The *Company Module* allows the business that is using Tryton to be represented inside Tryton. This may be a single company or multiple companies. It also allows recording which employees belong to which companies and what their organisational structure is.
The *Companies* and *Employees* on Tryton can be found under the [Company] main menu item.

**Tip:** For employees you can use the *Supervised by* item in *Open related records* to get a list of the employees that are supervised by the currently selected employee.

### 1.1 Setting your current company and employee

If you want to change your current *Company* or *Employee*, you should do this by opening your preferences and selecting the desired company and employee.

**Note:** If you try and change your current company or employee from inside the [Administration → Users → Users] menu entry then it will only change when you next login, or update your preferences.
The *Company Module* introduces some new concepts, and extends other existing concepts:

### 2.1 Company

The *Company* is the main concept provided by the *Company Module*. In Tryton it is a *Party* that represents the company, or organisation, which the users of the system are members of.

See also:
Companies can be found by opening the main menu item:

*Company → Companies*

### 2.2 Employee

The *Employee* is another important concept introduced by the *Company Module*. In Tryton it is a *Party* that is employed by, or works for, one of the *Companies* which the users are members of.

Employees can be organised into a structure by setting each employee’s supervisor.

See also:
A list of the employees for all the companies can be found from the main menu item:

*Company → Employees*

### 2.3 User

The *Company Module* extends the User concept so that each user can be associated with one or more *Companies* and a set of *Employees*. From these a user then chooses a current company and employee.

This choice directly affects what data the user has access to in Tryton, as models will often link records to a company and restrict access based on the user’s setup.

See also:
The *User* concept is introduced by the Res Module.
2.4 Party

The *Company Module* adds an addition report that can be used with *Parties*.

See also:

The *Party* concept is introduced by the *Party Module*.

2.4.1 Reports

Letter

This report is a document that can be used as the starting point for a letter to the selected *Party*. The letter that is created is preformatted with information about the party, the User and the user’s current *Company*. The only thing that then needs to be added is the main contents of the letter.
3.1 Company Multi-Values

class trytond.modules.company.model.CompanyMultiValueMixin
A MultiValueMixin that makes it simpler to create MultiValue fields based on the Company in the context. It does this by including the company from the context in the pattern by default.

class trytond.modules.company.model.CompanyValueMixin
A ValueMixin used to store the values of a CompanyMultiValueMixin.

3.2 Employee Fields

trytond.modules.company.model.employee_field(string[, states][, company])
A function that returns a Many2One field. This field is intended to be used to store the Employee that performed some action on the Model that contains the Field.

Parameters
- **string** – The string that is used as the label for the field.
- **states** – The states in which the field will be read-only.
- **company** – The name of the field that contains the company.

Returns The employee Many2One field.

@trytond.modules.company.model.set_employee(field[, company])
Used to decorate methods which need to record the employee that last ran them. The specified field is updated with the User’s current Employee, but only if the employee works for the company.

Parameters
- **field** – The name of the field to set to the user’s current employee.
- **company** – The name of the field that contains the company. Defaults to ‘company’.

@trytond.modules.company.model.reset_employee(*fields)
Used to decorate methods which indicate that the document is now in a state where the action has not yet been performed, so the Employee should be cleared from the fields.

Parameters **fields** – The names of the fields that should have the employee removed.
3.3 Company Reports

class trytond.modules.company.company.CompanyReport
    A report that places the Company of the record in the header key and adds it to the context.
B

  built-in function
  trytond.modules.company.model.employee_field(), 7
  trytond.modules.company.model.reset_employee(), 7
  trytond.modules.company.model.set_employee(), 7

T

  trytond.modules.company.company.CompanyReport
    (built-in class), 8
  trytond.modules.company.model.CompanyMultiValueMixin
    (built-in class), 7
  trytond.modules.company.model.CompanyValueMixin
    (built-in class), 7
  trytond.modules.company.model.employee_field()
    built-in function, 7
  trytond.modules.company.model.reset_employee()
    built-in function, 7
  trytond.modules.company.model.set_employee()
    built-in function, 7