixbuf cells
- Allow cell to have multiple renderers
- Fill 'to' with 'from' value on filter box
- Add tab domain in URL
- Use Popover for search filter
- Display version on login dialog
- Use tab key for tab navigation
- Use ShortcutsWindow
- Improve default size of widgets and dialog
- Use record and field properties of widget instead of method arguments
- Reduce offset when no search result
- Ensure URL scheme of bus (issue7792)
- Add drag & drop support on binary widget

1.4.11 Version 5.0.0 - 2018-10-01

- Bug fixes (see mercurial logs for details)
- Use dropdown for attachment
- Apply factor on domain parser
- Use tab name in CSV Export/Import
- Add bus management
- Use CSS to style label
- New icons
- Support timestamp field
- Add support for Python 3.7
- Use GtkApplication
- Validate dictionary items
- Add (i)like support to domain inversion
- Use Python 3
- Improve loading strategy by using views
- Use non conflicting keyboard shortcuts in sync with sao
- Use the context to get the suffix of window name

- Call on_change methods after setting default dtstart on calendar view

1.4.12 Version 4.8.0 - 2018-04-23

- Bug fixes (see mercurial logs for details)
- Manage active field on search widget
- Reset record to its original state when discarding the popup window
- Add support of expand attribute on group tag
- Add the context model name in the screen context
- Use limit to expand tree
- Add keyword attribute to button tag
- Support field name on image tag
- Add option to check new version
- Show related record names for all windows
- Remove support of GTK+ 2.0
- Add icons on Many2One in editable tree
- Use translated values when exporting Reference and Selection fields

1.4.13 Version 4.6.0 - 2017-10-30

- Bug fixes (see mercurial logs for details)
- Allow to export model and record name of Reference fields
- Re-position buttons on Binary/Image widget
- Improve treeview headers
- Update states of both toolbar and menu
- Improve toolbar order
- Make readonly, required and invalid widget themable
- Load user CSS theme
- Use profile name or login details in title

1.4.14 Version 4.4.0 - 2017-05-01

- Bug fixes (see mercurial logs for details)
- Verify certificate with default CA
- Replace URL entry by a toolbar button

1.4.15 Version 4.2.0 - 2016-11-28

- Bug fixes (see mercurial logs for details)
- Add support for GTK+ 3.0
- Add PYSON Widget
- Show records names in wizard title
- Add support for datetime_field to Reference
- Show non editable widget as insensitive
- Move info-bar at the bottom
- Manage readonly on fields translation dialog
- Restore tab default order when no manual change
- Call autocompletion when setting record value
- Remove database management
- Limit readonly state for xxx2Many
- Show records names in relate window title
- Add support for count on Action Window Domains
- Manage custom login process
- Add clear icon on Many2One
- Re-position icons on Many2One widget

1.4.16 Version 4.0.0 - 2016-05-02

- Bug fixes (see mercurial logs for details)
- Improvement of charts design
- Timedelta uses 30 days for month
- Add all available encoding for import/export
- Add CSV parameters to export window
- Add DnD on import window
- Manage context model of ir.action.act_window
- Add Note
- Manage view_ids on tree view
- Use Apple key for copy/past of list

1.4.17 Version 3.8.0 - 2015-11-02

- Bug fixes (see mercurial logs for details)
- Use italic labels for editable fields
- Escape ‘_’ in button string and page string
- Export selected records only
- Use HTML for RichText widget
- Change Progressbar to work with float between 0 and 1
- Add Fast Tabbing option
- Remove colors on widgets
- Bold label of required fields
- Explicit error message for invalid record
- Add reversed operators to PYSON expressions

1.4.18 Version 3.6.0 - 2015-04-20

- Bug fixes (see mercurial logs for details)
- Hide columns containing always the same value
- Remove Tabs Position option
- Manage product attribute
- Remove float_time widget
- Add TimeDelta field
- Improve date/time widgets
- Remove datetime widget on list/tree view
- Allow to put many times the same field on tree view
- Add search completion on dictionary widget
- Merge host and port field in profile editor
- New color scheme for graph
- Replace `img_{width,height}` by width and height attributes

1.4.19 Version 3.4.0 - 2014-10-20

- Bug fixes (see mercurial logs for details)
- Add DnD on export window
- Allow to override predefined export
- Prefill export window with current view fields
- Manage field context on Group value
- Add export of selection string

- Load plugins from local user directory
- Manage tree_state attribute
- Add simple tree_state support on form view
- Change range operator of search widget to be included
- Explicitly set value of parent field

1.4.20 Version 3.2.0 - 2014-04-21

- Bug fixes (see mercurial logs for details)
- Add option to show revisions
- Add a multi selection widget for many2many
- Remove auto-refresh
- Add support of domain for non-relation field
- Allow drag & drop on the attachment button
- Replace sha widget by password
- Add Len to PYSON
- Use a pool of connection
- Manage client actions from button and wizard
- Add tree_invisible attribute to button in tree view
- Add buttons of the view in actions menu
- Don't evaluate anymore relate action with the record
- Paste on editable list create new records if needed
- Drop support of Python 2.6
- Allow to search on rec_name of Reference fields
- Use local timezone
- Sanitize report file extension

1.4.21 Version 3.0.0 - 2013-10-21

- Bug fixes (see mercurial logs for details)
- Add factor on number widgets
- Add calendar view
- Add URL entry
- Remove request

1.4.22 Version 2.8.0 - 2013-04-22

- Bug fixes (see mercurial logs for details)
- Manage dynamic label
- Manage prefix, suffix on tree view
- Manage selection_change_with on Selection and Reference fields
- Add Dict fields on form
- Remove Goto
- Add global search
- Add toggle button for menu
- Replace shortcuts by menu favorites
- Move Plugins into toolbar Actions
- Add url to list view
- Add dynamic icon for url
- Add completion on Many2One, Many2Many and One2Many
- Add bookmark on search widget
- Manage domains on Action Window
- Use range for Date/Time fields in filter box
- Allow multi-selection for Selection field in filter box

1.4.23 Version 2.6.0 - 2012-10-22

- Bug fixes (see mercurial logs for details)
- Allow to paste in editable list view
- Manage readonly state on group
- Remove “Please wait” box
- Refactorize date widgets
- Manage tuple on Reference
- Add constant interpolation on line graph
- Make Import/Export Data no-modal
- Deactivate switch view button when there is only 1 view
- Manage create/delete field access
- Add dynamic size limit on the One2Many, Many2Many and Char
- Search only on fields in XML view
- Cache action keyword
- Always use cached views
- Manage model access

- Manage time format
- Deactivate attachment button when record is not created

1.4.24 Version 2.4.0 - 2012-04-23

- Bug fixes (see mercurial logs for details)
- Remove workflow
- **Improve contextual menu:**
 - Actions for all relation fields on list view
 - Use icons
 - Add attachments entry
- Use RPCProgress almost everywhere
- Add a simple search box
- New domain parser using shlex
- Better translation dialog box
- Add the richtext widget for WYSIWYG Editor on text fields
- Add a record pool to prefetch more records
- Add Time widget
- Add support for fuzzy_translation
- Activate save button on editing
- Refactor set/get and set/get_client of fields
- Display binary size alongside the buttons in treeview
- Set correctly the focus on tab switching
- Add shortcut to focus search entry
- Add binary field to tree views
- **Improve board view:**
 - use the same search widget as form
 - link double click to open popup

1.4.25 Version 2.2.0 - 2011-10-24

- Bug fixes (see mercurial logs for details)
- New search widget
- Improve memory management of Binary fields
- Support buffer for Binary fields
- Remove delete on Escape in editable tree
- Use JSON-RPC
- Limit size of field when possible

- Add xalign and yalign as fields attributes
- Convert many popup to be no-modal
- **Add window manager for:**
 - replace current window
 - prevent simmilar window
- Merge and review toolbars in form and board
- Drop support of Python 2.5
- Use the same design for Many2Many than One2Many
- Allow resize columns smaller than the header

1.4.26 Version 2.0.0 - 2011-04-26

- Bug fixes (see mercurial logs for details)
- Popup form dialog has 3 buttons (close, ok, new)
- New UI layout
- Add DnD on tree view
- Merge tree and list views
- Remove generic default value on right-click
- Made numpad locale aware
- Selection widget used for many2one dynamically change their content following the domain specification
- Add open button on binary and image widgets
- Hide buttons on image widget if readonly
- Added a connection manager à la gajim
- Fix warning in wizards
- Added possibility to use server-side icons
- Added additional gtk.Entry for filename on BinaryField
- Display deleted lines in One2Many and Many2Many
- Handle URL
- Add communication between boards
- Added domain inversion feature
- Handle loading attribute on fields
- Use default format for value in wizard form
- Add One2One field

1.4.27 Version 1.8.0 - 2010-11-01

- Bug fixes (see mercurial logs for details)
- More fully integrate GTK menubar for Mac OS
- Allow to configure search limit
- Add Previous/Next on list view
- Set non-applicable form controls in one2many view to be insensitive

1.4.28 Version 1.6.0 - 2010-05-08

- Bug fixes (see mercurial logs for details)
- Don't stop wizard execution when exception occurs
- Use ir.attachment view instead of a custom one
- Add fingerprint and CA check for SSL connection
- Use lazy load in Import/Export windows
- Validate record before switching view
- **Refactoring:**
 - Better naming in model
 - Group extends list
 - Add an index to Group
 - New common windows for dialog of many2one, one2many and many2many
 - one2many and many2many dialog use the same screen than the widget
 - Reduce the number of option in Screen
 - Remove RPCProxy to handle logout exception on every server call
 - Better naming of event signals
 - Fix on_change detection for many2many for issue1400
- Add PySON to replace python string evaluation
- Don't show "Spell Checking" option if gtkspell is not present
- Use the same internal model for many2many and one2many fields
- Remove egenix-mx-base and replace it by python-dateutil
- Add cache to safe_eval
- Use versioned configuration directory
- Next and Previous scroll per page on list and don't loop
- Add AccelGroup on search windows (CTRL+Return)
- Use same keyboard shortcut for xxx2many than for other fields

1.4.29 Version 1.4.0 - 2009-10-19

- Bug fixes (see mercurial logs for details)
- Handle datetime_field in xxx2Many
- Add new safe_eval
- Ask previous password for set_preferences on password change
- Add “Statusbar” option
- Add default filename for database backup
- Add checkbox on restore to update database
- Add ‘login.host’ options to hide server connection
- Handle required attribute with local domains
- Allow to run wizard in tabs
- Remove statusbar on form for more space
- Add “Change Accelerators” option
- Use gzip in psocket
- Use the report name to create the temporary file to open it
- Allow to store wizard size
- Add reset default on fields
- Store in config default width and height of main window
- Added arrow navigation if supported by gtk
- Add ‘starts with’ and ‘ends with’ on char search
- Handle domain with ‘=’ or ‘!=’ as operator on selection
- Extend fields domain with local domains
- Improve float time widget to handle year, month, week and day and handle float_time attribute for contextual time conversion

1.4.30 Version 1.2.0 - 2009-04-20

- Bug fixes (see mercurial logs for details)
- Make graph works also with datetime
- Add edition on Many2Many
- Fix open in csv export to use file actions
- Update client labels at language change
- Handle datetime_field in Many2One
- Set readonly on records with _datetime in the context
- Display values of reference fields even if there is no model
- Allow to directly print or create email with reports
- Handle invisible states on list view

- Add user warnings
- Add Model in logs
- Allow to duplicate many records at once
- Improve netrpc communication speed
- Improve date widget to display mask only when having focus
- Fix for host with IPv6 enable but without default IPv6 route
- Add desktop entry
- Add win32 single executable
- Allow egg installation

1.4.31 Version 1.0.0 - 2008-11-17

- Initial release

BIBLIOGRAPHY

[WP-XMLRPC] <http://en.wikipedia.org/wiki/Xmlrpc>

[WP-WebDAV] <http://en.wikipedia.org/wiki/Webdav>

[WP-CSV] http://en.wikipedia.org/wiki/Comma-separated_values

[WP-ENCOD] http://en.wikipedia.org/wiki/Character_encoding

INDEX

A

Actions, [12](#)

B

Board, [12](#)

C

Character Encoding, [12](#)

CSV, [12](#)

D

Data, [13](#)

Dialog, [13](#)

F

Fields, [13](#)

Form, [13](#)

Form View, [13](#)

G

Graph View, [13](#)

M

Main Frame, [13](#)

Model, [13](#)

Modules, [13](#)

P

Plugins, [13](#)

Popup, [13](#)

R

Records, [13](#)

T

Tabs, [13](#)

Three-Tiers, [13](#)

Tree View, [14](#)

Tryton Client, [14](#)

Tryton Server, [14](#)

V

Views, [14](#)

W

Widgets, [14](#)

Wizards, [14](#)