

# iTop Documentation for Administrators

- [The Administrator's Guide](#) explains how to install, upgrade and operate an iTop instance
- [The OQL Reference Guide](#) explains the syntax of the OQL language
- [The Implementation Guide](#) explains step-by-step how to load the data in iTop for creating a production environment.

## iTop 2.0 Administrator's Guide

### About iTop

iTop stands for **IT Operational Portal**. It is an Open Source web application for the day to day operations of an IT environment. iTop was designed with the ITIL best practices in mind but does not dictate any specific process, the application is flexible enough to adapt to *your* processes whether you want rather informal and pragmatic processes or a strict ITIL aligned behavior.

At the heart of iTop is the CMDB (Configuration Management Data Base), this is originally the first part of iTop that was developed. Then came the tickets and all the derived processes.

The belief behind iTop is that a CMDB *must be* an operational tool. The only way for a CMDB to be accurate and up to date is to be used day-to-day by the IT teams (support agents, IT engineers, etc.). Moreover, the more the CMDB is integrated with other IT tools (monitoring systems, reporting tools, automated inventory, etc.), the better.

Using iTop you can:

- Document your IT infrastructure and all the relationships between the various pieces and stakeholders of the infrastructure (servers, applications, network devices, virtual machines, contacts, locations...)
- Manage incidents, user requests, planned outages...
- Document IT services and contracts with external providers including service level agreements
- Export all the information in a manual or scripted manner
- Import (manually and using scripts) or synchronize/federate any data from external systems

iTop can be used by different types of persons:

- Help Desk agents
- Support engineers (1st level, 2nd level...)
- Service managers
- IT managers
- End-users: a simplified “portal” interface is available to let them submit their requests directly.

iTop is relying on Apache/IIS, MySQL and PHP, so it can run on whatever operating system supporting those applications. It has been tested on Windows, Linux (Debian, Ubuntu and Redhat). The application also runs on Solaris and MacOS X. Since iTop is a web based application you don't need to deploy any client software on each user's PC. A simple web browser is enough (IE 8+, FF 3.5+, Chrome or Safari 5+).

## Licensing

iTop is licensed under the terms of the GNU Affero General Public License Version 3 as published by the Free Software Foundation. This gives you legal permission to copy, distribute and/or modify iTop under certain conditions. Read the 'license.txt' file in the iTop distribution. iTop is provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

## Installing iTop

### Hardware requirement

| Operating System | Resource  | Minimum                | Recommended         |
|------------------|-----------|------------------------|---------------------|
| Linux            | Disk      | 5 Gb                   | 20 Gb               |
|                  | RAM       | 1 Gb                   | 2 Gb                |
|                  | Processor | 1 GHz (single Pentium) | 2 GHz + (dual-core) |
| Windows          | Disk      | 5 Gb                   | 20 Gb               |
|                  | RAM       | 1 Gb                   | 2 Gb                |
|                  | Processor | 1 GHz (single Pentium) | 2 GHz + (dual-core) |

The minimum screen size for *using* iTop is 1024\*768 pixels, but the higher resolution the better.

## Downloading iTop

You can download iTop from SourceForge:

<https://sourceforge.net/projects/itop/files/itop>

## Software requirements

iTop is based on MySQL and PHP, it requires **PHP version 5.2.0** or newer and **MySQL version 5.0** or newer, plus of-course a web server: Apache, IIS or any web server supporting PHP 5.

Optional requirements:

- For LDAP authentication iTop requires the PHP LDAP module. If this PHP module is not present iTop will not support LDAP or Active Directory identification.
- For strong encryption of passwords inside iTop, the PHP mcrypt module is required.

## Installing the Prerequisites on Debian or Ubuntu

To install the required software on Debian / Ubuntu, run the following commands (as root):

```
apt-get install apache2
apt-get install mysql-server
apt-get install php5 php5-mysql php5-ldap php5-mcrypt php5-cli
```

## Installing the Prerequisites on Redhat, CentOS or Fedora

To install the required software on Redhat / CentOS / Fedora, run the following commands (as root):

```
yum install httpd
yum install mysql mysql-server
yum install php php-mysql php-mcrypt php-xml php-cli
```

## Installing the Prerequisites on Windows

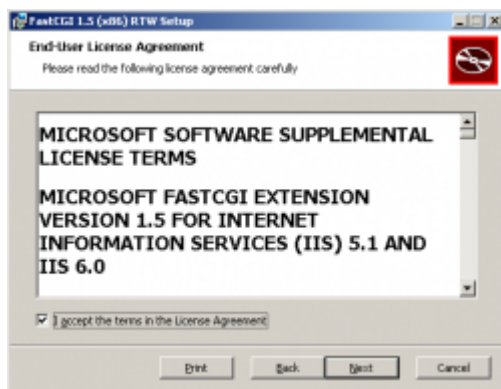
## ITop Installation on Windows using IIS

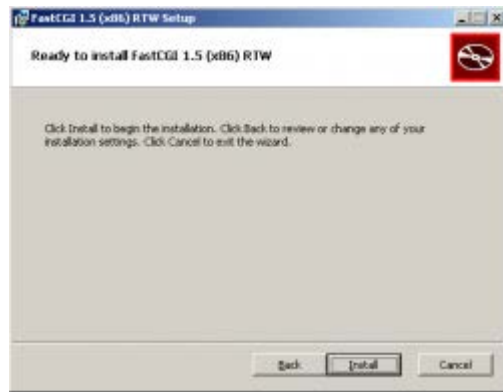
Before running iTop on Windows using IIS, you need to setup IIS to run PHP pages.

### Installing Fast-GCI for IIS 6

IIS 6 is the version of IIS that comes with Windows XP and Windows Server 2003. The process is quite similar for IIS7 (see the [Useful Links & References](#) below)

1. Download IIS's Fast-CGI installer from: <http://www.iis.net/download/FastCGI>. Pick the version (32 or 64-bit) suitable for your system.
2. Start the installation by launching the .msi installer

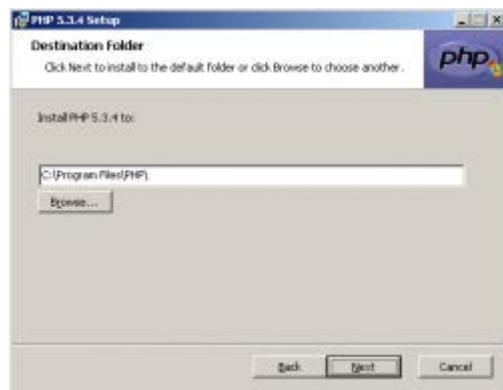
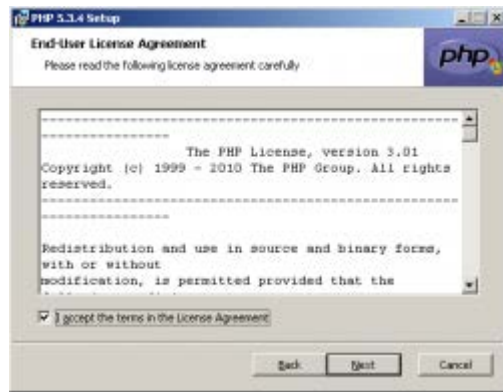




## Installing and configuring PHP

1. Download PHP from <http://windows.php.net/download/>. Select the appropriate version for your system (32 or 64-bit), and pick the Non Thread Safe version. If you don't know between VC6 and VC9, pick the VC6 version.

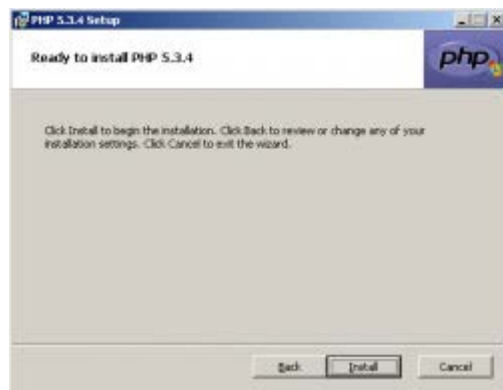
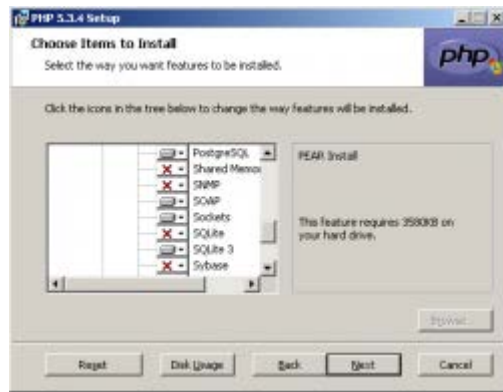




Select the IIS Fast-CGI interface:



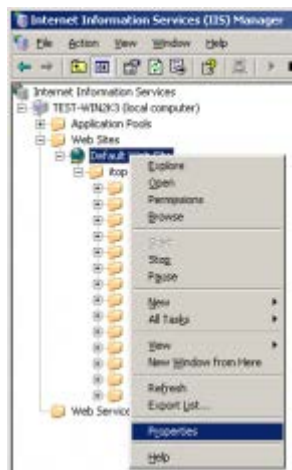
Make sure that you select the “SOAP” extension which is not enabled by default. Also In the “Extras” section, check the “PEAR Install”.



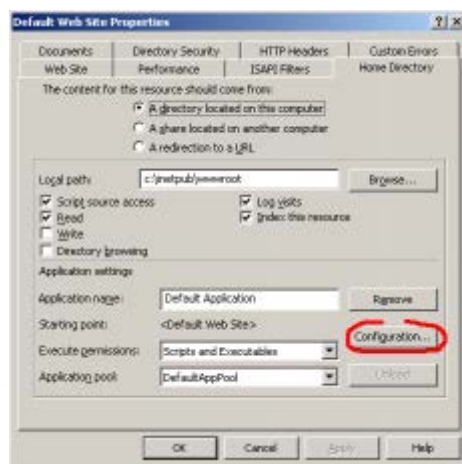
Then register the FastCGI extension for the whole web site:

```
cd %windir%\system32\inetsrv
cscript fcgiconfig.js -add -section:"PHP\" -extension:php
-path:"C:\Program Files\PHP\php-cgi.exe\"
```

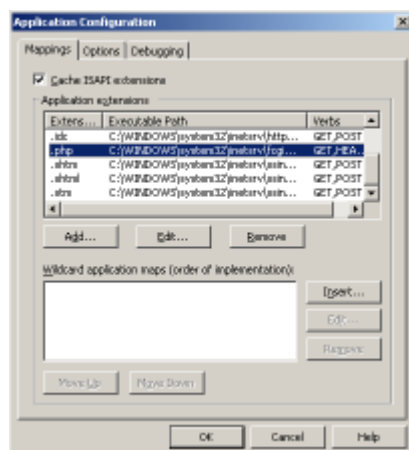
Open the Internet Information Service (IIS) Manager (either from the start menu "Administrative Tools/Internet Information Services (IIS) Manager" or right-click on "My Computer" then "Manage" and "Services And Applications/Internet Information Services (IIS) Manager") and right-click on the "Default Web Site". Then choose "Properties".



Then click on the tab “Home directory” and click on the “Configuration...” button:



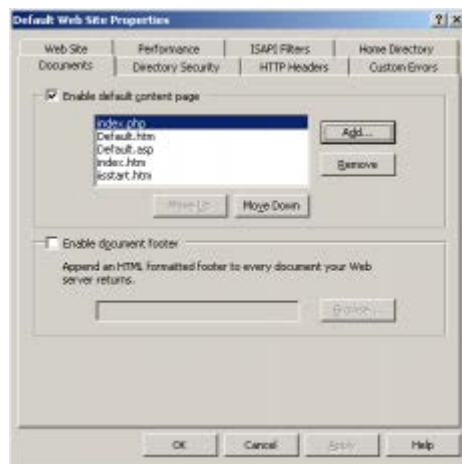
Check in the “Mappings” tab that there is an entry for the “.php” extension.



If not, press the “Add...” button and fill the form as below (adjust the path to the correct location of fcgiext.dll)



Go back to the “Documents” tab and check that “index.php” is present, at the top of the list. if it’s not there, add an entry for it using the “Add...” button.



## Installing MySQL

If you do not already have a MySQL database server, you can install it on the same system as the rest of the application.

1. Go to MySQL's download site: <http://www.mysql.com/downloads/mysql/>
2. Download the installer suitable for your system: select package named “MSI Installer” either the 32 or 64-bit version.
3. Launch the installation, by opening the downloaded .msi file.

## Useful Links & References

- [php.iis.net](http://php.iis.net) Explanations about how to run PHP applications on Windows using IIS
- [Using FastCGI to Host PHP Applications on IIS 6.0](#), by Ruslan Yakushev
- [Install PHP and FastCGI Support on Server Core](#)

- [Enable FastCGI Support in IIS 7 on Windows Server 2008, Windows Server 2008 R2, Windows Vista, or Windows 7](#) by Tali Smith

## Installation wizard

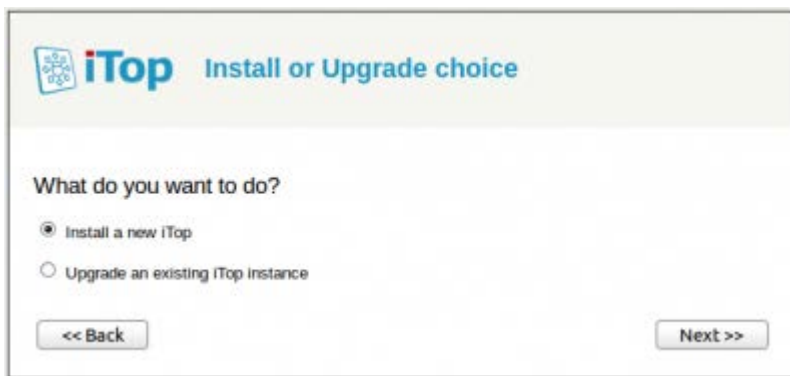
1. Make sure that you have a properly configured instance of Apache/PHP running
2. Unpack the content of the “web” directory in a directory served by your web server (for example /var/www/itop on Linux or C:\inetpub\wwwroot\itop on Windows/IIS).
3. Point your web browser to the URL corresponding to the directory where the files have been unpacked (for example http://localhost/itop) and follow the indications on the screen.

iTop provides a step by step wizard to install the application.

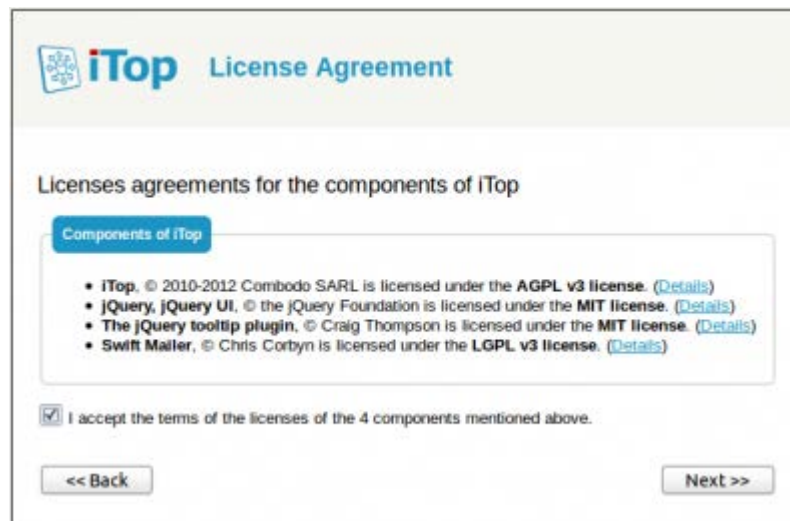
The first step of the installation consists in checking the consistency of the configuration and the prerequisites for MySQL, PHP and optional PHP extensions.



The second step of the installation offers the choice to either upgrade an existing iTop or to install the application from scratch.



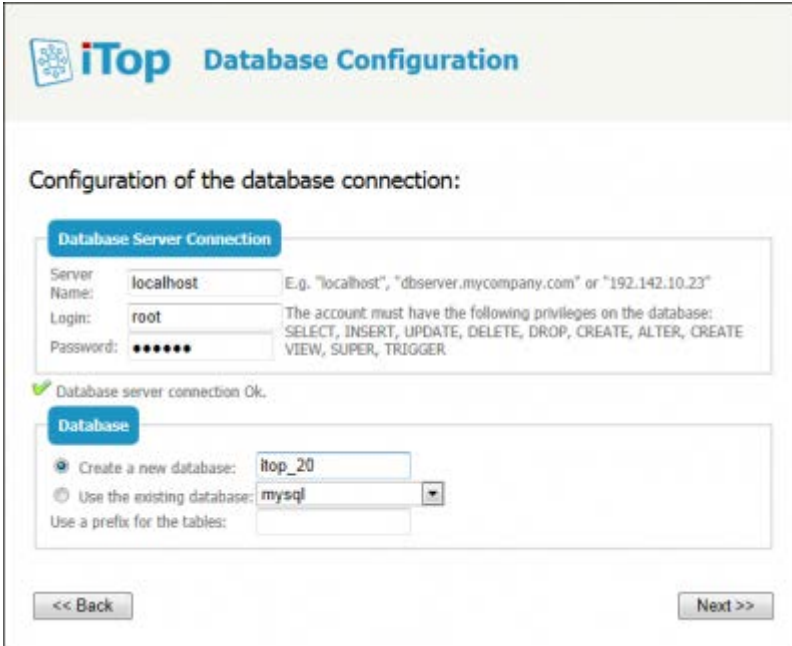
If you select to install a new instance then third step of the wizard will prompt you to accept the license agreement.



The fourth step of the wizard prompts you for the credentials to connect to MySQL and the database to use for installing iTop. Make sure that you supply an account with enough rights for installing and running iTop. This account needs to create tables, triggers and views in the database, as well as drop views in case of upgrade. If you choose to let the setup create a new database for you, this account needs to have enough rights to do it.

You can also specify a prefix to be added to the names of all tables and views created by iTop in the database. This is useful if you want to install several applications inside the same MySQL database.

Note that the database server does not need to be on the same physical server as the PHP/web server it can be a remote host if you prefer to have a two-tier architecture, or connect to an already installed MySQL server.



**iTop Database Configuration**

Configuration of the database connection:

**Database Server Connection**

Server Name:  E.g. "localhost", "dbserver.mycompany.com" or "192.142.10.23"

Login:  The account must have the following privileges on the database: SELECT, INSERT, UPDATE, DELETE, DROP, CREATE, ALTER, CREATE VIEW, SUPER, TRIGGER

Password:

✓ Database server connection Ok.

**Database**

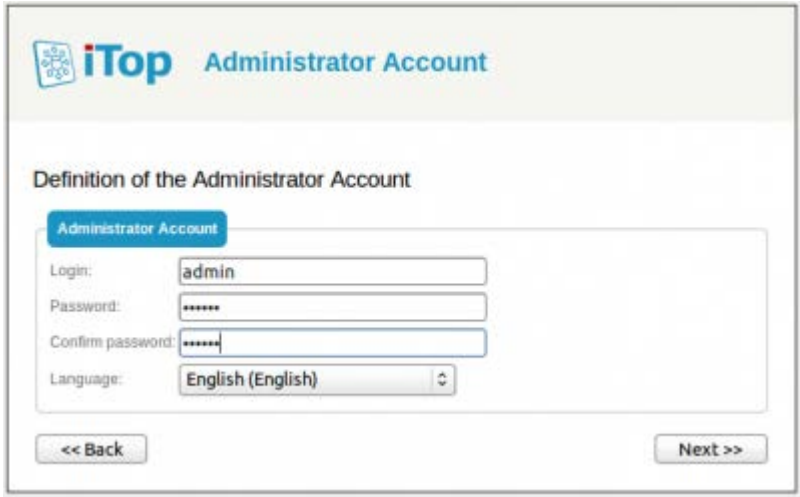
☒ Create a new database:

☐ Use the existing database:

Use a prefix for the tables:

<< Back Next >>

The next step of the wizard prompts you for the information about the administrator account. iTop authentication can rely on external mechanisms (LDAP, Active Directory or JASIG-CAS) but the installation requires that at least one administrator account be created with a local password.



**iTop Administrator Account**

Definition of the Administrator Account

**Administrator Account**

Login:

Password:

Confirm password:

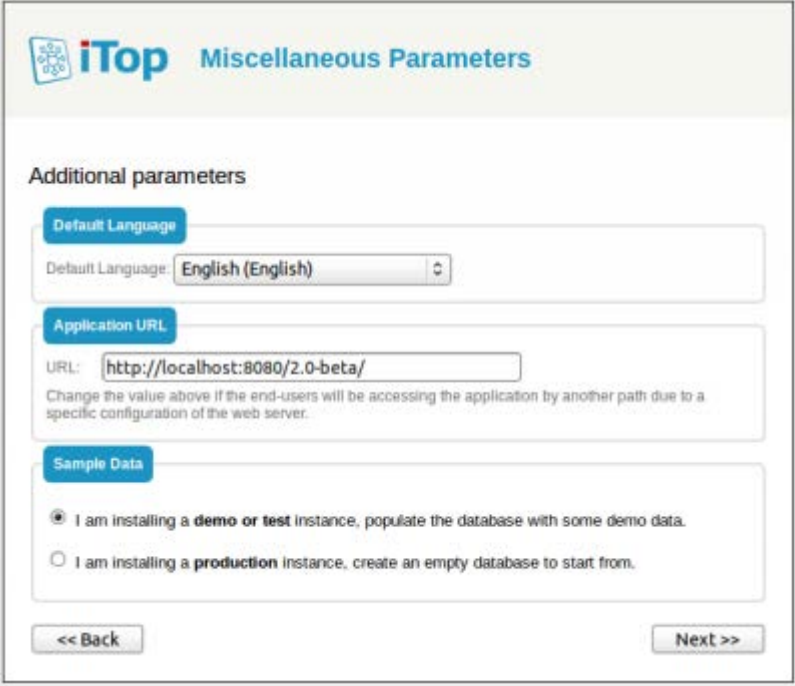
Language:

<< Back Next >>

The next screen prompts your for several miscellaneous information:

- default language: this is the language of the login screen and the default language for newly created user accounts.
- URL to access iTop: change this value if users will be accessing iTop using a different URL (thanks to a DNS alias, a different IP adress, etc.)
- sample data: install sample data if you want to quickly experiment with iTop. Sample data provide some sample configuration items (servers, applications, contacts) and

default settings for the service catalog in order to ensure a quick deployment of the application. Do not select this option if you are deploying a production instance of iTop and want a “clean” installation.



The screenshot shows the 'Miscellaneous Parameters' screen of the iTop installation wizard. It features three main sections: 'Default Language' with a dropdown menu set to 'English (English)'; 'Application URL' with a text input field containing 'http://localhost:8080/2.0-beta/' and a note about changing the path; and 'Sample Data' with two radio button options: 'I am installing a demo or test instance, populate the database with some demo data.' (selected) and 'I am installing a production instance, create an empty database to start from.' At the bottom are '<< Back' and 'Next >>' buttons.

In case you have multiple virtual hosts pointing to iTop or if you are using a test system with a dynamic IP address (DHCP) and you want to connect anyway from another system, the “address” of the iTop server cannot be fixed at installation time. In this case you can use the placeholder `$_SERVER_NAME_` in the URL parameter. For example you can type: `http://$_SERVER_NAME_/itop` as the URL to access the application. At runtime this value will be substituted by the value of the PHP variable `SERVER_NAME` for the current connection.

The following steps of the installation wizard will help you tailor the “data model” of iTop in order to best suit your needs.

## Configuring the CMDB

The CMDB is made of 5 modules:

- The base/core module which is always installed. This module provides the basic objects that must always be present in iTop: Organizations, Locations, Contacts and a few common CIs like Servers, Groups, Applications, etc...
- The Datacenter Management module provides configurations items for managing the logistics of a data center: racks, enclosures, power (PDUs, power sources...). You probably don't need this module if you don't manage a data center.
- The End-User Devices modules defines the devices related to end-users: phones, PCs, tablets... if you don't plan to manage such devices in iTop, you may uncheck this option.

- The Storage module defines configuration items related to the management of storage devices: SAN switches, NAS, Tape libraries and tapes, etc... It is optional.
- The Virtualization module defines Virtual Machines, hypervisors, farms, etc. You may uncheck it if you don't plan to manage virtual devices in iTop.



The screenshot shows the 'iTop Configuration Management options' window. It features a header with the iTop logo and the title. Below the header is a graphic of three colored cubes (red, blue, green) and a text box explaining that the options below allow configuring the type of elements to be managed inside iTop. A list of five options is presented, each with a checked checkbox and a brief description: 'Configuration Management Core' (mandatory base objects), 'Data Center Devices' (Racks, Enclosures, PDUs), 'End-User Devices' (PCs, Phones, Tablets), 'Storage Devices' (NAS, SAN Switches, Tape Libraries and Tapes), and 'Virtualization' (Hypervisors, Virtual Machines and Farms). At the bottom, there are two buttons: '<< Back' and 'Next >>'.

**iTop Configuration Management options**

The options below allow you to configure the type of elements that are to be managed inside iTop.

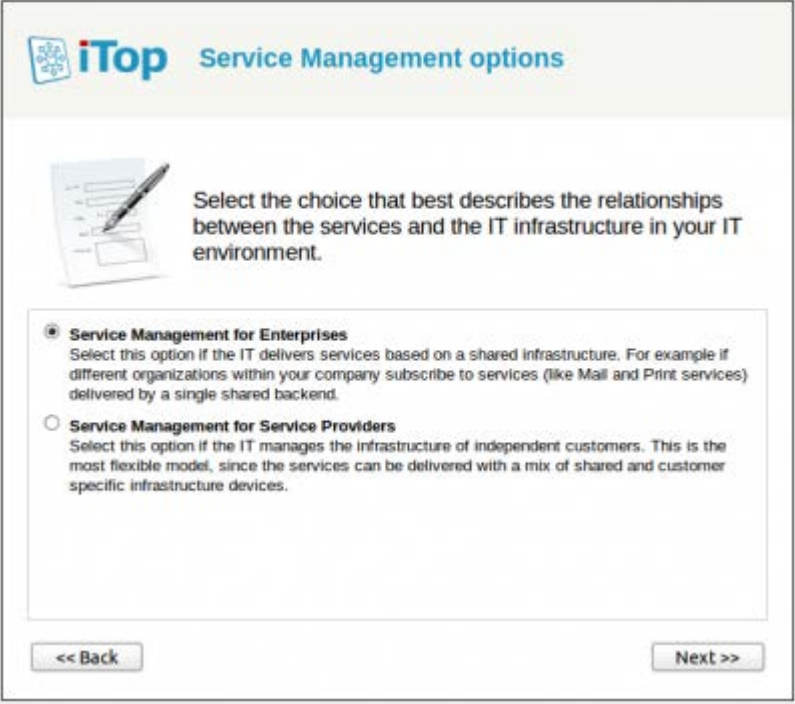
- ☒ **Configuration Management Core**  
All the base objects that are mandatory in the iTop CMDB: Organizations, Locations, Teams, Persons, etc.
- ☒ **Data Center Devices**  
Manage Data Center devices such as Racks, Enclosures, PDUs, etc.
- ☒ **End-User Devices**  
Manage devices related to end-users: PCs, Phones, Tablets, etc.
- ☒ **Storage Devices**  
Manage storage devices such as NAS, SAN Switches, Tape Libraries and Tapes, etc.
- ☒ **Virtualization**  
Manage Hypervisors, Virtual Machines and Farms.

<< Back      Next >>

## Configuring Service Management

Service management comes in two flavors in iTop. Choose the option that best represents your situation:

- Service Management for Enterprises is best suited for describing a situation where the same infrastructure is used to deliver services to a set of customers. Customers can be internal entities of a common organization or external customers.
- Service Management for Service Providers is best suited if the infrastructure used for delivering/supporting a customer is specific to each customer with little overlap (sharing) between customers.



The image shows a screenshot of the iTop Service Management options configuration screen. At the top left is the iTop logo, followed by the title "Service Management options". Below the title is an icon of a document with a pen. To the right of the icon is the instruction: "Select the choice that best describes the relationships between the services and the IT infrastructure in your IT environment." Below this instruction is a list of two radio button options. The first option, "Service Management for Enterprises", is selected. The second option is "Service Management for Service Providers". At the bottom of the screen are two buttons: "<< Back" on the left and "Next >>" on the right.

**iTop Service Management options**

Select the choice that best describes the relationships between the services and the IT infrastructure in your IT environment.

- ☒ **Service Management for Enterprises**  
Select this option if the IT delivers services based on a shared infrastructure. For example if different organizations within your company subscribe to services (like Mail and Print services) delivered by a single shared backend.
- ☐ **Service Management for Service Providers**  
Select this option if the IT manages the infrastructure of independent customers. This is the most flexible model, since the services can be delivered with a mix of shared and customer specific infrastructure devices.

<< Back      Next >>

## Configuring Tickets Management

If you plan to manage tickets in iTop, there are basically two options:

- Simple tickets management is best suited for implementing simple processes where little difference is made between the incidents and user requests. All tickets share the same life cycle and you can change the characterization of a ticket at any time. This choice is probably appropriate if you plan to let end-users create tickets directly through the portal either to place service requests or to report incidents.
- ITIL tickets is best suited for implementing strict ITIL compliant processes. A clear distinction is made between User Request and Incident tickets. (You may choose to deploy only one of these processes or both)

Of course you can also select not to manage such tickets in iTop at all. In this case, select the third option.

A screenshot of the iTop 'Tickets Management options' configuration window. The window has a light green header with the iTop logo and title. Below the header, there is a red circular icon with a white exclamation mark and the text 'Select the type of tickets you want to use in order to respond to user requests and incidents.' The main content area contains three radio button options. The first option, 'Simple Ticket Management', is selected and includes the description 'Select this option to use one single type of tickets for all kind of requests.' The second option, 'ITIL Compliant Tickets Management', is unselected and includes the description 'Select this option to have different types of ticket for managing user requests and incidents. Each type of ticket has a specific life cycle and specific fields'. Under this option, there are two checkboxes: 'User Request Management' (unselected) with the sub-description 'Manage User Request tickets in iTop', and 'Incident Management' (unselected) with the sub-description 'Manage Incidents tickets in iTop'. The third option, 'No Tickets Management', is unselected and includes the description 'Don't manage incidents or user requests in iTop'. At the bottom of the window, there are two buttons: '<< Back' on the left and 'Next >>' on the right.

**iTop Tickets Management options**

Select the type of tickets you want to use in order to respond to user requests and incidents.

☒ **Simple Ticket Management**  
Select this option to use one single type of tickets for all kind of requests.

☐ **ITIL Compliant Tickets Management**  
Select this option to have different types of ticket for managing user requests and incidents. Each type of ticket has a specific life cycle and specific fields

☐ **User Request Management**  
Manage User Request tickets in iTop

☐ **Incident Management**  
Manage Incidents tickets in iTop

☐ **No Tickets Management**  
Don't manage incidents or user requests in iTop


<< Back      Next >>


## Configuring Change Management

If you plan to implement Change Management in iTop there are also two options:

- Simple change management provides one single type of all purposes Change Request tickets.
- ITIL Change Management provides the complete set of ITIL compliant change processes with Normal, Routine and Emergency Change tickets.

You can also completely deactivate the Change Management in iTop by selecting the third choice.

 **Change Management options**

 Select the type of tickets you want to use in order to manage changes to the IT infrastructure.

☒ **Simple Change Management**  
Select this option to use one type of ticket for all kind of changes.

☐ **ITIL Change Management**  
Select this option to use Normal/Routine/Emergency change tickets.


☐ **No Change Management**  
Don't manage changes in iTop


<< Back

Next >>

## Additional ITIL processes

In addition to change and incident management, two other type of ITIL compliant processes can be deployed by selecting the corresponding type of tickets: Problem Management and Known Error management (which also includes FAQs).

 **Additional ITIL tickets**

 Pick from the list below the additional ITIL processes that are to be implemented in iTop.

☐ **Known Errors Management**  
Select this option to track "Known Errors" and FAQs in iTop.

☐ **Problem Management**  
Select this option track "Problems" in iTop.

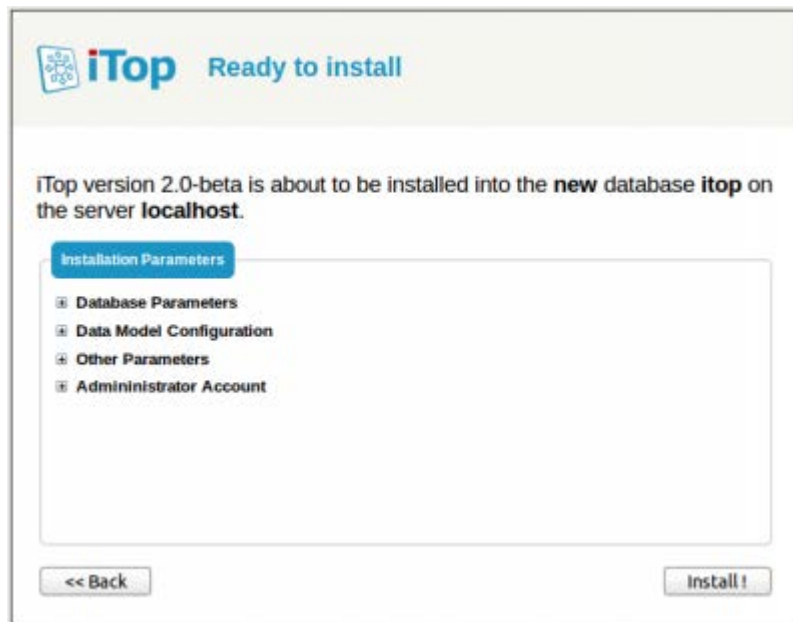
<< Back

Next >>

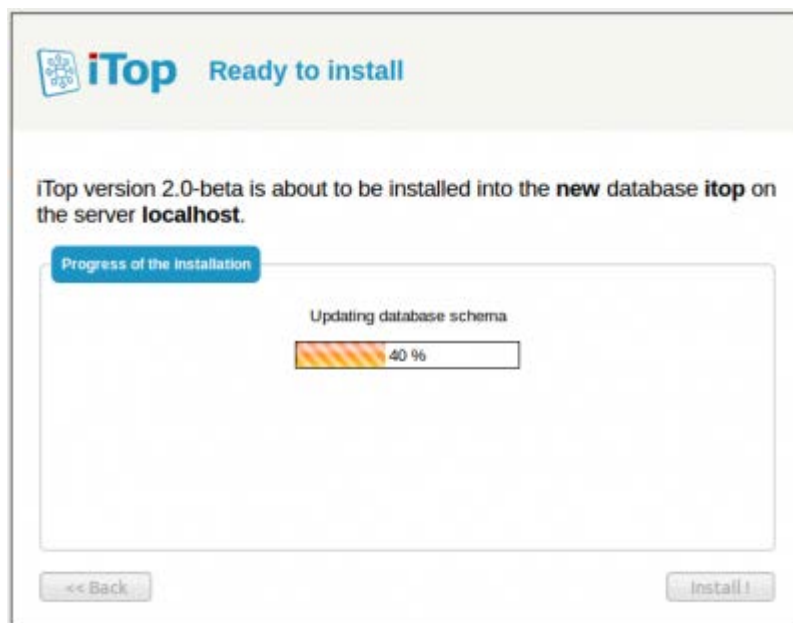
## Launching the installation

Once all the choices are made, a summary is displayed. You can drill down to check your choices by clicking on the “plus” button for each section.

To start the actual installation of iTop, click on the “Install !” button.

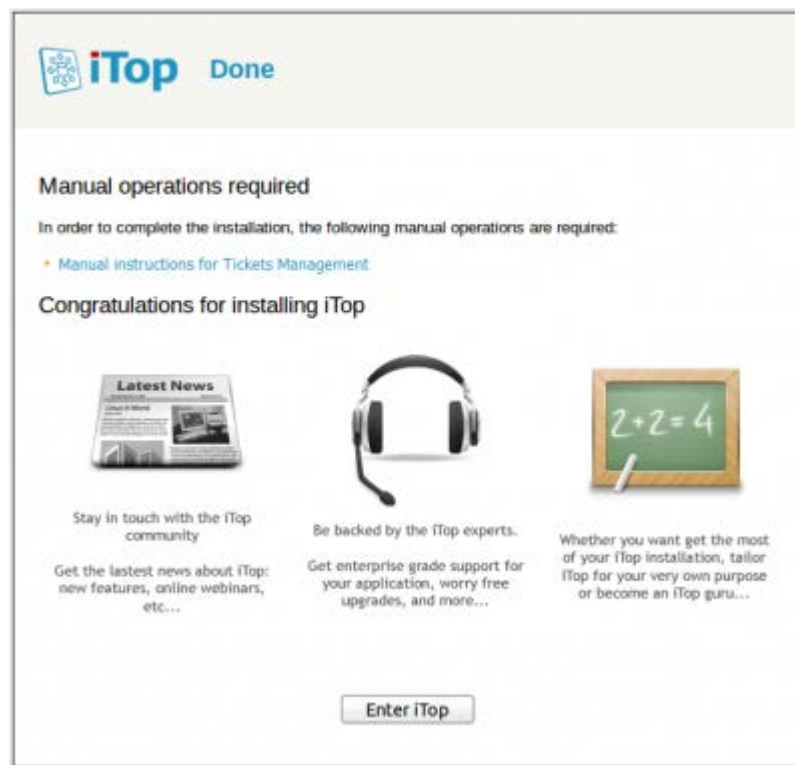


The progress of the installation is displayed with a moving progress bar.



Once the installation is done, you can: Review the instructions about setting up the background tasks, by clicking the link “Manual instructions for Tickets Management”. Or enter iTop immediately by clicking on the “Enter iTop” button.

Instructions to setup the background tasks are also available and potentially more up to date [here](#)



Congratulation, you' ve successfully installed iTop!

## How to upgrade

Extract the content of the “web” directory of the iTop archive into a new directory on your web server. Point your web browser to the location where you extracted the files and follow the on-screen instructions. At step 2, select “Upgrade an existing instance”.

If mysqldump is executable on your web server (must be in the path), then the setup program will offer you to perform a full backup of the iTop database (including the configuration file).

## Automatic upgrade from 1.x to 2.0

The automatic upgrade detects the installed version of iTop. Depending on this version several things happen:

If the version is 1.x then the setup checks if the version was modified compared to the signature of the files of the original iTop package.

If no modification is detected, the installation program will upgrade the data model to the version 1.3 bundled inside the iTop 2.0 package. This migrates the data automatically. Once the application is upgraded you will benefit from the new features (edition of dashboards, customizable lists, etc.) but retaining a data model similar to the 1.x version.

If modifications are detected, you can either discard them (and the upgrade will occur as in the case listed above) or preserve them. If you select to preserve the modifications, the setup program will copy the directories “modules” and “portal” from the previous installation and use the content of these folders in the new installation of iTop. Once upgraded, the edition of dashboards will not be possible since they require a different type of menus. Customizable lists and other user interface enhancements will work.

## **I have installed iTop 1.x out of the box what will happen when I upgrade?**

The data model will be upgraded to the version 1.3 and the data will be migrated automatically. All features of iTop are operational, but the data model is different from a version 2.0 installed “from scratch”.

## **I have customized iTop 1.x, what will happen to my customizations?**

If you supply the proper path to the location where iTop was installed, the setup should be capable of detecting these modifications and will preserve them. Depending on the type of customizations you made, make sure that you test your customizations before moving to production with the new versions. Some internals of iTop changed in 2.0, and this may affect your customization.

The edition of dashboards will not be possible since they require a different type of menus that were not part of the 1.x data model. Customizable lists and other user interface enhancements will work.

## I want to benefit from the 2.0 data model, how can I upgrade?

If you want to use the new data model, you have to install a new instance of iTop and export the data from your old instance, then re-import them in the new instance (some transformations may be needed for some classes).

## iTop 1.x to 2.0 migration instructions

### Data migration versus iTop upgrade

iTop 2.0 ships with two data-models:

- The 1.3 data model which is fully compatible with iTop 1.x. The only differences are a few bug fixes and the support of user-editable dashboards.
- The 2.0 data model which uses quite different concepts and was designed to be cleaner and more modular, but not necessarily aligned with the 1.x data model.

When upgrading the application using the setup process of iTop, the data-model of a previous (non-customized) 1.x instance is automatically migrated to the 1.3 data-model.

The “core” engine of iTop is upgraded to the 2.0 codebase and provides all the enhancements and bug fixes of the new version (user configurable lists and dashboards, shortcuts, etc.)

A “data migration”, on the other hand, consists in transforming the data stored in the database using the iTop 1.x data-model into the new 2.0 data model. Since the two models are significantly different, this process cannot be completely automated.

### Do I need to migrate my data to the 2.0 data-model?

If you are happy with the data model of iTop 1.2 and don't plan to document pieces of your IT infrastructure that are only covered by the 2.0 data-model, you don't need to “migrate” your data to 2.0.

You need to *migrate your data only if* you want to use (some part of) the new data-model of iTop 2.0. In all other cases you can rely on the automatic upgrade.

## User configurable dashboards and customized 1.x data-models

This is a special case for those who customized their 1.x data model and want to benefit from user editable dashboards without migrating to the 2.0 data model.

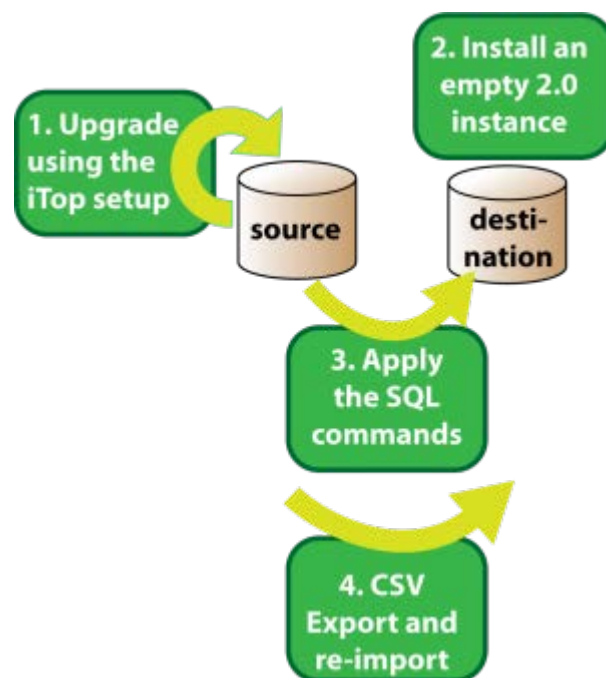
When upgrading iTop, the setup compares the signature of all the data model files of the previous instance (located in the “modules” directory) with the signature of a corresponding 1.x genuine version. If any modification is detected in one of these files, the end-user is prompted whether or not to preserve the modifications. If the user answers to keep the modifications, then the whole “modules” folder will be used as the data-model definition. Otherwise a standard 1.3 model is used.

Since the User Editable dashboards are a special kind of *Menu Node* they are part of the data-model definition. You can benefit from the new feature by replacing the *TemplateMenuNode* instances by *DashboardMenuNodes* in your customized 1.x data-model.

- On a development machine, install your customized 1.x iTop instance
- Use the standard upgrade procedure to upgrade it with the 2.0 iTop package
- When prompted to keep or discard your modifications, tell the setup to keep them
- Install the 2.0 toolkit on this new instance of iTop
- Edit the data model to add a new “empty” *DashboardMenuNode* menu item. Run the toolkit to apply your modifications
- Use the interactive editor dashboard editor to design your dashboard, as any user would be when customizing her/his own dashboard
- Save the customized dashboard using the menu “Export to file...”
- Copy/paste the resulting XML in the definition of the *DashboardMenuNode* in your new data model.
- Use the toolkit to apply the modifications.
- Connect to iTop and select “Revert to Original Version...” to check that the customization is now part of your data model.
- Remove the un-needed *TemplateMenuNode* from your data model
- Upgrade your production instance with newly customized data model

## Migration scenario overview

Be aware that the migration scenario supports neither the Tickets nor the Attachments yet!



If you want to migrate your 1.x iTOP instance to the new 2.0 data model, here is an overview of the process:

- Upgrade your “old” iTOP instance to 2.0. This will upgrade the data-model schema to 1.3 and bring to your iTOP instance all the benefits of the code enhancements and bug fixes (especially in terms of data export capabilities).
- On the same database server, but in a different database, install a brand new instance of iTOP that will receive your migrated data with the desired combination of modules from the 2.0 data-model. Don’t install the “Demo” data on this instance.
- Use the SQL script (see below) to transfer the Organizations, Documents, Contacts and User records from the old instance to the new instance. The SQL script will also create all the new Typology for you.
- For the remaining classes: Servers, PCs, Network Devices, Services, Contracts, etc... use the CSV export and CSV import features to transfer the data in the order explained below.

## Common migration pitfalls

Why is there no simple mapping between iTOP 1.2.1 and iTOP 2.0?

There are several cases where the two data models (1.x and 2.0) are significantly different and where a simple automated mapping is not possible. The major cases where such a manual intervention is required are listed below:

## Split of classes

Some concepts were absent from the 1.x data model. For example in iTop 1.2.1 there was no such thing as a “Virtual Machine”. In many cases the Virtual Machines were just documented using the “Server” object. Now that the “Virtual Machine” object exists in iTop 2.0, migrating from the old data-model to the new one requires distinguishing between servers and virtual machines. The way to perform this “split” depends on the data you entered in iTop. You may be fortunate enough to have used a strict naming convention that distinguishes servers from VMs, in which case a simple sort/filter in Excel can do the trick. In other cases you may need to review the data one by one (either in Excel or by creating two groups in iTop for instance).

## Typology

Some of the fields that were either closed enumerated lists or plain-text fields have been turned into configurable enumeration, grouped together under the “Typology” umbrella.

Example: Brand and Model.

The transformation of such fields is handled by the SQL script provided below. You may still need to review and cleanup the data for near duplicates like “Hewlett Packard”, “Hewlett-Packard” or, even worse, “HP”.

## Limitations / no more supported features

A number of features were removed from the 2.0 data-model: either because there are other (better?) ways of documenting the same kind of information or because they were useless and too confusing in most cases. In such case, migrating to the standard 2.0 data-model will result in a loss of data.

Examples:

- A Team can no longer be a member of another Team.
- Locations are no longer in a hierarchy (there is no parent location anymore)

## Duplicate names

If you have several objects of the same class that have the same name, be aware that the re-import of other objects that point to such “duplicates” can be challenging.

## Different structure

The structure for documenting some information changed with the 2.0 data-model. In some cases there is no simple one to one mapping between the elements in 1.x and the elements in 2.x. for example, in 1.x the network connections were documented by connecting interfaces together. Since this was confusing (and hard to document) for most of the users, a simpler, direct relation between the devices was introduced in the 2.0 data-model. It is however possible to transfer the network topology between both models, but this requires some non-trivial processing of the data.

## Order of operations

Once your two instances are ready for the migration, proceed as follows:

- Download the SQL script below and use find/replace in your favorite text editor to adjust the names of the source and destination databases. The source database is named `itopl1x` in the SQL below and the destination database is named `itop20`.

### [1x to-20 migration.sql](#)

```
/**
 * Organization
 */
TRUNCATE TABLE itop20.organization;
INSERT INTO itop20.organization (id, name, code, STATUS, parent_id,
parent_id_left, parent_id_right)
  SELECT id, name, code, STATUS, parent_id, parent_id_left,
parent_id_right FROM itopl1x.organization;

/**
 * Typology
 */
TRUNCATE TABLE itop20.typology;

/**
```

```

    * Typology - DocumentType
*/
INSERT INTO itop20.typology (name, finalclass)
    SELECT DISTINCT TYPE, 'DocumentType' FROM itopl1x.document WHERE
TYPE IS NOT NULL;

TRUNCATE TABLE itop20.documenttype;
INSERT INTO itop20.documenttype (id)
    SELECT id FROM itop20.typology WHERE finalclass = 'DocumentType';

/**
 * Document
*/
TRUNCATE TABLE itop20.document;
/**
 * Document - DocumentFile
*/
INSERT INTO itop20.document (id, name, org_id, documenttype_id,
description, STATUS, finalclass)
SELECT
    SOURCE.id,
    SOURCE.name,
    SOURCE.org_id,
    typo.id,
    SOURCE.description,
    SOURCE.STATUS,
    'DocumentFile'
FROM
    itopl1x.document AS SOURCE
    LEFT JOIN
        itop20.typology AS typo ON SOURCE.TYPE = typo.name
WHERE SOURCE.finalclass = 'FileDoc';

TRUNCATE TABLE itop20.documentfile;
INSERT INTO itop20.documentfile (id, file_data, file_mimetype,
file_filename)
SELECT id, contents_data, contents_mimetype, contents_filename
FROM itopl1x.filedoc;

/**
 * Document - DocumentNote
*/
INSERT INTO itop20.document (id, name, org_id, documenttype_id,
description, STATUS, finalclass)

```

```

SELECT
    SOURCE.id,
    SOURCE.name,
    SOURCE.org_id,
    typo.id,
    SOURCE.description,
    SOURCE.STATUS,
    'DocumentNote'
FROM
    itop1x.document AS SOURCE
    LEFT JOIN
        itop20.typology AS typo ON SOURCE.TYPE = typo.name
WHERE SOURCE.finalclass = 'Note';

TRUNCATE TABLE itop20.documentnote;
INSERT INTO itop20.documentnote (id, text)
SELECT id, note FROM itop1x.note;

/**
 * Document - DocumentWeb
 */
INSERT INTO itop20.document (id, name, org_id, documenttype_id,
description, STATUS, finalclass)
SELECT
    SOURCE.id,
    SOURCE.name,
    SOURCE.org_id,
    typo.id,
    SOURCE.description,
    SOURCE.STATUS,
    'DocumentWeb'
FROM
    itop1x.document AS SOURCE
    LEFT JOIN
        itop20.typology AS typo ON SOURCE.TYPE = typo.name
WHERE SOURCE.finalclass = 'WebDoc';

TRUNCATE TABLE itop20.documentweb;
INSERT INTO itop20.documentweb (id, url)
SELECT id, url FROM itop1x.externaldoc;

/**
 * Location
 */

```

```
TRUNCATE TABLE itop20.location;
INSERT INTO itop20.location (name, STATUS, org_id, address,
postal_code, city, country)
SELECT name, STATUS, org_id, address, postal_code, city, country
FROM itoplx.location;
```

```
/**
```

```
 * Contact
```

```
*/
```

```
TRUNCATE TABLE itop20.contact;
```

```
/**
```

```
 * Contact - Person
```

```
*/
```

```
INSERT INTO itop20.contact (id, name, STATUS, org_id, email, phone,
finalclass)
```

```
SELECT
```

```
    id, name, STATUS, org_id, email, phone, 'Person'
```

```
FROM
```

```
    itoplx.contact
```

```
WHERE finalclass = 'Person';
```

```
TRUNCATE TABLE itop20.person;
```

```
INSERT INTO itop20.person (id, first_name, employee_number,
location_id)
```

```
SELECT p.id, p.first_name, p.employee_id, c.location_id FROM
itoplx.person AS p JOIN itoplx.contact AS c ON c.id = p.id;
```

```
/**
```

```
 * Contact - Team
```

```
*/
```

```
INSERT INTO itop20.contact (id, name, STATUS, org_id, email, phone,
finalclass)
```

```
SELECT
```

```
    id, name, STATUS, org_id, email, phone, 'Team'
```

```
FROM
```

```
    itoplx.contact
```

```
WHERE finalclass = 'Team';
```

```
TRUNCATE TABLE itop20.team;
```

```
INSERT INTO itop20.team (id)
```

```
SELECT id FROM itoplx.team;
```

```
/**
```

```


    * Person to Team
*/
INSERT INTO itop20.typology (name, finalclass)
    SELECT DISTINCT ROLE, 'ContactType' FROM itop1x.lnkteamtocontact
WHERE ROLE !='';

TRUNCATE TABLE itop20.contacttype;
INSERT INTO itop20.contacttype (id)
    SELECT id FROM itop20.typology WHERE finalclass = 'ContactType';

TRUNCATE TABLE itop20.lnkpersontoteam;
INSERT INTO itop20.lnkpersontoteam (team_id, person_id, role_id)
    SELECT l.team_id, l.contact_id, t.id
    FROM itop1x.lnkteamtocontact AS l LEFT JOIN itop20.typology AS t
ON l.ROLE = t.name;

/**
    * User management
*/
TRUNCATE TABLE itop20.priv_user;
INSERT INTO itop20.priv_user SELECT * FROM itop1x.priv_user;
TRUNCATE TABLE itop20.priv_internaluser;
INSERT INTO itop20.priv_internaluser SELECT * FROM
itop1x.priv_internaluser;
TRUNCATE TABLE itop20.priv_user_local;
INSERT INTO itop20.priv_user_local SELECT * FROM
itop1x.priv_user_local;
TRUNCATE TABLE itop20.priv_urp_userorg;
INSERT INTO itop20.priv_urp_userorg SELECT * FROM
itop1x.priv_urp_userorg;
TRUNCATE TABLE itop20.priv_urp_profiles;
INSERT INTO itop20.priv_urp_profiles SELECT * FROM
itop1x.priv_urp_profiles;
TRUNCATE TABLE itop20.priv_urp_userprofile;
INSERT INTO itop20.priv_urp_userprofile SELECT * FROM
itop1x.priv_urp_userprofile;

```

- Connect to the MySQL server and run the SQL script. This will import the Organizations, Documents, Contacts, Locations, Users and will create all the Typology (not yet all ) objects in the new database.
- Use the CSV export (either interactive or scripted) to export the following data from the “old” instance.
- Use the CSV import (preferably scripted to avoid timeouts) on the new instance to import the data in the following order, and with the following field mapping:

## Organization

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
| name         | name              |
| code         | code              |
| status       | status            |
| parent_id    | parent_id         |

## Location

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
| name         | name              |
| status       | status            |
| org_id       | org_id            |
| address      | address           |
| postal_code  | postal_code       |
| city         | city              |
| country      | country           |

Locations are not longer part of a hierarchy.

## Person

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
| name         | name              |
| status       | status            |
| org_id       | org_id            |
| email        | email             |
| phone        | phone             |
| first_name   | first_name        |

employee\_id employee\_number  
location\_id location\_id

Team

Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
| name         | name              |
| status       | status            |
| org_id       | org_id            |
| email        | email             |
| phone        | phone             |

DocumentNote

Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
| name         | name              |
| org_id       | org_id            |
| description  | description       |
| status       | status            |
| note         | text              |

DocumentWeb

Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
| name         | name              |
| org_id       | org_id            |
| description  | description       |
| status       | status            |
| url          | url               |

## NetworkDevice

### Fields Mapping

| Source Field    | Destination Field    |
|-----------------|----------------------|
| name            | name                 |
| description     | description          |
| org_id          | org_id               |
| importance      | business_criticity   |
| serial_number   | serialnumber         |
| location_id     | location_id          |
| status          | status               |
| brand (*)       | brand_id             |
| model (*)       | model_id             |
| asset_ref       | asset_number         |
| management_ip   | managementip         |
| type (*)        | networkdevicetype_id |
| ios_version (*) | iosversion_id        |
| ram             | ram                  |

SNMP read/write, default\_gateway and location\_details are no longer part of the standard model.

## Server

### Fields Mapping

| Source Field  | Destination Field  |
|---------------|--------------------|
| name          | name               |
| description   | description        |
| org_id        | org_id             |
| importance    | business_criticity |
| serial_number | serialnumber       |
| location_id   | location_id        |
| status        | status             |
| brand (*)     | brand_id           |
| model (*)     | model_id           |

|                |              |
|----------------|--------------|
| asset_ref      | asset_number |
| management_ip  | managementip |
| os_family (*)  | osfamily_id  |
| os_version (*) | osversion_id |
| cpu            | cpu          |
| ram            | ram          |

## PC

### Fields Mapping

| Source Field   | Destination Field  |
|----------------|--------------------|
| name           | name               |
| org_id         | org_id             |
| importance     | business_criticity |
| serial_number  | serialnumber       |
| status         | status             |
| brand (*)      | brand_id           |
| model (*)      | model_id           |
| asset_ref      | asset_number       |
| os_family (*)  | osfamily_id        |
| os_version (*) | osversion_id       |
| cpu            | cpu                |
| ram            | ram                |

## Printer

### Fields Mapping

| Source Field  | Destination Field  |
|---------------|--------------------|
| name          | name               |
| description   | description        |
| org_id        | org_id             |
| importance    | business_criticity |
| serial_number | serialnumber       |
| location_id   | location_id        |

|           |              |
|-----------|--------------|
| status    | status       |
| brand (*) | brand_id     |
| model (*) | model_id     |
| asset_ref | asset_number |

## MobilePhone

### Fields Mapping

| Source Field  | Destination Field  |
|---------------|--------------------|
| name          | name               |
| org_id        | org_id             |
| importance    | business_criticity |
| serial_number | serialnumber       |
| status        | status             |
| brand (*)     | brand_id           |
| model (*)     | model_id           |
| asset_ref     | asset_number       |
| number        | phonenumber        |
| imei          | imei               |
| hw_pin        | hw_pin             |

## ApplicationSolution

### Fields Mapping

| Source Field | Destination Field  |
|--------------|--------------------|
| name         | name               |
| description  | description        |
| org_id       | org_id             |
| importance   | business_criticity |
| status       | status             |

## BusinessProcess

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
|--------------|-------------------|

|             |                    |
|-------------|--------------------|
| name        | name               |
| description | description        |
| org_id      | org_id             |
| importance  | business_criticity |
| status      | status             |

## DBServer

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
|--------------|-------------------|

|             |                    |
|-------------|--------------------|
| name        | name               |
| description | description        |
| org_id      | org_id             |
| importance  | business_criticity |
| device_id   | system_id          |
| software_id | software_id        |
| licence_id  | softwarelicence_id |
| status      | status             |

## OtherSoftware

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
|--------------|-------------------|

|             |                    |
|-------------|--------------------|
| name        | name               |
| description | description        |
| org_id      | org_id             |
| importance  | business_criticity |
| device_id   | system_id          |

|             |                    |
|-------------|--------------------|
| software_id | software_id        |
| licence_id  | softwarelicence_id |
| status      | status             |

## DatabaseSchema

### Fields Mapping

| Source Field          | Destination Field  |
|-----------------------|--------------------|
| name                  | name               |
| description           | description        |
| org_id                | org_id             |
| importance            | business_criticity |
| db_server_instance_id | dbserver_id        |

## SoftwareLicence

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
| name         | name              |
| description  | description       |
| org_id       | org_id            |
| importance   | usage_limite      |
| start_date   | start_date        |
| end_date     | end_date          |
| licence_key  | licence_key       |
| usage_limit  | usage_limit       |

## InkContactToFunctionalCI

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
| ci_id        | functionalci_id   |

contact\_id    contact\_id

No more role in 2.0 version

## InkDocumentToFunctionalCI

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
|--------------|-------------------|

|       |                 |
|-------|-----------------|
| ci_id | functionalci_id |
|-------|-----------------|

|             |             |
|-------------|-------------|
| document_id | document_id |
|-------------|-------------|

### Subnet

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
|--------------|-------------------|

|             |             |
|-------------|-------------|
| description | description |
|-------------|-------------|

|        |        |
|--------|--------|
| org_id | org_id |
|--------|--------|

|    |    |
|----|----|
| ip | ip |
|----|----|

|         |         |
|---------|---------|
| ip_mask | ip_mask |
|---------|---------|

## PhysicalInterface

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
|--------------|-------------------|

|      |      |
|------|------|
| name | name |
|------|------|

|            |           |
|------------|-----------|
| ip_address | ipaddress |
|------------|-----------|

|             |            |
|-------------|------------|
| mac_address | macaddress |
|-------------|------------|

|               |         |
|---------------|---------|
| serial_number | comment |
|---------------|---------|

|         |        |
|---------|--------|
| ip_mask | ipmask |
|---------|--------|

|       |       |
|-------|-------|
| speed | speed |
|-------|-------|

|           |                  |
|-----------|------------------|
| device_id | connectableci_id |
|-----------|------------------|

## InkConnectableCIToNetworkDevice

### Fields Mapping

| Source Field             | Destination Field |
|--------------------------|-------------------|
| source(if_dev)→device_id | networkdevice_id  |
| source(if_ci)→device_id  | connectableci_id  |
| source(if_dev)→name      | network_port      |
| source(if_ci)→name       | device_port       |
| source(if_dev)→link_type | connection_type   |

## InkApplicationSolutionToFunctionalCI

### Fields Mapping

| Source Field | Destination Field      |
|--------------|------------------------|
| solution_id  | applicationsolution_id |
| ci_id        | functionalci_id        |

## InkApplicationSolutionToBusinessProcess

### Fields Mapping

| Source Field | Destination Field      |
|--------------|------------------------|
| solution_id  | applicationsolution_id |
| process_id   | businessprocess_id     |

## InkPersonToTeam

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
| team_id      | team_id           |
| contact_id   | person_id         |

## Group

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
| name         | name              |
| status       | status            |
| org_id       | org_id            |
| description  | description       |
| type         | type              |
| parent_id    | parent_id         |

## InkGroupToCI

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
| group_id     | group_id          |
| ci_id        | ci_id             |
| reason       | reason            |

## CustomerContract

### Fields Mapping

| Source Field      | Destination Field |
|-------------------|-------------------|
| name              | name              |
| org_id            | org_id            |
| description       | description       |
| start_date        | start_date        |
| end_date          | end_date          |
| cost              | cost              |
| cost_currency     | cost_currency     |
| billing_frequency | billing_frequency |
| cost_unit         | cost_unit         |

provider\_id      provider\_id

## ProviderContract

### Fields Mapping

| Source Field      | Destination Field |
|-------------------|-------------------|
| name              | name              |
| org_id            | org_id            |
| description       | description       |
| start_date        | start_date        |
| end_date          | end_date          |
| cost              | cost              |
| cost_currency     | cost_currency     |
| billing_frequency | billing_frequency |
| cost_unit         | cost_unit         |
| provider_id       | provider_id       |
| sla               | sla               |
| coverage          | coverage          |

## InkContactToContract

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
| contact_id   | contact_id        |
| contract_id  | contract_id       |

## InkContractToDocument

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
| document_id  | document_id       |
| contract_id  | contract_id       |

## Service

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
| name         | name              |
| org_id       | org_id            |
| description  | description       |
| status       | status            |

## InkDocumentToService

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
| document_id  | document_id       |
| service_id   | service_id        |

## InkContactToService

### Fields Mapping

| Source Field                           | Destination Field |
|--|-------------------|
| contact_id                             | contact_id        |
| service_id                             | service_id        |
| Role is no more used in 2.0 data model |                   |

## ServiceSubcategory

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
| name         | name              |
| description  | description       |

|              |              |
|--------------|--------------|
| service_id   | service_id   |
| 'incident'   | request_type |
| 'production' | status       |

## SLA

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
|--------------|-------------------|

|      |        |
|------|--------|
| name | name   |
| 1    | org_id |

## SLT

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
|--------------|-------------------|

|                 |          |
|-----------------|----------|
| name            | name     |
| ticket_priority | priority |
| metric          | metric   |
| value           | value    |
| value_unit      | unit     |

## InkSLAToSLT

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
|--------------|-------------------|

|        |        |
|--------|--------|
| sla_id | sla_id |
| slt_id | slt_id |

## InkCustomerContractToService

### Fields Mapping

| Source Field            | Destination Field   |
|-------------------------|---------------------|
| source(lnk)→contract_id | customercontract_id |
| source(sla)→service_id  | service_id          |
| source(lnk)→sla_id      | sla_id              |

## InkCustomerContractToProviderContract

### Fields Mapping

| Source Field         | Destination Field   |
|----------------------|---------------------|
| customer_contract_id | customercontract_id |
| provider_contract_id | providercontract_id |

## Problem

### Fields Mapping

| Source Field          | Destination Field     |
|-----------------------|-----------------------|
| ref                   | ref                   |
| org_id                | org_id                |
| agent_id              | agent_id              |
| title                 | title                 |
| description           | description           |
| start_date            | start_date            |
| last_update           | last_update           |
| close_date            | close_date            |
| status                | status                |
| service_id            | service_id            |
| servicesubcategory_id | servicesubcategory_id |
| product               | product               |
| impact                | impact                |

|                   |                   |
|-------------------|-------------------|
| urgency           | urgency           |
| priority          | priority          |
| related_change_id | related_change_id |
| assignment_date   | assignment_date   |
| resolution_date   | resolution_date   |

## RoutineChange

### Fields Mapping

| Source Field        | Destination Field   |
|---------------------|---------------------|
| ref                 | ref                 |
| org_id              | org_id              |
| agent_id            | agent_id            |
| title               | title               |
| description         | description         |
| start_date          | start_date          |
| end_date            | end_date            |
| last_update         | last_update         |
| close_date          | close_date          |
| status              | status              |
| reason              | reason              |
| requestor_id        | requestor_id        |
| creation_date       | creation_date       |
| impact              | impact              |
| supervisor_group_id | supervisor_group_id |
| supervisor_id       | supervisor_id       |
| manager_group_id    | manager_group_id    |
| manager_id          | manager_id          |
| outage              | outage              |
| fallback            | fallback            |

## NormalChange

### Fields Mapping

| Source Field        | Destination Field   |
|---------------------|---------------------|
| ref                 | ref                 |
| org_id              | org_id              |
| agent_id            | agent_id            |
| title               | title               |
| description         | description         |
| start_date          | start_date          |
| end_date            | end_date            |
| last_update         | last_update         |
| close_date          | close_date          |
| status              | status              |
| reason              | reason              |
| requestor_id        | requestor_id        |
| creation_date       | creation_date       |
| impact              | impact              |
| supervisor_group_id | supervisor_group_id |
| supervisor_id       | supervisor_id       |
| manager_group_id    | manager_group_id    |
| manager_id          | manager_id          |
| outage              | outage              |
| fallback            | fallback            |
| approval_date       | approval_date       |
| approval_comment    | approval_comment    |
| acceptance_date     | acceptance_date     |
| acceptance_comment  | acceptance_comment  |

## EmergencyChange

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
|--------------|-------------------|

|                     |                     |
|---------------------|---------------------|
| ref                 | ref                 |
| org_id              | org_id              |
| agent_id            | agent_id            |
| title               | title               |
| description         | description         |
| start_date          | start_date          |
| end_date            | end_date            |
| last_update         | last_update         |
| close_date          | close_date          |
| status              | status              |
| reason              | reason              |
| requestor_id        | requestor_id        |
| creation_date       | creation_date       |
| impact              | impact              |
| supervisor_group_id | supervisor_group_id |
| supervisor_id       | supervisor_id       |
| manager_group_id    | manager_group_id    |
| manager_id          | manager_id          |
| outage              | outage              |
| fallback            | fallback            |
| approval_date       | approval_date       |
| approval_comment    | approval_comment    |

## InkContactToTicket

### Fields Mapping

#### Source Field Destination Field

|            |            |
|------------|------------|
| role       | role       |
| ticket_id  | ticket_id  |
| contact_id | contact_id |

## InkFunctionalCIToTicket

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
| impact       | impact            |
| ticket_id    | ticket_id         |
| ci_id        | functionalci_id   |

## KnownError

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
| name         | name              |
| org_id       | org_id            |
| problem_id   | problem_id        |
| symptom      | symptom           |
| root_cause   | root_cause        |
| workaround   | workaround        |
| solution     | solution          |
| error_code   | error_code        |
| domain       | domain            |
| vendor       | vendor            |
| model        | model             |
| version      | version           |

## InkErrorToFunctionalCI

### Fields Mapping

| Source Field | Destination Field |
|--------------|-------------------|
| error_id     | error_id          |
| infra_id     | functionalci_id   |
| reason       | reason            |

# InkDocumentToError

## Fields Mapping

**Source Field    Destination Field**

|           |             |
|-----------|-------------|
| error_id  | error_id    |
| doc_id    | document_id |
| link_type | link_type   |

# UserRequest

## Fields Mapping

| Source Field            | Destination Field       |
|-------------------------|-------------------------|
| ref                     | ref                     |
| org_id                  | org_id                  |
| caller_id               | caller_id               |
| agent_id                | agent_id                |
| title                   | title                   |
| description             | description             |
| start_date              | start_date              |
| last_update             | last_update             |
| close_date              | close_date              |
| status                  | status                  |
| request_type            | request_type            |
| impact                  | impact                  |
| priority                | priority                |
| urgency                 | urgency                 |
| service_id              | service_id              |
| servicesubcategory_id   | servicesubcategory_id   |
| assignment_date         | assignment_date         |
| resolution_date         | resolution_date         |
| tto_escalation_deadline | tto_escalation_deadline |
| ttr_escalation_deadline | ttr_escalation_deadline |
| resolution_code         | resolution_code         |

|                   |                   |
|-------------------|-------------------|
| solution          | solution          |
| user_satisfaction | user_satisfaction |

## PHP and MySQL settings

iTop is capable of uploading and storing documents (i.e files) as attachments to various objects (Tickets, CIs...). These documents are stored as binary blobs in the iTop database. In order to safely upload and store documents, several settings must be adjusted consistently across PHP and MySQL.

In PHP, several variables govern the upload of files:

|                     |  |
|---------------------|--|
| file_uploads        | Set to 1 to allow file upload, to zero to prevent all file uploads.<br><br>The temporary location (on the server) where the uploaded files will be stored. Make sure that this parameter points to a location that is accessible (and writable) by the process running the web server (or by the end users in case of IIS with the Windows built-in authentication) and that there is enough space left. |
| upload_tmp_dir      |  |
| upload_max_filesize | The maximum size allowed for an uploaded file. The value is expressed in bytes. You can use units like K for kilobytes (=1024 bytes), M for megabytes and G for gigabytes. Example: 4M stands for 4 megabytes.   |
| max_file_uploads    | The maximum number of files that can be uploaded simultaneously in a single web page. iTop should normally upload only one file at a time. You can safely use the default value, which is 20.  |
| post_max_size       | The maximum amount of data that can be sent to the server via a POST request. This value MUST BE bigger than upload_max_filesize, since the same request will contain some more information (the title of the document, an operation code...). So it's better to put a bigger value here. For example, if upload_max_filesize is 4M, then put 5M for post_max_size.                                      |
| memory_limit        | After being uploaded on the server, the file will be read in memory before being stored in the database. Therefore make sure that memory_limit (if enabled) is far bigger than upload_max_filesize.  |
| max_input_time      | This value defines the maximum time allowed for the server to read its input. This includes the time spent uploading the files. The default of 60 seconds may be exceeded for uploading big files over slow connections.   |

The uploaded files are stored into the MySQL database, each file in one query. Therefore the maximum size allowed for a query MUST BE BIGGER than the maximum size of the uploaded file. This is configured via the variable

`max_allowed_packet` in the `my.cnf` configuration file (on the MySQL server).

It is good practice to have the following relation between the various settings:

`upload_max_filesize < post_max_size < max_allowed_packet < memory_limit`

*php.ini*

*php.ini*

*my.cnf*

*php.ini*

## iTop and Suhosin

**Suhosin** is an extra security layer which applies a number of security checks on top of PHP. If your web server is not public on the Internet and if you don't plan to deliver the iTop application to millions of simultaneous users, you may consider relaxing a bit some of the Suhosin limits in favor of more functionalities.

Suhosin is installed and activated by default on some Linux distributions (for instance Debian & Ubuntu).

Here is a list of the known issues and the related Suhosin configuration parameter:

| Parameter                                 | Value  | Symptom  | Suggested fix                                   |
|---|--------|--|---|
| <code>suhosin.post.max_vars</code>        | 1000   | Could not delete/modify more than 997 objects at a time.   | Increase the limit in the Suhosin configuration |
| <code>suhosin.get.max_value_length</code> | < 1024 | Unpredictable effects. iTop will detect that at the setup. | Increase the limit in the Suhosin configuration |

More information about Suhosin: <http://www.hardened-php.net/>

## Changing iTop Settings

All iTop configuration settings are stored in the file `config-itop.php` in the `conf/production` directory inside the iTop installation directory. Changing a parameter has an immediate effect on iTop, there is no need to restart the web server, just refresh the iTop web page in your browser to take into account the new parameter(s).

Refer to the chapter “[Configuration parameters](#)” for the full list of configuration parameters.

Hereafter are described a few usages that can be made of the configuration parameters.

## Ready-only mode

It is sometimes desirable (while performing some maintenance tasks for example) to make the iTop application read-only. Since version 1.0.2, two parameters can be used to control:

- whether or not the application is read-only (and for who)
- which message is displayed when the application is read-only.

These 2 parameters are: **access\_mode** and **access\_message**.

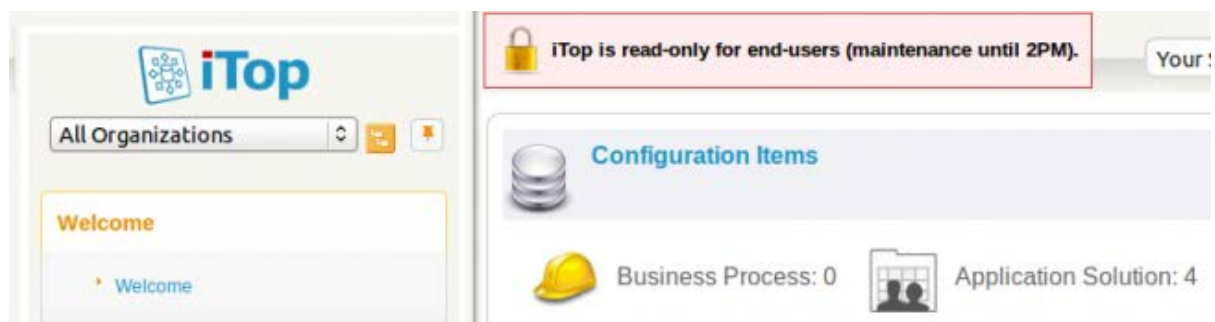
The parameter **access\_mode** can take one of the following values:

| Access_mode value  | Actual value | Effect  |
|--------------------|--------------|---|
| ACCESS_READONLY    | 0            | The application is read-only for all users. The users can browse the application but nothing will be written to the MySQL database. |
| ACCESS_ADMIN_WRITE | 2            | Only administrator users can write into the database. The application is in read-only mode for all other users.                     |
| ACCESS_FULL        | 3            | All users can write into the database. This is the default mode.  |

Example:

```
'access_mode' => ACCESS_ADMIN_WRITE,  
'access_message' => ' (maintenance until 2PM).',
```

This results in the following display in iTop:



Starting 2.0, the customer portal takes that mode into account and displays the exact same message.

## Accessing iTop from different URLs

Under some circumstances, it may be helpful to access the same instance of iTop from different URLs. (For example from the intranet for users within the company and from internet for customers).

One possibility is to install two iTop instances pointing to the same database. This allows to have a complete control over all parameters, but increases the maintenance effort to keep the two instances in sync (same version of iTop, consistent parameters...).

Another possibility is to put some code in the configuration file to adjust the settings depending on some server parameter. Since the configuration file is a PHP file that is loaded (i.e. evaluated) on each 'page', the settings can be fully dynamic.

For example, imagine you have a reverse proxy that controls the access to iTop, and you want all Internet users to access iTop using HTTPS (secure) and intranet users to connect through http (faster). You can do the following:

- From internet :
  - Add a rule to redirect all HTTP traffic on HTTPS
  - Add a specific header in HTTPS virtualhost config :
  - `APACHE:RequestHeader set HTTPS On`
  - `NGINX:proxy_set_header HTTPS On;`
- From intranet :
  - Add a rule to redirect all HTTPS traffic on HTTP
- Set the following value in the config file for `app_root_url`:

```
'app_root_url' => (isset($_SERVER['HTTP_HTTPS']) &&
($_SERVER['HTTP_HTTPS'] == 'On')) ? 'https://itop.yourdomain/' :
'http://itop.yourdomain/';
```

Other combinations are possible: in PHP the `$_SERVER` and `$_COOKIE` variables are accessible.

## Related articles

[Authentication options](#)

# Using iTop

## Connecting to iTop

To access iTop, just point your web browser to the URL where iTop is installed. For example: `http://yourserver/<itop_alias>`

When prompted, to enter your login and password.

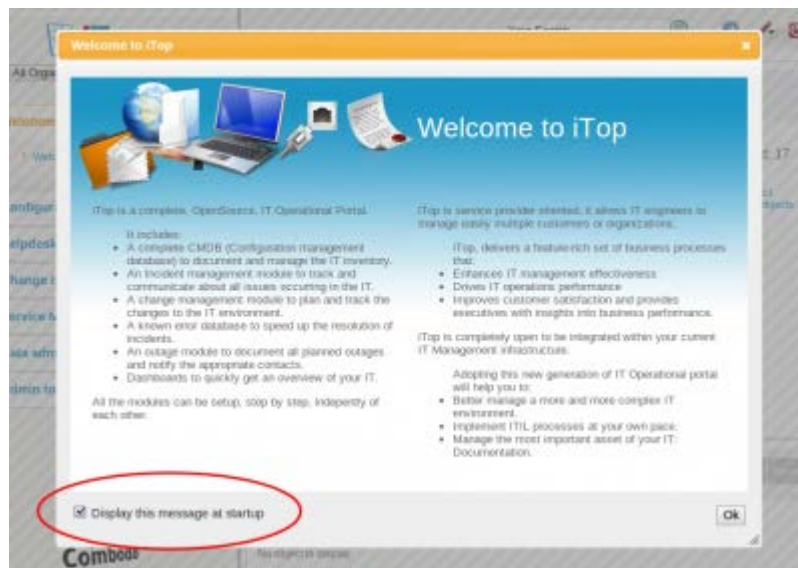
The image shows the iTop login interface. At the top, there is a logo consisting of a blue square with a white network diagram and the text "iTop" in blue. Below the logo, the text "Welcome to iTop!" is displayed in blue. Underneath, the instruction "Identify yourself before continuing" is shown in black. There are two input fields: "User Name:" and "Password:". Below these fields is a button labeled "Enter iTop".

iTop uses PHP session's handling mechanism to cache the user's authentication across pages. This means that all tabs and windows opened with the same web browser share the same authentication. The session is closed when the last tab/window of the browser is closed, or when expiring as specified by the PHP parameter [session.gc\\_maxlifetime](#)

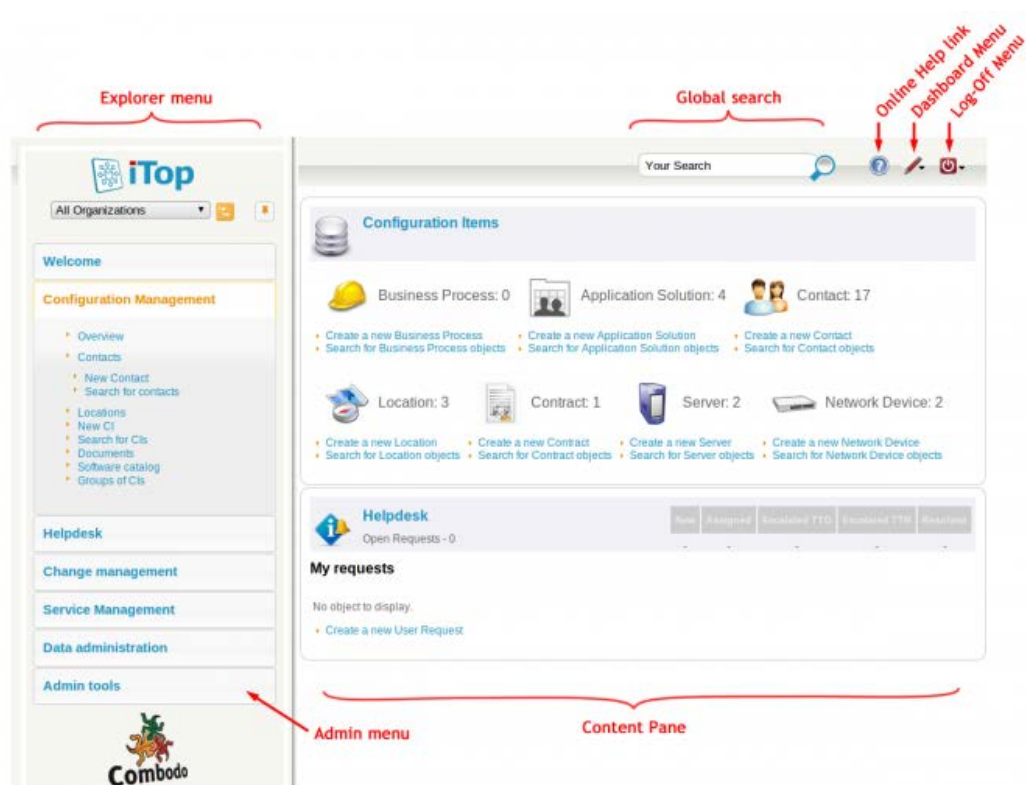
As iTop authentication is depending on the PHP session mechanism, changing any of the [PHP session settings](#) can lead to undesired effects such as users prompted to enter their password at each page. In particular, the iTop setup program does check that the file used by PHP can be written by the web server.

## Navigating in iTop

Once authenticated, the user accesses the main iTop page. The first time a user connects the Welcome to iTop popup is displayed. This popup can be removed for the next time by un-checking "Display this message at startup".



The main screen of iTop is the following:




The main page is divided in three major parts:

- The menu on the left (also called explorer menu) contains links to access the different modules (CMDB, Incidents, Changes, Services and contracts) of iTop. The exact list shown

depends on the modules installed and the rights of the current user. For example the menu “Admin tools” (at the bottom) is only visible to administrators.

- The content pane, on the right, displays the lists of items resulting from a search or the details of a given item.
- The top frame contains the global search, the dashboard menu (when the page shown is an editable dashboard) and the logoff menu, also used for managing user's preferences.

The URL of the online-help hyperlink is configurable through the `online_help` configuration parameter.

Refer to the document “iTop user guide” for details about how to use the application. 

## Managing User Accounts

iTop provides a user management module allowing administrators to assign users with one (or more) predefined profiles. The combination of profiles determines for each user the actions she/he is allowed to performed in iTop (viewing, creating/modifying or deleting which objects).

In the current version of iTop, the profiles are predefined; there is no user interface to modify them or to create new profiles.

## Viewing Profiles

Use the “Admin Tools / Profiles” menu to access the profiles and see their corresponding definitions as shown below:



The screenshot shows the 'Profiles' page in iTop. It features a table with two columns: 'Profile' and 'Description'. The 'Profile' column lists various roles, and the 'Description' column provides details for each. An 'Actions' button is visible in the top right corner of the table area.

| Profile               | Description   |
|-----------------------|---|
| Administrator         | Has the rights on everything (bypassing any control)  |
| Change Approver       | Person who could be impacted by some changes  |
| Change Implementor    | Person executing the changes  |
| Change Supervisor     | Person responsible for the overall change execution   |
| Configuration Manager | Person in charge of the documentation of the managed CIs  |
| Document author       | Any person who could contribute to documentation  |
| Portal user           | Has the rights to access to the user portal. People having this profile will not be allowed to access the standard application, they will be automatically redirected to the user portal. |
| Problem Manager       | Person analysing and solving the current problems   |
| Service Desk Agent    | Person in charge of creating incident reports   |
| Service Manager       | Person responsible for the service delivered to the [internal] customer   |
| Support Agent         | Person analysing and solving the current incidents  |

When you click on a given profile, the details of this profile are displayed.

| Profile: Configuration Manager        |      |                    |        |             |        |             |         |  |  |
|---------------------------------------|------|--------------------|--------|-------------|--------|-------------|---------|--|--|
| Actions                               |      |                    |        |             |        |             |         |  |  |
| Properties Users Grant Matrix History |      |                    |        |             |        |             |         |  |  |
| Class                                 | Read | Bulk Read (Export) | Modify | Bulk Modify | Delete | Bulk Delete | Stimuli |  |  |
| Organization                          | Yes  | Yes                | Yes    | Yes         | Yes    | NO          |         |  |  |
| Location                              | Yes  | Yes                | Yes    | Yes         | Yes    | NO          |         |  |  |
| Person                                | Yes  | Yes                | Yes    | Yes         | Yes    | NO          |         |  |  |
| Team                                  | Yes  | Yes                | Yes    | Yes         | Yes    | NO          |         |  |  |
| Team Members                          | Yes  | Yes                | Yes    | Yes         | Yes    | NO          |         |  |  |
| Web Document                          | Yes  | Yes                | Yes    | Yes         | Yes    | NO          |         |  |  |
| Note                                  | Yes  | Yes                | Yes    | Yes         | Yes    | NO          |         |  |  |
| Document (file)                       | Yes  | Yes                | Yes    | Yes         | Yes    | NO          |         |  |  |
| Licence                               | Yes  | Yes                | Yes    | Yes         | Yes    | NO          |         |  |  |
| Subnet                                | Yes  | Yes                | Yes    | Yes         | Yes    | NO          |         |  |  |
| Patch                                 | Yes  | Yes                | Yes    | Yes         | Yes    | NO          |         |  |  |
| Application                           | Yes  | Yes                | Yes    | Yes         | Yes    | NO          |         |  |  |
| Database                              | Yes  | Yes                | Yes    | Yes         | Yes    | NO          |         |  |  |
| Patch Usage                           | Yes  | Yes                | Yes    | Yes         | Yes    | NO          |         |  |  |
| DB Server Instance                    | Yes  | Yes                | Yes    | Yes         | Yes    | NO          |         |  |  |
| Application Instance                  | Yes  | Yes                | Yes    | Yes         | Yes    | NO          |         |  |  |
| Database Instance                     | Yes  | Yes                | Yes    | Yes         | Yes    | NO          |         |  |  |
| Group                                 | Yes  | Yes                | Yes    | Yes         | Yes    | NO          |         |  |  |
| Group / CI                            | Yes  | Yes                | Yes    | Yes         | Yes    | NO          |         |  |  |
| Application Solution                  | Yes  | Yes                | Yes    | Yes         | Yes    | NO          |         |  |  |
| Business Process                      | Yes  | Yes                | Yes    | Yes         | Yes    | NO          |         |  |  |
| Network Interface                     | Yes  | Yes                | Yes    | Yes         | Yes    | NO          |         |  |  |

- The tab “Users”, lists all users having this profile.
- The tab “Grant matrix” displays, for each class of objects, all the actions allowed for this profile.

## Default profiles

| Profile               | Description   |
|-----------------------|---|
| Administrator         | Has the rights on everything (bypassing any control)  |
| Change Approver       | Person who could be impacted by some changes.   |
| Change Implementor    | Person executing the changes.   |
| Change Supervisor     | Person responsible for the overall change execution.  |
| Configuration Manager | Person in charge of the documentation of the managed CIs.   |
| Document author       | Any person who could contribute to documentation.   |
| Portal user           | Has the rights to access to the user portal. People having this profile will not be allowed to access the standard application; they will be automatically redirected to the user portal. |
| Problem Manager       | Person analyzing and solving the current problems.  |
| Service Desk Agent    | Person in charge of creating incident reports.  |
| Service Manager       | Person responsible for the service delivered to the [internal] customer.  |

Support Agent    Person analyzing and solving the current incidents.

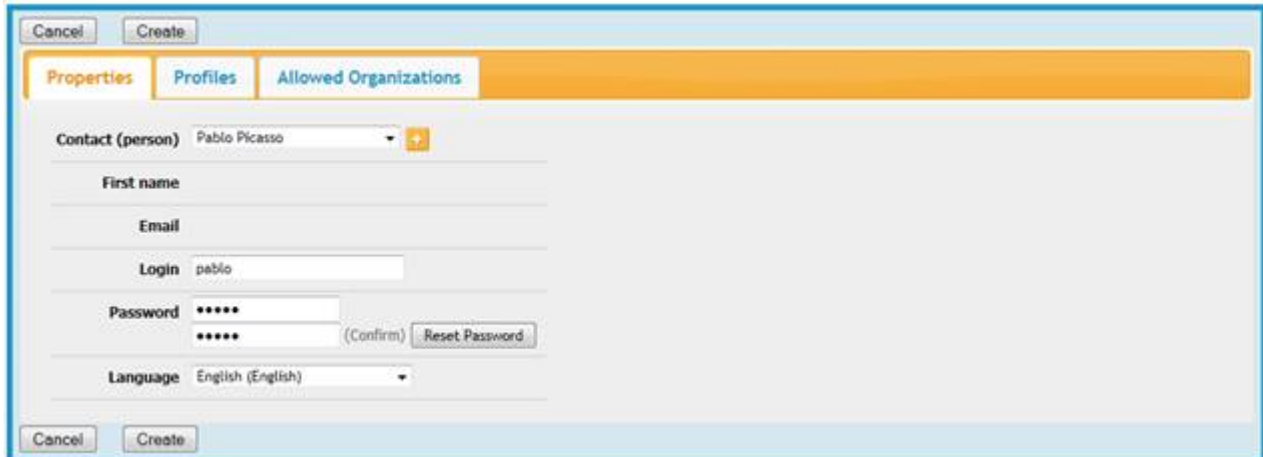
## Viewing User Accounts

The menu “User Accounts” under “Admin Tools” module, enables you to see all logins defined for your iTOP instance.



| User  | Type of account | First name    | Last name    | Login |
|-------|-----------------|---------------|--------------|-------|
| admin | ITop user       | My first name | My last name | admin |
| pablo | ITop user       | Pablo         | Picasso      | pablo |

When clicking on a user you get the following details:



Cancel Create

Properties Profiles Allowed Organizations

Contact (person) Pablo Picasso

First name

Email

Login pablo

Password \*\*\*\*\* (Confirm) \*\*\*\*\* Reset Password

Language English (English)

Cancel Create

A user account must be linked to a Person stored in the CMDB (See the [CMDB Module documentation](#)). Prior to creating a login, make sure that the user is documented as a Person in the CMDB.

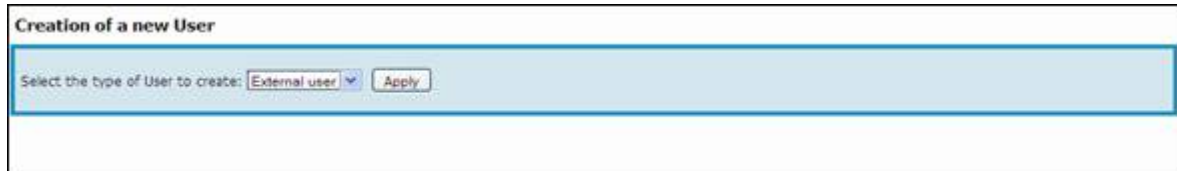
If no contact is defined for a login, then that login will suffer several limitations (list not exhaustive):

- Cannot receive email notifications. Example: a ticket has been created for customer x.
- Cannot be responsible for something. Example: the agent a ticket is assigned to.
- No access to the customer portal.

The tab “Profiles” list all profiles that are linked to this user. The tab “Grants matrix” display rights allowed for this user. It is the merge of all rights corresponding to associated profiles. The tab “Allowed Organizations” display list of organization this user is allowed to see.

## Creating a user

To create a new user you just have to click on “New” in action drop down list, from either user list or a given user detail. The following wizard then appears:



Creation of a new User

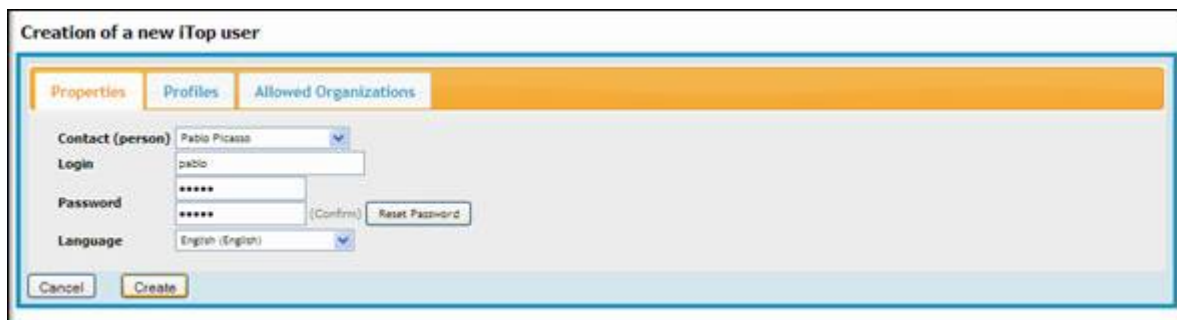
Select the type of User to create: External user Apply

Administrators can define different types of user accounts, depending on the desired type of authentication:

- iTop user accounts are internal to iTop. Their passwords are stored (encrypted) within the database of iTop. This type of account is useful for administrative users, for scripts and integration with other applications.
- LDAP user accounts have their authentication done by an external LDAP or Active Directory server.
- External user accounts have their authentication managed directly by the web server, for example when using an Apache .htaccess file or when using an external single-sign-on solution, like for example JASIG-CAS.

All the details about authentication in iTop are described in the chapter [User authentication options](#).

If you decide to create an iTop user, you have to type-in the password and to retype it a second time for confirmation. An exclamation sign appears at the right of the password field if both passwords do not match.



Creation of a new iTop user

**Properties** Profiles Allowed Organizations

Contact (person) Pablo Picasso

Login pablo

Password \*\*\*\*\* (Confirm) \*\*\*\*\* Reset Password

Language English (English)

Cancel Create

A user record defines:

- The favorite language of this user, that will be used for displaying the iTop user interface.

- The contact linked to this user account. This contact is also used - for portal users - to determine the default organization of the portal.
- The list of profiles for this account. Each iTop user account must have at least one profile.

The screenshot shows the 'Creation of a new iTop user' dialog box with the 'Profiles' tab selected. It features a table for assigning roles and profiles to a user.

| Reason                   | Profile               | Description  |
|--------------------------|-----------------------|--|
| <input type="checkbox"/> | Change Implementor    | Person executing the changes                             |
| <input type="checkbox"/> | Configuration Manager | Person in charge of the documentation of the managed CIs |

Buttons at the bottom include 'Cancel', 'Create', 'Remove selected objects', and 'Add Profiles...'. The 'Add Profiles...' button is highlighted.

The “Add Profiles...” button displays the search window for selecting the profiles you want to assign to the user.

The screenshot shows the 'Add User to profile objects to link with the iTop user' search window. It includes a search bar and a list of profiles.

Search for Profile Objects

Name:

Search

| Profile   | Description   |
|---|---|
| <input type="checkbox"/> Administrator                    | Has the rights on everything (bypassing any control)  |
| <input type="checkbox"/> Change Approver                  | Person who could be impacted by some changes  |
| <input checked="" type="checkbox"/> Change Implementor    | Person executing the changes  |
| <input type="checkbox"/> Change Supervisor                | Person responsible for the overall change execution   |
| <input checked="" type="checkbox"/> Configuration Manager | Person in charge of the documentation of the managed CIs  |
| <input type="checkbox"/> Document author                  | Any person who could contribute to documentation  |
| <input type="checkbox"/> Portal user                      | Has the rights to access to the user portal. People having this profile will not be allowed to access the standard application, they will be automatically redirected to the user portal. |
| <input type="checkbox"/> Problem Manager                  | Person analysing and solving the current problems   |
| <input type="checkbox"/> System Backup Agent              | Person in charge of creating backup records   |

Buttons at the bottom include 'Cancel' and 'Add'.

The profiles assigned to the user can be changed later on using the “Modify” action for a user.

## Import logins massively

To create many logins in a few steps, you can use the CSV import tools.

[Import user account in CLI](#)

## **Restricting access to a set of Organizations**

Administrators can define for each user the list of organizations she/he is allowed to access using the “Allowed Organizations” tab. If no organization is selected, the user is allowed to see all of them.

In case of a hierarchy of organizations (when some organizations have a parent organization), the rights are inherited from the parent to the child organizations. In other words, if a user has the rights to access the parent organization, then this user has also the rights to access all the child organizations of this organization.

All the objects belonging to an organization which is forbidden to a given user are completely hidden from this user. For this user, the application behaves as if such object did not exist.

If the contact corresponding to a user is in a forbidden organization for her/him, it looks (for this user) as if the contact does not exist. Since all users accessing the portal must be linked to a contact, such a configuration will prevent this user from accessing the iTop portal!

The selected organizations can be changed later on using the “Modify” action for a user.

## **Changing a user password**

The administrator can change a user password if required by simply using the “Modify” action for a user. This can be useful to reset the password of a user.

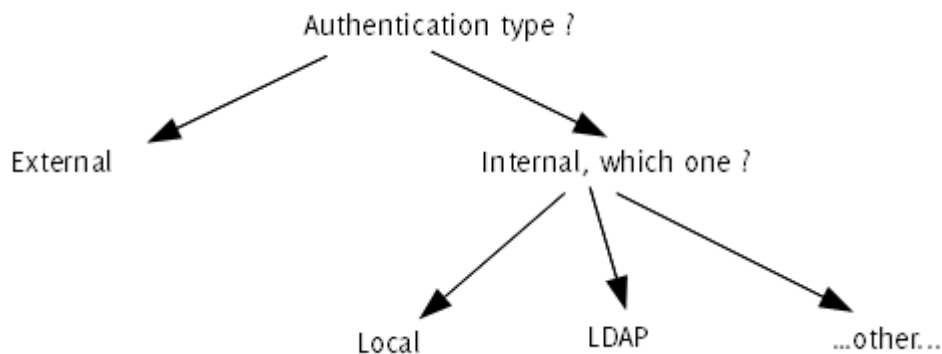
Users can change their own password by clicking on the “Log-Off” menu and selecting “Change password...”.

The passwords are stored encrypted (one way) in the iTop database, and therefore cannot be reconstructed from the content of the database.

## **User Authentication Options**

In order to seamlessly integrate with your environment, iTop can use several authentication mechanisms. Some part of the iTop behavior is defined at the web server level, however it is possible to mix the different authentication types on a per user basis.

For example if you already have an LDAP server for authenticating the end-users, it may be convenient to use this LDAP authentication for end-users and local passwords for functional logins (for automation processes and scripts).



In the context of this document, “External authentication” means that the user authentication is performed completely outside of iTop (by the web server or some kind of web server add-on). In this case the iTop application will just be passed the name of the authenticated user, once it has been validated by the web server. This is what happens if you install iTop under Apache in a directory protected with a .htaccess file.

When deploying iTop, you have to decide whether the application will use an external authentication mechanism or rely on the iTop internal mechanism. If the application uses “internal” authentication, then several types of authentication may coexist for the same installation (for example you can configure iTop to use the form based dialog and use a mix of LDAP and Internal users).

### Creation of a new User

Select the type of User to create: External user Apply

- External user
- LDAP user
- iTop user


If the authentication is not “external”, then iTop will actively participate to this authentication. Currently iTop supports two types of internal authentications:

- “Local”: based on passwords stored encrypted in the iTop database
- “LDAP”: by interacting with a remote LDAP or Active Directory server

## Logon Types

In order to authenticate a user, the application has to prompt the user for a login and a password. iTop provides several methods to handle this dialog with the end user:

- “Form based dialog”: this is the default mechanism that displays a formatted dialog to the end-user, as shown on the picture below:

A screenshot of a web-based login form. At the top, there is a logo consisting of a blue square with a white grid pattern and the text "iTop" in blue. Below the logo, the text "Welcome to iTop!" is displayed in a large, bold, blue font. Underneath, the instruction "Identify yourself before continuing" is shown in a smaller, black font. There are two input fields: "User Name:" followed by a text box, and "Password:" followed by a text box. Below these fields is a button labeled "Enter iTop".

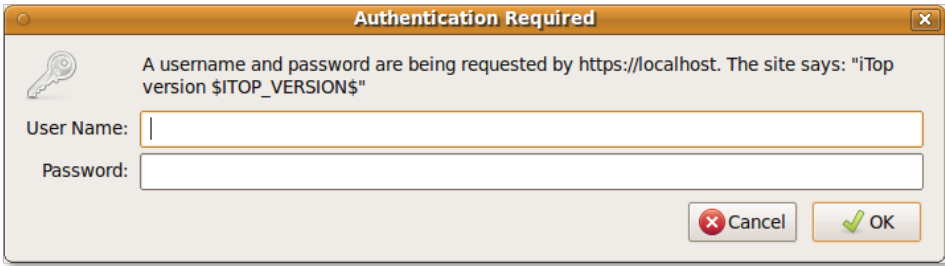
**Welcome to iTop!**

Identify yourself before continuing

User Name:

Password:

- “Basic Authentication”: this mechanism relies on the Basic Authentication of the HTTP protocol. With most web browsers, this appears as a pop-up to the end user. Command line tools (like `wget`) can generally use this type of authentication.

A screenshot of a standard web browser authentication dialog box. The title bar reads "Authentication Required". On the left is a key icon. The main text says: "A username and password are being requested by https://localhost. The site says: 'iTop version \$ITOP\_VERSION\$'". Below this text are two input fields: "User Name:" and "Password:". At the bottom right are two buttons: "Cancel" (with a red X icon) and "OK" (with a green checkmark icon).

**Authentication Required**

A username and password are being requested by https://localhost. The site says: "iTop version \$ITOP\_VERSION\$"

User Name:

Password:

- “CAS”: in this mode iTop delegates the authentication to a CAS compatible server
- “URL”: Some scripts or applications have very limited capabilities for passing authentication credentials. In such a case it is possible to pass directly a login and password as parameters in the URL. Note that due to security considerations, this method is disabled by default, but can be enabled by configuration.
- “External”: This type of login is used when the authentication has already been performed by the web server and iTop is just passed the name of the authenticated user as server side variable.

The different possibilities can be summarized in the table below:

| Logon Type | Purpose  | Limitations  |
|------------|--|--|
| form       | Form based logon, best suited for interactive logon. Provides also a logout/logoff functionality   | Not designed for use by batch/scripts or when the web browser is “hidden” from the end-user (i.e. Excel web queries) |
| basic      | Basic HTTP authentication. Compatible with all applications supporting this protocol (e.g. wget, Excel web queries)                        | No logoff/logout functionality.  |
| cas        | Use JA-SIG CAS APIs to connect to a CAS server for authentication  | Provides transparent single-sign-on with other CAS applications, for example a LifeRay portal                        |
| url        | Compatible with any application. Needed if you want to run web queries in OpenOffice.  | The password has to be written in clear text in the address of the page !  |
| external   | Used when the authentication is performed by the web server itself when an “external” user connects before even accessing the application. | iTop does not perform any authentication. iTop simply trusts the parameters passed by the web server.                |

## Configuration file

The parameter `allowed_login_types` in the iTop configuration file determines the behavior of the iTop application by defining which logon methods are enabled, and in which order the application will try them.

This parameter is a text string, in which the values are separated by the pipe ( | ) character. The string is made of a combination of form, basic, url, cas and external in the desired order.

The default mode is the first one in the list. It defines the behavior of the application when no credential is supplied. The first “mode” must be one of form, basic, cas or external.

### 'login\_mode' page parameter

When connecting to any iTop web page, it is possible to override the default login mode, by specifying the parameter “login\_mode” in the URL.

The value is one of the allowed modes: form, basic, url, cas, external. This login mode will be used (for this connection only, it's not persisted)

provided that the corresponding mode is enabled in the configuration file. If the value is not valid, it is ignored and the first “mode” defined in the configuration file is used.

For example if the configuration file contains:

```
allowed_login_types => 'form|basic|external',
```

and a page is called with the parameter ?login\_mode=basic, then the Basic authentication mechanism will be used for this page.

Now if a page is called with the parameters ?login\_mode=url&auth\_user=foo&auth\_pwd=bar, the url mode being excluded from the configuration, the default mode (in the example above: form) will be used.

## Example

For an instance of iTop in which the users are authenticated either by the local password (stored in the iTop database) or by an LDAP server, and using either their web browser (to navigate in the application) or Microsoft Excel to run some reports (as web queries), the following value can be put in the configuration file:

```
'allowed_login_types' => 'form|basic',
```

When connecting to the application, end-users will be prompted to enter their login/password using the usual iTop login form. When using the application interactively, the end-user can log off the application using the “log off menu”.

When using Excel, it is possible, by adding &login\_mode=basic at the end of the link to the page used for the web query, to have Excel prompt the user for their login/password when the web query is launched for the first time.

In this configuration is it also possible to have scripts retrieving data from iTop using a command line application like wget, with the following syntax:

```
wget --http-user=<login> --http-password=<password> -O test.csv  
--auth-no-challenge  
"http://127.0.0.1/webservices/export.php?expression=SELECT%20Contact&  
format=csv"
```

Or by using this alternate command line:

```
wget --no-check-certificate --http-user=admin --http-password=admin2 -O
test.csv
"https://127.0.0.1/webservices/export.php?expression=SELECT%20Contact
&format=csv&login_mode=basic"
```

In the first command line, wget will send automatically the credentials to the web server (using the Basic authentication mechanism) and iTop will detect their presence.

In the second case, wget will not send any credentials first, but telling iTop that the preferred way to authenticate is “basic” (i.e. basic authentication), wget will get a response “401 unauthorized” from the web server, and will retry with the supplied credentials.

## Basic Authentication and Apache in CGI mode

PHP inside the Apache web server can be configured to run in two different modes: as a module or as a CGI.

When using the basic authentication mechanism, iTop relies on the web server to populate two PHP internal variables: `$_SERVER['PHP_AUTH_USER']` and `$_SERVER['PHP_AUTH_PASS']`. When PHP runs as an Apache module, the two variables are automatically populated by the web server when the end-user fills the “basic authentication” pop-up dialog but this is not the case when PHP runs as a CGI.

In this latter case, a simple rewrite rule can be used to pass the authentication information to PHP.

Here is how to do it:

Make sure that the rewrite engine module is activated in Apache, otherwise the “rewrite rule” will be ignored (if you are not sure if the rewrite engine is no, you can comment out the two lines `<IfModule mod_rewrite.c>` and `</IfModule>` in the sample below. Then restart Apache. If you seen an error about “RewriteEngine on” not being recognized as a valid configuration directive in the Apache log file, then rewrite engine is not configured on your web server).

Use your favorite test editor to write the following code in a `.htaccess` file at the root of the iTop web directory (make sure that the user that runs the PHP/CGI process has enough rights to read this file).

```
<IfModule mod_rewrite.c>
RewriteEngine on
RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization},L]
</IfModule>
```

This causes the Apache server to populate the server variable `$_SERVER['HTTP_AUTHORIZATION']` with the credentials (base64 encoded). iTop reads and decodes this variable to extract the login and password.

## Integrating with an external authentication

iTop can use an external authentication mechanism (Apache's .htaccess file, CA SiteMinder agent, etc...), however the users' profile is always defined in iTop and for each user allowed to connect to iTop there must be an iTop "External user" record in the database.

When using an external authentication mechanism, any request to the iTop application is intercepted and the end-user is asked to enter her/his authentication before connecting to iTop. In such a case iTop is unaware of the authentication means (login/password, certificate stored on smart card...). The only information that matters to iTop is the "login" (i.e. identifier) of the authenticated user.

iTop uses the standard mechanism for PHP applications by reading the server variable `$_SERVER['REMOTE_USER']`. If an "External user" record with the corresponding login is found in the iTop database, this user will become connected to iTop. If no match is found, the next authentication mechanism will be used and the user will be prompted to authenticate with iTop.

Depending on the external authentication mechanism used, the web server may place the authentication information in different variables, therefore the variable used to retrieve this "external authentication" can be specified in the iTop configuration file. Look for the parameter:

```
ext_auth_variable => '$_SERVER[\`REMOTE_USER\`]',
```

For example if your single-sign-on system uses the cookie variable `auth_user` to pass the user name, you can use something like:

```
ext_auth_variable => '$_COOKIE[\`auth_user\`]',
```

In order to ensure that the external authentication is used first (preventing iTop from prompting the already authenticated user a second time), make sure that in the iTop configuration file, the order for `allowed_login_types` specifies "external" as the first login mode, as shown below:

```
'allowed_login_types' => 'external|form|basic',
```

## Sample configuration for external authentication

In the sample below the users are authenticated by a `.htpasswd` file stored in `/var/www/`.

### iTop configuration

```
$MySettings = array(  
    ...  
    'allowed_login_types' => 'external|form|basic',  
    ...  
);
```

Apache configuration file

[httpd.conf](#)

```
<Directory /var/www/>  
    AuthUserFile /var/www/.htpasswd  
    AuthName "Please identify yourself..."  
    AuthType Basic  
    require valid-user  
</Directory>
```

### Integration Test script

If you use a third party authentication mechanism, you can place the following PHP script in a file in the `/pages` directory of iTop. Then point your web browser to it to test if your external authentication passes the appropriate information to iTop.

[auth-test.php](#)

```
<?php  
//  
// Script to test External Authentication  
//  
require_once('../approot.inc.php');  
require_once(APPROOT.'core/config.class.inc.php');  
require_once(APPROOT.'application/utils.inc.php');  
  
echo "<h1>External Authentication: Integration Test Script</h1>";  
try  
{
```

```

    $aAllowedLoginTypes =
utils::GetConfig()->GetAllowedLoginTypes();
    $sAllowedLoginTypes = implode(' |', $aAllowedLoginTypes);
    echo "<p>allowed_login_types set to
'sAllowedLoginTypes' </p>\n";

    if (!in\_array('external', $aAllowedLoginTypes))
    {
        echo "<p>Check your iTop configuration file, ".
            "<b>allowed_login_type</b> does not enable ".
            "'external' authentication.</p>";
    }
    else
    {
        echo "<p>Ok, external authentication is enabled by
allowed_login_type.</p>";
        $sExternalAuthVariable =
utils::GetConfig()->GetExternalAuthenticationVariable();
        echo "<p>External Authentication Variable:
'sExternalAuthVariable' </p>\n";
        if ($sExternalAuthVariable != '')
        {

            $sEval = '$bVarIsSet =
isset('.$sExternalAuthVariable.')';
            eval($sEval);
            if ($bVarIsSet)
            {
                $sEval = '$sAuthUser =
'.$sExternalAuthVariable.'';
                eval($sEval);
                echo "<p>External Authentication
Variable set to: ".
                    "'$sAuthUser' </p>\n";
                if ($sAuthUser != '')
                {
                    echo "<p><b>Ok, integration with
external authentication ".
                        "successful !</b></p>\n";
                }
                else
                {

```

```

                                echo "<p>Empty user information
passed by the web ".
                                "server.</p>\n";
                                }
                                }
                                else
                                {
                                    echo "<p>External Authentication
Variable <b>NOT</b> set.</p>\n";
                                }
                                }
                                else
                                {
                                    echo "<p>Check your iTop configuration file,
<b>ext_auth_var</b> is set ".
                                    "to an empty string.</p>\n";
                                }
                                }
                                }
                                }
                                catch(ConfigException $e)
                                {
                                    echo "<p>iTop configuration file not found. Did you already
install iTop ?</p>\n";
                                }
                                echo "<hr/>\n";
                                echo "<p>For information:</p>";
                                echo "<pre>\$_SERVER variable:\n";
                                print_r($_SERVER);
                                echo "\$_COOKIE variable:";
                                print_r($_COOKIE);
                                echo "</pre>\n";
                                ?>

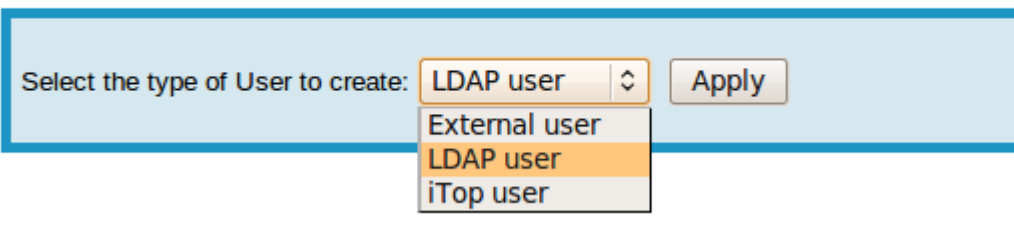
```

## Configuration of LDAP authentication

iTop supports LDAP authentication against a remote LDAP server (or Windows Active Directory server). Both types of users (users with an account stored in the iTop local database and users with an LDAP account) can coexist on the same instance of iTop, on a per user basis.

LDAP users are created by selecting the appropriate type when prompted for a “New user Account” :

## Creation of a new User



Select the type of User to create: LDAP user Apply

External user  
LDAP user  
iTop user

## Configure iTop to connect to your LDAP server

Once iTop is installed, open the file 'config-itop.php' and look for the section `authent-ldap` . It should looks as shown below:

[config-itop.php](#)

```
'authent-ldap' => array (  
    'host' => 'localhost',  
    'port' => 389,  
    'default_user' => '',  
    'default_pwd' => '',  
    'base_dn' => 'dc=mycompany,dc=com',  
    'user_query' => '(&(uid=%1$s))',  
    'options' => array (  
        17 => 3,  
        8 => 0,  
    ),  
)
```

Where:

- The 'host' parameter must contain the IP address or host name of your LDAP server
- The 'port' parameter is the TCP port number to be used for connecting to LDAP (LDAP's default is 389)
- 'default\_user' and 'default\_pwd' identify an LDAP account with enough rights to connect to your LDAP server (in read-only mode) in order to search for the specified user. On most systems an anonymous user can do this, so you can leave these two fields blank.
- 'base\_dn' defines the "root" used for searching for the user/logins. It can be something like "dc=mycompany,dc=com" or "ou=People,dc=mycompany,dc=com". If you're not sure, ask your LDAP administrator.
- 'user\_query' defines a LDAP query that will be used for searching for the users. You can write whatever valid LDAP query using any of the following placeholders:
  - %1\$s : iTop login of the user (i.e. what he/she types in the login form)
  - %2\$s : User's First Name
  - %3\$s : User's last name

- %4\$s : User's email

When connecting to a Windows Active Directory the query to retrieve the user based on her/his Windows login looks like: '(samaccountname=%1\$s) '

The parameters %2\$s, %3\$s and %4\$s are non-empty only if the User record (in iTop) is associated with a Person (i.e. if the “Contact” field is not empty for this User).

The last part of the configuration, options defines some LDAP specific options. See <http://www.php.net/manual/en/function.ldap-set-option.php> for the full list of possible options. You can use either the PHP constants (like LDAP\_OPT\_PROTOCOL\_VERSION – with no quote) or their numeric equivalent (i.e. LDAP\_OPT\_PROTOCOL\_VERSION equals 17). In the example above:

```
LDAP_OPT_PROTOCOL_VERSION => 3
LDAP_OPT_REFERRALS => false
```

If you get an error 'Cannot bind to LDAP server, invalid DN syntax', check the syntax of the 'default\_user' in your iTop configuration file. For some LDAP servers, the user name must be a complete name like: 'cn=admin,dc=mycompany,dc=com' and not just 'admin'.

## Managing Organizations

In iTop Organizations are used to define the boundaries which are the basis of the access restriction policies. Most of the objects managed in iTop have a field “Organization”. This field generally represents the organization “owning” the object (for example for configuration items like Servers, PCs, Racks...). Organizations are also used in Service Management, in this case the “owner” of a service is the “provider” of this service, whereas the “owner” of a contract is generally the customer “buying” services through this contract.

Organizations can be organized in a parent/child hierarchy, in which case iTop behaves as if the parent organization was also “owning” the objects owned by all its child organizations.

Only administrators and configuration managers can add, modify or remove organizations.

To add an organization click on “Organizations” in the “Data Administration” module then click on the “New...” button.


## All Organizations

Total: 2 objects.

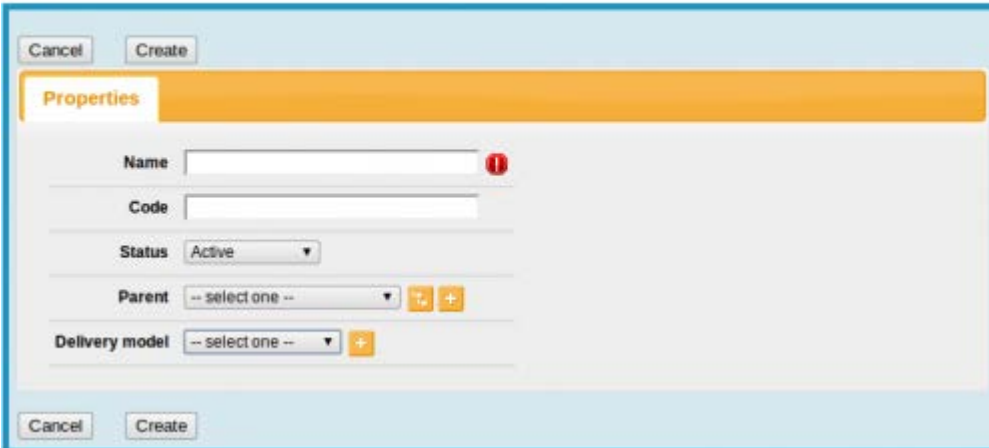
New... Other Actions -

| Organization         | Status | Parent    |
|----------------------|--------|-----------|
| Demo                 | Active | undefined |
| My CompanyDepartment | Active | undefined |

The form to create an organization contains:

- The name of the organization
- Its code
- Its status
- And a parent organization if you want to create hierarchy
- The Delivery Model for this Organization. 

### Creation of a new Organization



The form for creating a new organization is titled "Creation of a new Organization". It features a "Properties" section with the following fields:

- Name:** A text input field with a red error icon.
- Code:** A text input field.
- Status:** A dropdown menu currently set to "Active".
- Parent:** A dropdown menu set to "-- select one --" with two orange buttons (+ and -) to its right.
- Delivery model:** A dropdown menu set to "-- select one --" with an orange (+) button to its right.

At the top of the form are "Cancel" and "Create" buttons. At the bottom are "Cancel" and "Create" buttons.

You can later modify the attribute of a given organization by clicking on “Modify” action.

## Browsing the Data Model

Depending on the modules installed during the setup, the data model of iTop (also called the objects schema) can vary. You can view the current data model used by iTop by clicking on the link “Data Model” in the “Admin Tools” menu. An explorer tree-view allows you to navigate through the hierarchy of classes (in alphabetical order).

Each entry contains the display name of the class (potentially localized) and its internal PHP name (the name used when writing OQL queries and PHP code).

Clicking on a class name displays the details for this class.

### iTop objects schema



To quickly find the link to a specific class in the data model, use the “Find” feature of your web browser (CTRL + F) to search for the class name within the page.

Data model tabs definition:

| Tab             | Description   |
|-----------------|---|
| Attributes      | Displays all the attributes for this class.                       |
| Search criteria | Displays all the attributes you can use in search forms or in OQL |

|                     |   |
|---------------------|---|
| Referencing classes | Displays all classes having a reference to the current one              |
| Related classes     | Displays classes related to this one via an external key                |
| Lifecycle           | Displays the lifecycle graph for the class and some related information |
| Notification        | Display the notifications configured for the selected class.            |

## Running Object Queries

iTop uses its own internal query language to retrieve the objects from the database. Inside the MySQL database the objects manipulated by iTop are stored using several relational tables, but the **Object Query Language** (OQL) hides this complexity by manipulating objects almost as if they were plain SQL tables.

The **OQL** syntax is quite similar to the SQL syntax, except that there is no need to indicate which columns to retrieve, since iTop always manipulates “objects” and retrieves the columns as necessary.

The menu “Run Queries” allows Administrators to test and run **OQL** queries (For more information about the **OQL** syntax, refer to [Object Query Language Reference](#)). Some predefined queries are listed as examples at the top of the page (click on “Query Examples” at the top).

| Purpose  | OQL Expression   |      |
|--|--|------|
| Applications   | SELECT Application   | Test |
| Person having an 'A' in their name                             | SELECT Person AS p WHERE p.name LIKE '%A%'   | Test |
| Changes planned on new year's day                              | SELECT Change AS ch WHERE ch.start_date >= '2009-12-31' AND ch.end_date <= '2010-01-01'  | Test |
| IPs in a range   | SELECT InfrastructureCI AS dev WHERE INET_ATON(dev.management_ip) > INET_ATON('10.22.32.224') AND INET_ATON(dev.management_ip) < INET_ATON('10.22.32.255')   | Test |
| IPv4 interfaces of equipment in production for customer 'Demo' | SELECT NetworkInterface AS i JOIN InfrastructureCI AS dev ON i.device_id = dev.id WHERE i.status = 'production' AND dev.status = 'production' AND dev.owner_name = 'Demo' AND i.physical_type = 'ethernet' | Test |
| My tickets   | SELECT Incident AS i WHERE i.agent_id = i.current_contact_id   | Test |
| People being owner of an active ticket                         | SELECT Person AS p JOIN Incident AS i ON i.agent_id = p.id WHERE i.status != 'Closed'  | Test |
| Contracts terminating in the next thirty days                  | SELECT Contract AS c WHERE c.end_date > NOW() AND c.end_date < DATE_ADD(NOW(), INTERVAL 30 DAY)  | Test |
| Orphan tickets (opened one hour ago, still not assigned)       | SELECT Incident AS i WHERE i.start_date < DATE_SUB(NOW(), INTERVAL 60 MINUTE) AND i.status = 'New'   | Test |
| Long lasting incidents (duration > 8 hours)                    | SELECT Incident AS i WHERE i.close_date > DATE_ADD(i.start_date, INTERVAL 8 HOUR)  | Test |

Expression to evaluate:

Evaluate

To test a query, type the **OQL** expression in the text area, and click on “Evaluate” to get the results.

Since the result of the query is displayed as an HTML table, only objects that can be displayed as HTML can be queried by this method. The display of a list of internal iTop objects may cause an

error when queried this way.

## Query Phrasebook

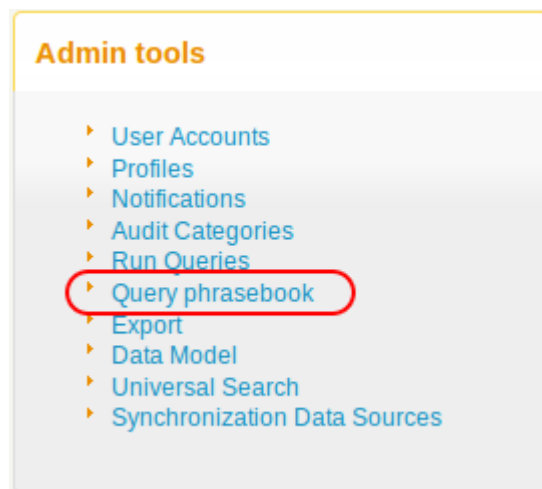
The Query Phrasebook stores a list of ready-to-run OQL queries so that end-users can easily execute complex queries defined by an administrator.

This is especially useful for building reports: simply integrate the results of such queries into a spreadsheet (for example using Excel's web query feature).

Only administrators have access to the "Query Phrasebook" menu.

## Creating a Stored Query

Click on the menu "Query phrasebook" in the "Admin tools" section, to display the list of stored queries:



The use the link "Create a new Query" link or the "New..." action to display the creation form:

## Creation of a new OQL Query

Cancel Create

**Properties**

**Name** Open Changes for a Customer

**Description** List all open Change tickets for a given customer

**Expression** SELECT Change WHERE status != 'closed' AND status != 'rejected' AND org\_id = :customer

Test query

**Fields** ref,status,title

Cancel Create

The fields “Name” and “Description” are used to identify the queries.

The field “Expression” contains the actual OQL query. A query can contain parameters (identified by the : placeholder followed by the parameter’s name). Using parameters allows to define generic queries that can be used in different contexts.

For example the following OQL query lists all the open Changes for a given customer (specified by its name):

```
SELECT Change WHERE status != 'closed' AND status != 'rejected' AND  
org_name = :customer
```

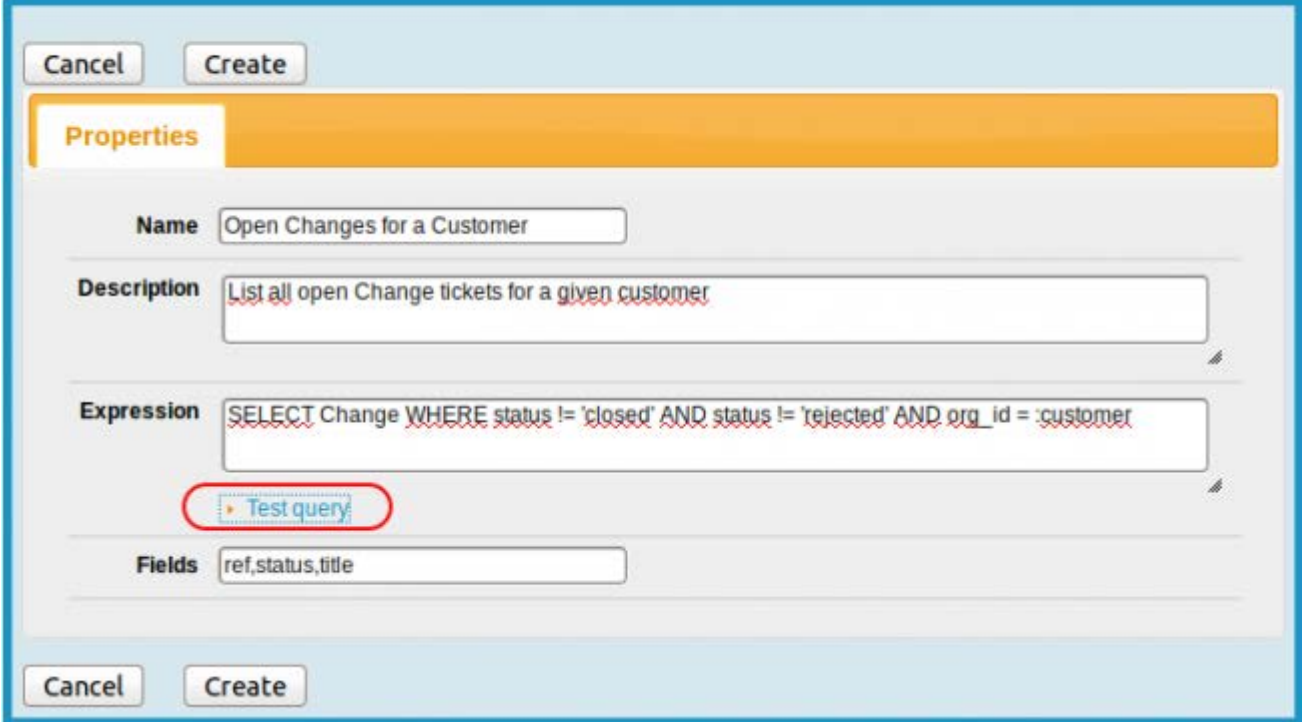
By default, all the fields of the selected objects are displayed as the output of the query. The field “Fields” allows to restrict this output to the given list of fields (by passing a comma separated list of field *codes*).

In the output, the order of the fields does not depend on the order in which the fields have been listed in “Fields”.

## Testing a stored query

To test your query while writing it, click on the link “Test query” below the edit box. The page “Run Query” then opens in another window of your browser. If the query contains parameters, the page displays a form to enter them.

### Creation of a new OQL Query



The screenshot shows a web-based form titled "Creation of a new OQL Query". At the top, there are two buttons: "Cancel" and "Create". Below this is a yellow header bar with the word "Properties" in orange. The form contains several fields: "Name" with the value "Open Changes for a Customer", "Description" with the text "List all open Change tickets for a given customer", and "Expression" with the SQL query "SELECT Change WHERE status != 'closed' AND status != 'rejected' AND org\_id = :customer". Below the "Expression" field is a red-outlined button labeled "Test query". At the bottom, there is a "Fields" field with the value "ref,status,title". The form is enclosed in a light blue border with "Cancel" and "Create" buttons at the bottom.

## Executing a stored Query

The execution of stored queries is done using the page `webservices/export.php`. By default, a link to this page with the suitable options for integrating into Excel is listed in the details of the stored query, but other options can be passed to the page to produce different outputs.

OQL Query: **Open Changes for a Customer**

Properties

History

**Name**

Open Changes for a Customer

**Description**

List all open Change tickets for a given customer

**Expression**

SELECT Change WHERE status != 'closed' AND status != 'rejected' AND org\_id = :customer

**Fields**

ref,status,title

URL to use for MS-Excel web queries:

`https://localhost/2.0/webservices/export.php?format=spreadsheet&login_mode=basic&query=1&arg_customer=["customer"]`

Refer to the documentation of [export.php](#) for more information about all the possible options.

For help about integrating the results into Excel, please refer to the chapter [How to create an Excel report using iTop queries](#)

## Notification

iTop integrates a notification system linked to the life cycle of the objects. This allows administrators to define e-mail notification rules when an object of a given class enters or leaves a specified state, when a new object is created, when an update occurs from the portal or when certain thresholds are reached.

The notification mechanism is divided in two parts:

- Triggers define **when** notifications have to be sent. Example: when a ticket reaches the state “assigned”.
- Actions define **what** will be done. In the current version of iTop, the only available kind of action consist in sending an email.

For a given trigger you can define several actions to be executed, and their sequence. Also, a given action can be executed by several triggers.

Use the link “Notifications” in the “Admin tools” menu to manage triggers and actions:

**Configuration of the Notifications**

• Help

In iTop the notifications are fully customizable. They are based on two sets of objects: triggers and actions.


**Triggers** define when a notification will be executed. There are 5 types of triggers for covering 3 different phases of an object life cycle:

1. the "on object creation" triggers get executed when an object of the specified class is created
2. the "on entering a state" triggers get executed before an object of the given class enters a specified state (coming from another state)
3. the "on leaving a state" triggers get executed when an object of the given class is leaving a specified state
4. the "on threshold" triggers get executed when a threshold for TTR or TTO as been reached
5. the "on portal update" triggers get executed when a ticket is updated from the portal

**Actions** define the actions to be performed when the triggers execute. For now there is only one kind of action consisting in sending an email message. Such actions also define the template to be used for sending the email as well as the other parameters of the message like the recipients, importance, etc.

A special page: [email.test.php](#) is available for testing and troubleshooting your PHP mail configuration.

To be executed, actions must be associated to triggers. When associated with a trigger, each action is given an "order" number, specifying in which order the actions are to be executed.



**Triggers** **Actions**

Available triggers

Total: 3 objects.

[New...](#) [Other Actions -](#) [X-](#)

| Trigger               | Type                          |
|-----------------------|-------------------------------|
| 75% of SLA reached    | Trigger (on threshold)        |
| New User Request      | Trigger (on object creation)  |
| User Request Assigned | Trigger (on entering a state) |

- The “Triggers” tab displays all created triggers.
- The “Actions” tab displays all Actions

## Creating an action

Before creating a useful trigger, at least one action must be defined. Email actions are templates for formatting the messages to be sent, the define the content of the message as well that the subject, sender and recipients.

To create a new action, go to the “Actions” tab and click on “New...”. The following wizard appears:

Creation of a new Email notification

Properties Related Triggers

Name

Description

Status Being tested

Test recipient

From

Reply to

To

Cc

The mandatory fields for an email action are:

- Subject: the subject of the message. May contain placeholders.
- Body: the body of the message. May contain placeholders. By default iTop sends all the messages with the MIME Type text/html for the body of the email, so you may put HTML tags to format the message.
- From: this field contains a static email address. Note that some mail servers will reject the message if the “from” address is not valid.

The contacts to be notified in the “To”, “Cc”, and “Bcc” are defined by an OQL query. This allows to specify multiple recipients for the notification, like “all the contacts attached to a ticket” or “all the contacts on the impacted site” ... (Refer to [Object Query Language Reference](#) for more information about writing OQL queries)

This OQL query must return a list of objects containing an e-mail attribute, namely:

- Contact
- Person
- Team

For instance, to notify all persons whose name starts with John, the **To** field can contain:

```
SELECT Person WHERE name LIKE 'John%'
```

The query can contain placeholders (using the syntax `:this→attribute`) that refer to the current object for which the notification is being sent.

For example, to send a notification to the person who is the “caller” of a ticket, the **To** field will contain:

```
SELECT Person WHERE id= :this->caller_id
```

If the list returned by the query is empty no mail is sent. By default the importance of the mail is “normal”.

## Placeholders

Inside the “Subject” and “Body” of the message, you can refer to fields of the object that triggered the action. The syntax to be used for such placeholders is `$this->attribute$`. Where attribute is the code of the field as defined in the Data Model.

There are also some specific placeholders:

| Placeholder                               | Meaning   |
|---|---|
| <code>\$this-&gt;name()</code>            | The name of the current object                          |
| <code>\$this-&gt;hyperlink()</code>       | The url to access the current object in iTop            |
| <code>\$this-&gt;hyperlink(portal)</code> | The url to access the current object in the iTop portal |
| <code>\$this-&gt;html(attribute)</code>   | The HTML representation of the value of the attribute   |

Moreover, for Case Log attributes, the syntax `$this->head(attribute)` returns the text of the latest entry in the case log.

## Testing notifications

To test a new action, you can use the status “Being tested” and fill “Test recipient” with a test address. In that case, the notification will be sent to this latter address. Once the notification have been tested, change its status to “In Production” to have notifications flow to their actual recipients.

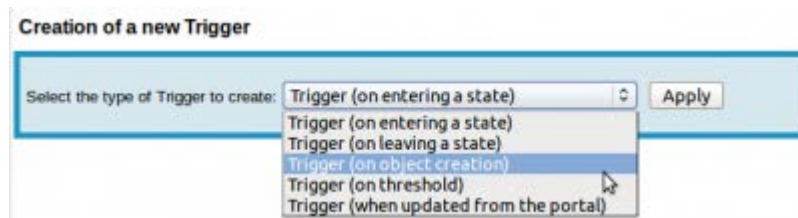
If you want to de-activate an action, just set its status to “Inactive”.

## Creating a trigger

- When a new object is created
- When an object enters in a given state

- When an object leaves a given state
- When an object is updated from the iTop portal
- When the given threshold for a Time-To-Resolve (TTR) or a Time-To-Own (TTO) is reached

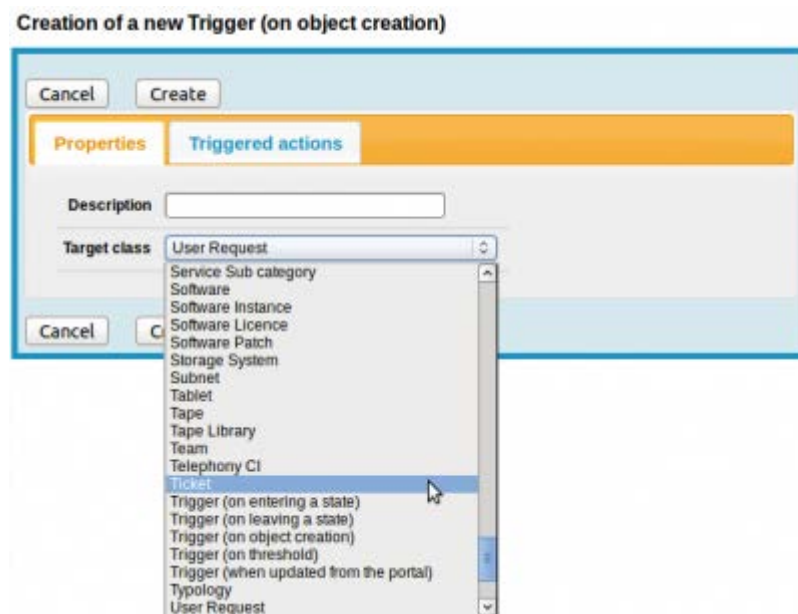
To create a new trigger, click on “New” in action drop down list for the given category in “Trigger” tab. The following wizard appears:



You have to select which type of trigger you want to create:

- Trigger (on entering a state)
- Trigger (on leaving a state)
- Trigger (on object creation)
- Trigger (on threshold)
- Trigger (when updated from the portal)

Once you have selected the type of trigger you get the following form:



For each trigger you have to define the class of object for which this trigger is applicable and for “on entering a state” and “on leaving a state”, the state. The value to be entered for the “state”, is actually the internal code of the state, as defined in the data model.

State codes can be seen in the “Life Cycle” tab of the “Data Model”, section “Transitions”. The value code is the value listed between parentheses.

The “Triggered Actions” tab defines which action(s) will be executed when this trigger fires. Remember that one action can be linked to several triggers, so it’s possible to reuse some actions. The “Order” field determines in which order, for a given trigger, the actions are executed (actions are launched in ascending order).

We strongly encourage you to test triggers and actions before moving them to production, by using the “Being Tested” status on actions.

You can use the menu “Application log” where all notifications are tracked to check if a mail was triggered. A detailed log of event describes what happened with a given notification, for an easier troubleshooting.

You can as well see which notification had been sent for a given ticket (User Request, Incident, Change) using the tab “Notifications” in the details of the ticket.



You can also list all sent and failed notifications by using the page “Admin tools” / “Run Queries” and running the query:

```
SELECT EventNotification
```

If you are running iTop on a Linux server, make sure that the variable “sendmail\_path” value in php.ini. For example:

```
sendmail_path = "/usr/sbin/sendmail -t -i"
```

Depending on your actual environment, the configuration may be different. For example it is also possible to use SMTP as a proxy to the actual mail server, as explained in the following link:

<http://tombuntu.com/index.php/2008/10/21/sending-email-from-your-system-with-smtp/>

If you are running iTop on a Windows server, you need to make sure that the php.ini file contains

the following lines:

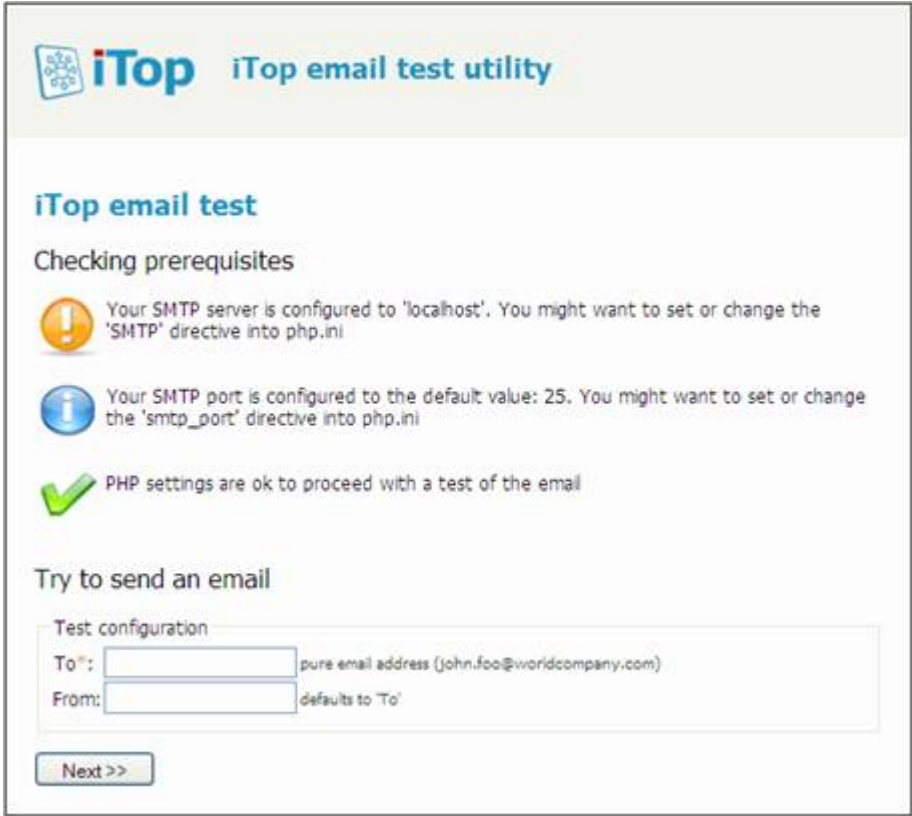
```
SMTP = <smtp server>
```

```
smtp_port = 25
```

In order to test mail notifications you can use, the “Test Page” (follow the link from the “Notifications” pages) or type:

```
http://<itop server location>/setup/email.test.php
```

The test page performs a number of checks on the PHP configuration and allows you to send a plain-text email to the recipient of your choice. This is useful for validating that the PHP configuration of the server is indeed correct for sending e-mails.



The screenshot shows the 'iTop email test utility' interface. At the top, there's a header with the iTop logo and the title 'iTop email test utility'. Below this, the main heading is 'iTop email test'. Underneath, it says 'Checking prerequisites'. There are three items listed: 1. A warning icon (yellow circle with an exclamation mark) indicating 'Your SMTP server is configured to 'localhost'. You might want to set or change the 'SMTP' directive into php.ini'. 2. An information icon (blue circle with an 'i') indicating 'Your SMTP port is configured to the default value: 25. You might want to set or change the 'smtp\_port' directive into php.ini'. 3. A success icon (green checkmark) indicating 'PHP settings are ok to proceed with a test of the email'. Below the prerequisites, there's a section titled 'Try to send an email'. It contains a 'Test configuration' box with two input fields: 'To:' and 'From:'. The 'To:' field has a placeholder text 'pure email address (john.foo@worldcompany.com)'. The 'From:' field has a placeholder text 'defaults to 'To''. At the bottom of the 'Test configuration' box is a 'Next >>' button.

## Email Configuration

iTop 2.0 supports two methods for sending emails: the built-in mail function of PHP or SMTP via the [Swift Mailer library](#) (installed with iTop). The configuration parameter `email_transport` determines which method is used for sending eMails from iTop. If the value of the `email_transport` parameter is `PHPMail` (which is the default value), then the built-in `mail()` function is used. If the value is `SMTP` then the SMTP transport of Swift Mailer is used.

When using PHP's mail function the language's settings are used. Check the [PHP documentation](#) for more information.

When using the SMTP transport, the following parameters can be set in the iTop configuration file:

| Configuration parameter         | Type    | Visible | Description                        | Default Value |
|---------------------------------|---------|---------|------------------------------------|---------------|
| email_transport_smtp.encryption | string  | No      | tls or ssl (optional)              |               |
| email_transport_smtp.host       | string  | No      | host name or IP address (optional) | localhost     |
| email_transport_smtp.password   | string  | No      | Authentication password (optional) |               |
| email_transport_smtp.port       | integer | No      | port number (optional)             | 25            |
| email_transport_smtp.username   | string  | No      | Authentication user (optional)     |               |

## Auditing the CMDB

The “Audit” is an iTop feature designed to check the consistency of information stored in the iTop database. Using the audit, you can answer questions like: “Do we have an active support contract defined for any device in production?” or “Do we know the localization of all the servers for on-site support?”



| Interactive Audit     |           |          |         |
|-----------------------|-----------|----------|---------|
| Audit Rule            | # Objects | # Errors | % Ok    |
| Servers in production | 126       | 63       | 50.00 % |
| Unknown location      |           | 63       | 50.00 % |

The verifications are named **audit rules**.

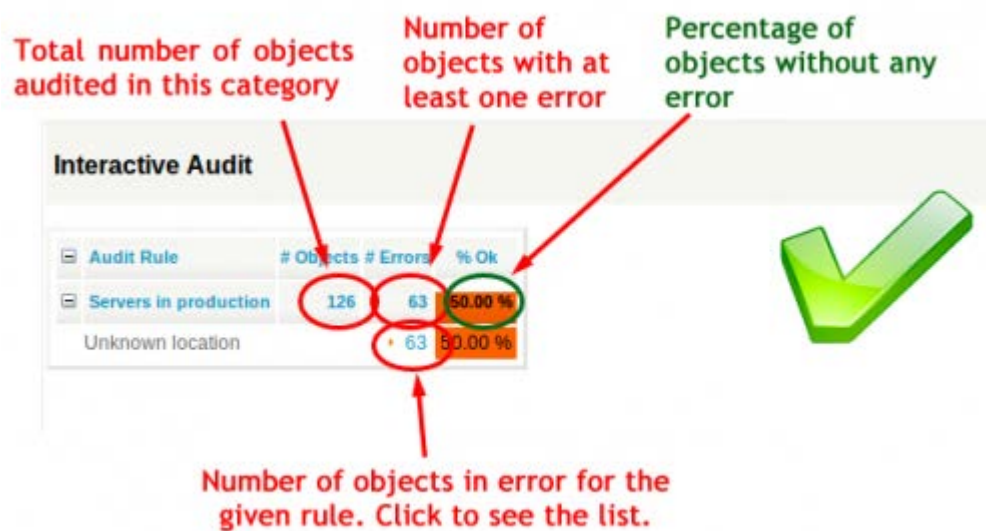
**Audit rules** are grouped into **audit categories**. A category defines the list of objects (the scope) that must be compliant with the rules. This scope is defined using an OQL query, so that the actual scope will be computed dynamically. Example: all devices that are in production ⇒ `SELECT Device WHERE status = 'production'`.

An **audit rule** defines the actual constraint that needs to be checked. For instance “Devices being attached to an active support contract.”. It is also defined by an OQL query.

To add or modify an **audit category** or an **audit rule**, click on “Audit Categories” in the “Admin tools” module.

## Understanding the Audit Report

The audit is run interactively by clicking the “Audit” link in the “Data administration” menu.



The audit report list all the categories to be audited. For each category the header line contains:

- The number of objects in this category
- The number of objects with **at least one error** (as reported by the category's rules)
- The percentage of objects without error in this category
- The for each rule, an additional line indicates:
  - the number of objects in error *for this rule*. Clicking on the number displays the list of objects in error
  - the percentage of objects without error *for this rule*

## Managing Audit Categories

An audit category is defined by a name, a description and a definition set. The definition set defines the scope of objects that will be subject to the related audit rules. It is an OQL query.

The screenshot shows a dialog box titled "Creation of a new Audit Category". It has two tabs: "Properties" (selected) and "Audit Rules". The "Properties" tab contains three input fields: "Category Name" with the value "Servers in production", "Audit Category Description" with the value "All servers in production", and "Definition Set" with the OQL query "SELECT Server WHERE status = 'production'". Below the "Definition Set" field is a "Test query" button. At the top and bottom of the dialog are "Cancel" and "Create" buttons.

To create or remove audit rules for a given category, use the button "Create a new Audit Rule" or "Delete" from the "Audit Rules" tab on the Audit Category

The screenshot shows a dialog box titled "Creation of a new Audit Rule". It has a single "Properties" tab. The "Properties" tab contains four input fields: "Rule Name" with the value "Unknown location", "Audit Rule Description" with the value "Servers which location is unknown", "Query to Run" with the OQL query "SELECT Server WHERE location\_id = 0", and "Valid objects?" with a dropdown menu set to "false". Below the "Query to Run" field is a "Test query" button. At the top and bottom of the dialog are "Cancel" and "Create" buttons.

## Managing Audit Rules

An audit rule is defined by a name, a description, the query to check and a Valid Object flag.

The query defines the list of objects (under the scope defined by the category) that pass/fail the audit.

Since it is sometimes easier to list the objects that pass the audit than the objects that fail, the flag “Valid Objects ?” is used to indicate whether the query returns the “valid” objects or the “invalid” ones.

Note that a rule is always linked to only one category; and this category determines the scope of the rule.

## Data backup

In iTop all the data (including the uploaded documents) are stored in the MySql database. Therefore it is highly recommended to have a database backup in place on a regular basis.

You can run a full backup of the database using the following mysqldump command:

On Linux systems:

```
/usr/bin/mysqldump --opt --default-character-set=utf8
--single-transaction --add-drop-database - -user=<user>
--password=<password> <DB> | gzip > <file>
```

On windows systems:

```
mysqldump.exe --opt --default-character-set=utf8 --single-transaction
--add-drop-database - -user=<user> --password=<password> <DB> > <file>
```

Where:

- <user> is the user name to connect to MySQL
- <password> is the corresponding password
- <DB> is the name of the database in which iTop is installed
- <file> is the name of the archive file to produce

Once the content of the database is dumped, just archive this dump and the file `conf/production/config-itop.php` to keep a full image of your iTop instance.

- Make sure that you export the data in UTF-8 (`--default-character-set=utf8`) otherwise accentuated characters will be lost.
- The option `--single-transaction` has two effects: it produces a consistent backup since all tables are exported in one transaction, and it provides a workaround for the error 1449 (definer does not exist) which may happen if you export a database that was already re-imported from another system or created by a no longer existing user.

## Restoring a backup

To restore a backup just re-import the MySQL dump and restore the configuration file.

### Passwords encryption

If the datamodel uses password fields with reversible encryption, then be sure that the same encryption algorithm is used on the new and old systems: if the `mcrypt` PHP extension was present on the old system, then the `mcrypt` extension must be present on the new system as well. If `mcrypt` was not present, it must not be present either.

Note: User accounts passwords use a one-way encryption and are not affected by the presence of `mcrypt`.

### Restoring a Windows Backup on Linux (iTop 1.x and 2.0 beta 1 *only*)

If you backup the iTop database on a Windows server, all the database table names will be in lowercase. When restored on Linux, the table `priv_internalUser` will be spelled `priv_internaluser`, which is different. In you cannot log into iTop after restoring a backup, check that the table is properly spelled. If needed, rename the table to `priv_internalUser`.

This should no longer occur starting with iTop 2.0 beta 2, since all the tables are now in lowercase.

## Background tasks

In order to operate properly, iTop maintenance operations and asynchronous tasks must be executed on a regular basis. In order to ease

the installation, all the background tasks have been grouped to be launched from a single file: `webservices/cron.php`

The following features rely on the activation of `cron.php`

- Asynchronous emails. By default, this option is disabled. To enable it, set 'email\_asynchronous' to 1 in the configuration file
- Check ticket SLA. Tickets reaching the limits will be passed into Escalation TTO/TTR states

## Scheduling on Windows

Use `schtasks.exe` or the 'at' command to schedule `cron.php` to run every 5 minutes.

```
schtasks.exe /create /sc minute /mo 5 /tn "iTop CRON" /tr  
"/var/www/itop/webservices/cron.cmd"
```

## Scheduling on Linux

Edit the crontab to execute the script every 5 minutes:

```
5 * * * * root /usr/bin/php /var/www/itop/webservices/cron.php  
--param_file=/conf/itop/params >>/var/log/itop-cron.log 2>&1
```

## Scheduling from the web

The page `cron.php` can also be executed through the web server. This is useful if you don't have access to the system (for example on shared hosting systems). In such a case you can rely on a [web cron](#) service to run it for you. But the frequency and execution times will probably be restricted compared to a natively scheduled job.

## Arguments

| Argument                | Description  | Default value            |
|-------------------------|--|--------------------------|
| <code>param_file</code> | Path to the parameters file  | <code>cron.params</code> |
| <code>auth_login</code> | User login - CLI mode only, if no parameters file is used                  | -                        |
| <code>auth_pwd</code>   | User password - CLI mode only, if no parameters file is used               | -                        |
| <code>verbose</code>    | if set to 1, more information will be given, use this for troubleshooting. | 0                        |

**Beware:** since the process runs continuously in background, the log files tend to become big pretty quickly. Don't set this options to 1 for a long period of time.

(New in iTop 2.0.1) if set to 1, a status about the scheduled tasks is `status_only` displayed and the process stops immediatly (can be launched in parallel 0 to the normal execution of `cron.php`).

## Parameter file

Since the arguments passed to `cron.php` on the command line are visible to other users on the system, a recommended practice is to move all the sensitive parameters away from the command line. This is the purpose of the “parameter file”.

The argument `param_file` can be used with most of the REST/CLI web services. By convention, the `cron.php` service searches for a parameter file name « `cron.params` » to read its parameters from.

- A parameter file contains key/value pairs.
- Comments start with a # (any character found after `#` will be ignored)

The iTop package contains a file named `params.distrib` that can be used as a template for creating your own parameter file.

Example:

[params.txt](#)

```
# This is a parameter file
#
# If a parameter is given both in the file and in the arguments,
# then the value given as argument is retained
#

# Authentication
auth_user = qwertyuiop
auth_pwd = ded!catedL0gln

# My web service
size_min = 20 # Megabytes
time_limit = 40 # Minutes
```

Make sure that the parameter file is **not** reachable from the web! Either move it to a directory outside of the web server's scope (for example `/etc/itop/` on Linux systems would be a good location) or make a special exclusion rule in your webserver's configuration. Only the process that runs `cron.php` from the command line should be able to access this file.

## Settings

Those settings are configured into the main iTop configuration file:  
“itop-config.php”

| Setting                 | Description   | Default value |
|-------------------------|---|---------------|
| cron_max_execution_time | Duration (seconds) of the page cron.php, must be shorter than php setting max_execution_time and shorter than the 600 web server response timeout                         |               |
| cron_sleep              | Duration (seconds) before cron.php checks again if something must be done   | 2             |
| email_asynchronous      | If set to 1, the emails are sent off line, which requires cron.php to be activated. Exception: some features like the 0 email test utility will force the serialized mode |               |

## Allowed users

Only administrators are allowed to execute cron.php.

## Tuning iTop Performance

In order to enhance the performance of your iTop server, the first point to understand where the bottleneck is. Is it the PHP execution of the pages which is too slow, or is MySQL?

The real tuning of the application's performance requires deep-dive into the actual usage pattern of your iTop instance, however there are few generic points to check:

## Install APC

If it is available on your system, install and enable APC ([Alternative PHP Cache](#)). APC speeds-up PHP applications by caching the result of the PHP parsing as opcodes in shared memory. Moreover, iTop uses APC for caching a lot of its internal structures in memory instead of rebuilding them at each page load.

Installing APC will improve the overall responsiveness of the application, which is especially important for “small” pages.

## Remove un-needed dictionary files

Using iTop in your language is great, but do you really need Turkish, Spanish, Russian and Chinese on your system? If the answer is no, then you can free a good amount of memory that will be used for more productive purposes. Moreover, if you don't have APC installed on your system, all these dictionaries are loaded on each page request. To remove the non-needed dictionaries, the easiest way is to remove the files (they are called `XX.dictionary.YYYY.php`, where `XX` is the language code) from the `datamodels` directory and to run the setup again.

Another way to remove the dictionaries is to edit the iTop configuration file and to comment out (or remove) the lines corresponding to the dictionaries. Be aware however that the list of included dictionaries will be recomputed the next time you run the setup.

For example, keeping just the English, French and Italian translations:

```
'dictionaries' => array (
    '// dictionaries/ru.dictionary.itop.ui.php',
    'dictionaries/fr.dictionary.itop.core.php',
    'dictionaries/it.dictionary.itop.core.php',
    '// 'dictionaries/zh.dictionary.itop.core.php',
    '// 'dictionaries/zh.dictionary.itop.ui.php',
    '// 'dictionaries/es_cr.dictionary.itop.core.php',
    '// 'dictionaries/ja.dictionary.itop.ui.php',
    '// 'dictionaries/es_cr.dictionary.itop.ui.php',
    '// 'dictionaries/hu.dictionary.itop.ui.php',
    '// 'dictionaries/ru.dictionary.itop.core.php',
    '// 'dictionaries/ja.dictionary.itop.core.php',
    'dictionaries/dictionary.itop.core.php',
    'dictionaries/dictionary.itop.ui.php',
    '// 'dictionaries/tr.dictionary.itop.ui.php',
    '// 'dictionaries/de.dictionary.itop.ui.php',
    '// 'dictionaries/de.dictionary.itop.core.php',
    '// 'dictionaries/pt_br.dictionary.itop.core.php',
    '// 'dictionaries/pt_br.dictionary.itop.ui.php',
    '// 'dictionaries/tr.dictionary.itop.core.php',
    'dictionaries/fr.dictionary.itop.ui.php',
    'dictionaries/it.dictionary.itop.ui.php',
    'dictionaries/hu.dictionary.itop.core.php',

    'env-production/authent-ldap/fr.dict.authent-ldap.php',
```

```
'env-production/authent-ldap/it.dict.authent-ldap.php',

'env-production/authent-ldap/en.dict.authent-ldap.php',
    //
'env-production/authent-ldap/pt_br.dict.authent-ldap.php',
    //
'env-production/authent-ldap/de.dict.authent-ldap.php',
    //
'env-production/authent-ldap/es_cr.dict.authent-ldap.php',
    ...
```

## Check MySQL's Key cache

Make sure that MySQL has enough memory for **caching all its indexes** in memory. The important status variables are: `key_read_requests` and `key_reads`.

The actual ratio of `key_reads` / `key_read_requests` should be as low as possible (typically less than 0.1 %). If this ratio is too high, this means that you should increase the amount of memory allocated to cache MySQL's indexes (adjust the variable `key_buffer_size` in `my.cnf` and restart the MySQL server).

Here is what the statistics look like in PhpMyAdmin:

| Key cache              |         |  | <a href="#">Begin</a> |
|------------------------|---------|--|-----------------------|
| Variable               | Value   | Description  |                       |
| Key_blocks_not_flushed | 0       | The number of key blocks in the key cache that have changed but haven't yet been flushed to disk. It used to be known as <code>Not_flushed_key_blocks</code> .   |                       |
| Key_blocks_unused      | 14 k    | The number of unused blocks in the key cache. You can use this value to determine how much of the key cache is in use.   |                       |
| Key_blocks_used        | 125     | The number of used blocks in the key cache. This value is a high-water mark that indicates the maximum number of blocks that have ever been in use at one time.  |                       |
| Key_read_requests      | 138 k   | The number of requests to read a key block from the cache.   |                       |
| Key_reads              | 28      | The number of physical reads of a key block from disk. If <code>Key_reads</code> is big, then your <code>key_buffer_size</code> value is probably too small. The cache miss rate can be calculated as <code>Key_reads/Key_read_requests</code> . |                       |
| Key_write_requests     | 28 k    | The number of requests to write a key block to the cache.  |                       |
| Key_writes             | 26 k    | The number of physical writes of a key block to disk.  |                       |
| Key_buffer_fraction %  | 11.52 % |  |                       |
| Key_write_ratio %      | 93.61 % |  |                       |
| Key_read_ratio %       | 0.62 %  |  |                       |

## Check the Query Cache

Check if the MySQL Query Cache is enabled, with enough memory to cache the results of the most common queries.

Here is what the Query Cache status looks like in PhpMyAdmin:

| Query cache                             |         |   | Begin ▲ |
|---|---------|---|---------|
| Variable                                | Value   | Description   |         |
| Qcache_free_blocks                      | 33      | The number of free memory blocks in query cache.  |         |
| Qcache_free_memory                      | 7,417 k | The amount of free memory for query cache.  |         |
| Qcache_hits                             | 72 k    | The number of cache hits.   |         |
| Qcache_inserts                          | 11 k    | The number of queries added to the cache.   |         |
| Qcache_lowmem_prunes                    | 6       | The number of queries that have been removed from the cache to free up memory for caching new queries. This information can help you tune the query cache size. The query cache uses a least recently used (LRU) strategy to decide which queries to remove from the cache. |         |
| Qcache_not_cached                       | 3,304   | The number of non-cached queries (not cachable, or not cached due to the query_cache_type setting).   |         |
| Qcache_queries_in_cache                 | 5,065   | The number of queries registered in the cache.  |         |
| Qcache_total_blocks                     | 11 k    | The total number of blocks in the query cache.  |         |
| <a href="#">[Flush query cache] [ⓘ]</a> |         |   |         |

# Configuration Parameters

## Overview

Located in the the conf/production directory, the file itop-config.php contains all the configuration parameters of iTop.

This file is created during the setup, and updated upon upgrade.

If you have done some modifications to the settings, please backup this file prior to executing an upgrade, because the file will be generated again by the upgrade program. For example, comments will be lost.

The configuration file contains the following sections:

- MySettings: contains iTop settings. Some settings are not visible in the file generated by the installation program. These are settings for troubleshooting or disabling some features (workaround for complex integrations).
- MyModuleSettings: contains module settings.
- MyModules: lists the PHP files to include. It is a way to disable -temporarily- an installed feature. Depending on your need for language support, you might comment out some lines. This can **speedup significantly the execution of every single iTop page**. Please note that those changes will be lost when upgrading iTop.

## Settings

| Parameter      | Type   | Visible ? | Description   | Default Value   |
|----------------|--------|-----------|---|---|
| access_message | string | Yes       | Message displayed to the users when there is any access restriction | iTop is temporarily frozen, please wait... (the admin team) |

|                       |         |     |   |   |
|-----------------------|---------|-----|---|---|
| access_mode           | integer | Yes | Combination of flags (ACCESS_USER_WRITE, ACCESS_ADMIN_WRITE, or ACCESS_FULL)  | 3   |
| addon_list            | array   | Yes | Automatically populated by the installation process   |   |
| allow_target_creation | bool    | No  | Displays the + button on external keys to create target objects   | 1   |
| allowed_login_types   | string  | Yes | The list (and in which order) of authentication methods that the application allows. The value is a combination of form,cas,basic,external,url                | form,basic,external   |
| apc_cache.enabled     | bool    | Yes | If set, the APC cache is allowed (the PHP extension must also be active)  | 1   |
| apc_cache.query_ttl   | integer | Yes | Time to live set in APC for the prepared queries (seconds - 0 means no timeout)   | 3600  |
| app_env_label         | string  | Yes | Label displayed to describe the current application environment, defaults to the environment name (e.g. "production")   |   |
| app_icon_url          | string  | No  | Hyperlink to redirect the user when clicking on the application icon (in the main window, or login/logoff pages)  | <a href="http://www.combodo.com/itop">http://www.combodo.com/itop</a> |
| app_root_url          | string  | Yes | Root URL used for navigating within the application, or from an email to the application (you can put \$SERVER_NAME\$ as a placeholder for the server's name) |   |
| application_list      | array   | Yes | Automatically populated by the installation process   |   |
| buttons_position      | string  | Yes | Position of the forms buttons: bottom , top , both  | both  |

|                             |         |     |  |                |
|-----------------------------|---------|-----|--|----------------|
| cas_context                 | string  | Yes | The CAS context  |                |
| cas_debug                   | bool    | Yes | Activate the CAS debug   | 0              |
| cas_default_profiles        | string  | Yes | A semi-colon separated list of iTop Profiles to use when creating a new user if no profile is retrieved from CAS           | Portal user    |
| cas_host                    | string  | Yes | The name of the CAS host   |                |
| cas_include_path            | string  | Yes | The path where to find the phpCAS library  | /usr/share/php |
| cas_logout_redirect_service | string  | Yes | The redirect service (URL) to use when logging-out with CAS  |                |
| cas_memberof                | string  | Yes | A semicolon separated list of group names that the user must be member of (works only with SAML - e.g. cas_version⇒ “S1” ) |                |
| cas_port                    | integer | Yes | The port used by the CAS server  | 443            |
| cas_profile_pattern         | string  | Yes | A regular expression pattern to extract the name of the iTop profile from the name of an LDAP/CAS group                    | /^cn=([^,]+),/ |
| cas_server_ca_cert_path     | string  | Yes | The path where to find the certificate of the CA for validating the certificate of the CAS server                          |                |
| cas_update_profiles         | bool    | Yes | Whether or not to update the profiles of an existing user from the CAS information   | 0              |
| cas_user_synchro            | bool    | Yes | Whether or not to synchronize users with CAS/LDAP  | 0              |
| cas_version                 | string  | Yes | The CAS protocol version to use: “1.0” (CAS v1), “2.0” (CAS v2) or “S1” (SAML V1) )  | 2.0            |
| complex_actions_limit       | integer | No  | Display the “actions” menu items that require long computation only if the list of   | 50             |

|                                     |         |     |   |            |
|-------------------------------------|---------|-----|---|------------|
|                                     |         |     | objects is contains less objects than this number (0 means no limit)  |            |
| cron_max_execution_time             | integer | Yes | Duration (seconds) of the page cron.php, must be shorter than php setting max_execution_time and shorter than the web server response timeout                                     | 600        |
| cron_sleep                          | integer | No  | Duration (seconds) before cron.php checks again if something must be done   | 2          |
| csv_file_default_charset            | string  | Yes | Character set used by default for downloading and uploading data as a CSV file. Warning: it is case sensitive (uppercase is preferable).  | ISO-8859-1 |
| csv_import_charsets                 | array   | Yes | An array of character sets names to be added to the ones offered by the CSVImport menu item. Add your own charsets definitions here if the standard list does not fit your needs. |            |
| csv_import_creations_percentage     | integer | No  | Percentage of creations that trigger a confirmation in the CSV import   | 50         |
| csv_import_errors_percentage        | integer | No  | Percentage of errors that trigger a confirmation in the CSV import  | 50         |
| csv_import_history_display          | bool    | Yes | Display the history tab in the import wizard  | false      |
| csv_import_min_object_confirmation  | integer | No  | Minimum number of objects to check for the confirmation percentages   | 3          |
| csv_import_modifications_percentage | integer | No  | Percentage of modifications that trigger a confirmation in the CSV import   | 50         |
| datamodel_list                      | array   | Yes | Automatically populated by the installation process   |            |
| db_character_set                    | string  | Yes | Character set to use for the  | utf8       |

|                             |        |     |   |                 |
|-----------------------------|--------|-----|---|-----------------|
|                             | g      |     | MySQL database  |                 |
| db_collation                | string | Yes | Collation (i.e sort mechanism) to use for the MySQL database  | utf8_unicode_ci |
| db_host                     | string | Yes | Name of the host for the MySQL database server. (e.g. localhost, 192.168.10.234, mydbserver.demo.com, etc.)   |                 |
| db_name                     | string | Yes | Name of the MySQL database  |                 |
| db_pwd                      | string | Yes | Password to connect to the MySQL server   |                 |
| db_subname                  | string | Yes | Prefix of the tables in the MySQL database  |                 |
| db_user                     | string | Yes | user name to connect to the MySQL server  |                 |
| deadline_format             | string | Yes | The format used for displaying "deadline" attributes: any string with the following placeholders: \$date\$, \$difference\$  | \$difference\$  |
| debug_report_spurious_chars | bool   | No  | Report, in the error log, the characters found in the output buffer, echoed by mistake in the loaded modules, and potentially corrupting the output               | 0               |
| default_language            | string | Yes | The default language for the application, used for the login/logout pages. (Selected during the installation)   | EN US           |
| dictionary_list             | array  | Yes | Automatically populated by the installation process   |                 |
| email_asynchronous          | bool   | Yes | If set, the emails are sent offline, which requires cron.php to be activated. Exception: some features like the email test utility will force the serialized mode | 0               |
| email_transport             | string | Yes | Mean to send emails: PHPMail (uses the function   | PHPMail         |

|                                 |         |     |   |                  |
|---------------------------------|---------|-----|---|------------------|
|                                 |         |     | mail()) or SMTP (implements the client protocole)   |                  |
| email_transport_smtp.encryption | string  | No  | tls or ssl (optional)   |                  |
| email_transport_smtp.host       | string  | No  | host name or IP address (optional)  | localhost        |
| email_transport_smtp.password   | string  | No  | Authentication password (optional)  |                  |
| email_transport_smtp.port       | integer | No  | port number (optional)  | 25               |
| email_transport_smtp.username   | string  | No  | Authentication user (optional)  |                  |
| encryption_key                  | string  | Yes | A “salt” key used for encrypting secured fields in the application.   | @iT0pEncr1pti0n! |
| fast_reload_interval            | integer | Yes | The duration (in seconds) between two reloads of a list, if the reload interval is “fast”   | 60               |
| graphviz_path                   | string  | No  | Path to the Graphviz “dot” executable for graphing objects lifecycle  | /usr/bin/dot     |
| link_set_attribute_qualifier    | string  | Yes | Link set from string: attribute qualifier (encloses both the attcode and the value)   | '                |
| link_set_attribute_separator    | string  | Yes | Link set from string: attribute separator   | ;                |
| link_set_item_separator         | string  | Yes | Link set from string: line separator  |                  |
| link_set_value_separator        | string  | Yes | Link set from string: value separator (between the attcode and the value itself)  | :                |
| log_global                      | bool    | Yes | If set to 1, then the log is active. Which event is logged and where will depend on the following log_.... settings                   | 1                |
| log_issue                       | bool    | Yes | If set to 1, then internal errors (or some usage errors) will be traced both into <iTopRoot>/error.log (destination file could not be | 1                |

|                            |         |     |   |   |
|----------------------------|---------|-----|---|---|
|                            |         |     | changed) and the DB (OQL: SELECT EventIssue)  |   |
| log_notification           | bool    | Yes | If set to 1, then notifications sent by the means of the Trigger/Actions will be recorded into the DB (OQL: SELECT EventNotification) | 1   |
| log_usage                  | bool    | No  | Log the usage of the application (i.e. the date/time and the user name of each login)   | 0   |
| log_web_service            | bool    | Yes | If set to 1, then usage of the SOAP service(s) will be recorded into the DB (OQL: SELECT EventWebService)                             | 1   |
| max_combo_length           | integer | No  | The maximum number of elements in a drop-down list. If more than an autocomplete will be used   | 50  |
| max_display_limit          | integer | Yes | The maximum number of items that a list can display at once, before changing to a paginated list                                      | 15  |
| min_autocomplete_chars     | integer | No  | The minimum number of characters to type in order to trigger the "autocomplete" behavior  | 3   |
| min_display_limit          | integer | Yes | The number of items to display when a list is bigger than "max_display_limit"   | 10  |
| module_settings            | array   | Yes | Automatically populated by the installation process   |   |
| online_help                | string  | Yes | Hyperlink to the online-help web page   | <a href="http://www.combodo.com/itop-help">http://www.combodo.com/itop-help</a> |
| php_path                   | string  | Yes | Path to the php executable in CLI mode  | php   |
| secure_connection_required | bool    | Yes | Whether or not the application is allowed to run on a non-secure (i.e. non-HTTPS) connection  | 0   |
| session_name               | string  | Yes | The name of the cookie used   | iTop  |

|                            |         |     |  |                            |
|----------------------------|---------|-----|--|----------------------------|
|                            | g       |     | to store the PHP session id  |                            |
| shortcut_actions           | string  | Yes | Actions that are available as direct buttons next to the "Actions" menu  | UI:Menu:Modify,UI:Menu:New |
| skip_check_ext_keys        | bool    | No  | Disable external key check when checking the value of attributes   | 0                          |
| skip_check_to_write        | bool    | No  | Disable data format and integrity checks to boost up data load (insert or update)  | 0                          |
| skip_strong_security       | bool    | No  | Disable strong security - TEMPORARY: this flag should be removed when we are more confident in the recent change in security   | 1                          |
| source_dir                 | string  | Yes | Source directory for the datamodel files. (which gets compiled to env-production).   |                            |
| standard_reload_interval   | integer | Yes | The duration (in seconds) between two reloads of a list, if the reload interval is "standard"  | 300                        |
| synchro_prevent_delete_all | bool    | No  | Stop the synchro if all the replicas of a data source become obsolete at the same time.  | 1                          |
| synchro_trace              | string  | Yes | Synchronization details: none, display, save (includes 'display')  | none                       |
| timezone                   | string  | Yes | Timezone (reference: <a href="http://php.net/manual/en/timezones.php">http://php.net/manual/en/timezones.php</a> ). If empty, it will be left unchanged and MUST be explicitly configured in PHP | Europe/Paris               |
| webservice_list            | array   | Yes | Automatically populated by the installation process  |                            |

# Object Query Language Reference

*Version: iTop 1.2, iTop 2.0*

## Overview

The Object Query Language (in short: OQL) is the language used by iTop for defining queries that retrieve persisted objects.

OQL queries are used in many places in iTop (audit rules, notifications, etc.) to define or configure sets of objects.

The goal of the OQL language is to hide the complexity of the actual SQL schema of the data model while leveraging its definition for providing a secure (access rights are enforced by the queries), powerful (complex conditions can be expressed in OQL), efficient (one OQL query is translated into exactly one SQL query), object-aware (class inheritance is respected) and simple syntax.

The OQL syntax is similar to the syntax of SQL, the OQL grammar is basically a subset of the SQL grammar.

As of now, only SELECT statements have been implemented.

Such statements do return two kinds of data set:

- Either a list of objects of the selected class.
- Or a list of objects and related objects (several classes of objects per row)

The OQL language is case sensitive and all keywords are to be written in UPPERCASE

## Content of this book

- [Example: the bookstore](#)
- [OQL Statements Syntax](#)
- [OQL Grammar](#)
- [Known Issues and Limitations](#)

## Example: the Bookstore

The most basic query is simply a SELECT with just a class name:

```
SELECT
    Book
```

This query returns any Book existing in the database. As you can notice, there is no need to specify the expected columns, as we would do in a SQL SELECT statement, because OQL queries always return complete objects.

## Joining classes together

I would like to list all books written by someone whose name starts with 'Camus'

```
SELECT
    Book
    JOIN Artist ON Book.written_by = Artist.id
    WHERE Artist.name LIKE 'Camus%'
```

Note that there is no need to specify whether the JOIN is an INNER JOIN, or LEFT JOIN. This is well-known in the data model. The OQL engine will in turn create a SQL query based on the relevant option, but we do not want to care about it, do we?

Now, you may consider that the name of the author of a book is of importance. This is the case if it should be displayed anytime you will list a set of books, or if it is an important key to search for.

Then you have the option to change the data model, and define the name of the author as an external field. Such an external field would be defined by the external key `written_by` and the target attribute name. Let's define this new field as `writer_name`.

The query could then be simplified to:

```
SELECT
    Book
    WHERE Book.writer_name LIKE 'Camus%'
```

The join will actually be performed by the underlying SQL query, but this is completely transparent to the OQL. It will happen each and every time

those objects are queried so that the attribute `writer_name` becomes part of the properties of a book - though it will be read-only.

It is also possible to retrieve the authors and their books in one query. For example let's retrieve, for each author, a list of the books written in 2001 and after. This can be achieved using the query:

```
SELECT
    Artist, Book FROM
    Book
    JOIN Artist ON Book.written_by = Artist.id
    WHERE Book.issued > '2001-01-01'
```

The data set resulting from this query will have two columns for each “row” of the set: an Artist and a Book.

## Classes Inheritance

Now, as this is a modern bookstore, several types of media are available: Audio, Video, Book. All of them have been declared as classes derived from Item in the data model, as shown below:

- Item
  - Audio
  - Video
  - Book

Since Audio, Video and Book objects are also Items (thanks to the class inheritance), all of them can be retrieved in one single query. For example let's retrieve all Items not being produced by in France:

```
SELECT
    Item
    JOIN Producer ON Item.produced_by = Producer.id
    WHERE Producer.country != 'France'
```

This query will return books as well, because a Book is an Item... that's due to classes inheritance: a Book inherits from Item, or we can say that a Book is a specialized item.

## Data Hierarchies

Let's imagine that a book is characterized by one Topic.

Topics are organized as a hierarchy of objects. Therefore a Topic can have a parent Topic:

Let's consider the following hierarchy of Topics:

- Art
- History
  - Archaeology
    - Civilizations
    - Archeologists
    - Egyptology
  - History of science
  - War

Books can be attached at any level.

Let's list all books about History:

```
SELECT
  Book
  JOIN Topic AS NodeTopic ON Book.topic_id = NodeTopic.id
  JOIN Topic AS RootTopic ON NodeTopic.parent_id BELOW RootTopic.id
  WHERE RootTopic.name = ' History'
```

This query will return all books related to the Topic 'History', 'Archaeology', 'Civilizations', 'Egyptology', etc. but not 'Arts'. All the topics being in the tree under 'History' will be considered, whatever the depth of the hierarchy.

The following query will return only the Books which Topics is *strictly* below 'Archaeology', e.g. only 'Civilizations', 'Archeologists' and 'Egyptology':

```
SELECT
  Book
  JOIN Topic AS NodeTopic ON Book.topic_id = NodeTopic.id
  JOIN Topic AS RootTopic ON NodeTopic.parent_id BELOW STRICT
  RootTopic.id
  WHERE RootTopic.name = ' Archaeology'
```

## OQL Statements Syntax

There is currently only one type of statement in OQL: SELECT

The syntax of the SELECT statement is the following:

SELECT

```
[output_specification FROM]
class_reference
[class_joined]
[WHERE expression]
```

- `output_specification` indicates the class of objects that you want to retrieve, when omitted, it will default to `class_reference`. When it is made of a list of class aliases, then the first one determines what is the primary class expected.
- `class_reference` indicates the class of objects that you want to query.
- `class_joined` indicates a series of classes that you want to join, in order to restrict the set of selected objects (remember, it makes no sense to mention columns).
- `where_condition` is an expression, very close to what could be found in an SQL SELECT statement.

## output\_specification

```
class_alias [, output_specification]
```

- `class_alias` indicates an alias that is declared in the `class_reference` or `class_joined` clauses.

## class\_reference

```
class_name [AS class_alias]
```

- `class_name` indicates the class of objects that you want to select.
- `class_alias` indicates an alias that will be used to refer to the given class, in the expressions found into the WHERE clause.

## class\_name or class\_alias

```
name | `name`
```

Backticks must be used in the following circumstances:

- the name of the class is in conflict with a reserved word (example: JOIN...),
- the name of the class contains undesirable characters.

## class\_joined

```
JOIN class_reference
ON class_left.external_key join_operator class_right.id
```

- `class_reference` refers either to the class on the left of the join or the right... depending on the data model and the given external key.
- `class_right.id` has to be specified though it cannot be something else: it refers to the object that is pointed to by the other one. `Class_right` is an alias if any has been given.
- `class_left.external_key` indicates which attribute from which class should be pointing to `class_right.id`. In most cases, the external key attributes could be guessed, but the reference has to be specified explicitly anyway.
- `join_operator` can be either `=`, `BELOW`, `BELOW STRICT`, `ABOVE` or `ABOVE STRICT`. `BELOW` and `ABOVE` can be used only in conjunction with attributes of type `AttributeHierarchicalKey`.

## expression

```
literal
| function
| attribute
| expression operator expression
| (expression)
```

- `literal` is either a string (enclosed with single or double quotes) or a number (only integers are supported).
- `function` is one of the verbs listed above, the arguments are a coma separated list of expressions
- `attribute` is a reference to an object property as defined in the data model, in the form `class_ref.attribute_code` – use of backticks is necessary to solve conflict with reserved words or white characters.
- `operator` is any of the binary operators listed below.
- `expression` may be enclosed in parenthesis to cope with operators precedence.

## Binary operators

| Operator | Description                    |
|----------|--------------------------------|
| AND      | Logical AND                    |
| /        | Division operator              |
| =        | Equality operator              |
| >=       | Greater than or equal operator |
| >        | Greater than operator          |
| <=       | Less than or equal operator    |

|          |                                     |
|----------|-------------------------------------|
| <        | Less than operator                  |
| -        | Substraction operator               |
| !=, <>   | Non-equality operator               |
| LIKE     | Simple pattern matching             |
| NOT LIKE | Negation of simple pattern matching |
| IN       | List operator                       |
| NOT IN   | Negation of list operator           |

## function

verb(expression[, expression [, expression...][...] ...])

- verb is one the functions listed below

All functions are actually mapped to their equivalent in SQL. In other words, the same functions will be used in the resulting SQL queries that will be finally executed.

Therefore, the specification of those functions (number and type of arguments, returned values) is similar to the specification of the underlying database server. Any limitation or side-effect, will be related to the version of the database engine.

The hyperlinks provided hereafter will direct you to the reference documentation of `mySQL 5.0`, which is the standard recommended database engine (used for qualification of the `OQL` processor).

- Function names are case-sensitive. They have to be uppercase in the current `OQL` implementation, though `MySQL` is less restrictive.
- So far, no synonym has been implemented (There is just one single name for a given function ; example: `OQL` implements `DAY` whereas `mySQL` implements `DAY` and `DAYOFMONTH` as an alias to the same function)

| Function name                   | Description                        | Examples                                    |
|---------------------------------|------------------------------------|---|
| <a href="#"><u>IF</u></a>       | If/else construct                  | IF(a=b, 'equals', 'differs')                |
| <a href="#"><u>ELT</u></a>      | Return string at index number      | ELT(index, 'string1', 'string2', 'string3') |
| <a href="#"><u>COALESCE</u></a> | Return the first non-NULL argument | COLAESCCE(field1, field2, 'Undefined')      |
| <a href="#"><u>ISNULL</u></a>   |                                    | ISNULL(field1)                              |
| <a href="#"><u>CONCAT</u></a>   | Return concatenated string         | CONCAT(firstname, ' ', lastname)            |

|                                     |  |   |
|-------------------------------------|--|---|
| <a href="#"><u>SUBSTR</u></a>       | Return the substring as specified  | <code>SUBSTR('abcdef', 2, 3)</code>                         |
| <a href="#"><u>TRIM</u></a>         | Remove leading and trailing spaces   | <code>TRIM(' blah ')</code>                                 |
| <a href="#"><u>DATE</u></a>         | Extract the date part of a date or datetime expression                               | <code>DATE()</code>   |
| <a href="#"><u>DATE_FORMAT</u></a>  | Format date as specified   | <code>DATE_FORMAT('2009-10-04 22:23:00', '%W %M %Y')</code> |
| <a href="#"><u>CURRENT_DATE</u></a> | Return the current date  | <code>CURRENT_DATE()</code>                                 |
| <a href="#"><u>NOW</u></a>          | Return the current date and time   | <code>NOW()</code>  |
| <a href="#"><u>TIME</u></a>         | Extract the time portion of the expression passed                                    | <code>TIME()</code>   |
| <a href="#"><u>TO_DAYS</u></a>      | Return the date argument converted to days   | <code>TO_DAYS('2009-05-01')</code>                          |
| <a href="#"><u>FROM_DAYS</u></a>    | Convert a day number to a date   | <code>FROM_DAYS(12345)</code>                               |
| <a href="#"><u>YEAR</u></a>         | Return the year from the date passed   | <code>YEAR(DATE())</code>                                   |
| <a href="#"><u>MONTH</u></a>        | Return the month from the date passed  | <code>MONTH(DATE())</code>                                  |
| <a href="#"><u>DAY</u></a>          | Return the day of the month (0-31)   | <code>DAY(DATE())</code>                                    |
| <a href="#"><u>DATE_ADD</u></a>     | Add time values (intervals) to a date value. See allowed interval units below        | <code>DATE_ADD(NOW(), INTERVAL 1 HOUR)</code>               |
| <a href="#"><u>DATE_SUB</u></a>     | Subtract time values (intervals) from a date value. See allowed interval units below | <code>DATE_SUB(NOW(), INTERVAL 5 MINUTE)</code>             |
| <a href="#"><u>ROUND</u></a>        | Round the argument   | <code>ROUND(12.356, 2)</code>                               |
| <a href="#"><u>FLOOR</u></a>        | Return the largest integer value not greater than the argument                       | <code>FLOOR(12.356)</code>                                  |
| <a href="#"><u>INET_ATON</u></a>    | Return the numeric value of an IP address  | <code>INET_ATON('15.15.121.12')</code>                      |
| <a href="#"><u>INET_NTOA</u></a>    | Return the IP address from a numeric value   | <code>INET_NTOA(1231654)</code>                             |

The list of time interval units currently supported by the functions `DATE_ADD` and `DATE_SUB`, is a subset of the values allowed in MySQL.

OQL does accept:

#### **Time interval units**

YEAR

MONTH

DAY  
HOUR  
MINUTE  
SECOND

## OQL Grammar

The OQL grammar can be described as following using the BNF syntax:

```
select-query ::= SELECT class-reference [class-joined] [WHERE  
expression]  
class-reference ::= name [AS name]  
class-joined ::= JOIN class-reference ON name.name = name.id  
name ::= string | `string`  
expression ::=  
scalar  
| expression operator expression  
| (expression)  
scalar ::= number | 'string' | "string" | column  
operator ::= AND | OR | = | <> | != | > | >= | < | <= | LIKE | NOT LIKE  
column ::= name | name.name
```

## Known issues and limitations

Negative numbers are not accepted. The workaround is to specify negative values `-n` as the expression `(0 - n)`.

The operator `!` is not accepted.

`true` and `false` are not accepted.

`null` is not accepted.

The equivalent of `!ISNULL(start_date)` can be expressed `ISNULL(start_date) != 1`

## Examples

The following examples are useful queries for a specific version of iTop.

## How to retrieve the members of a Team

- Purpose: retrieve all the members of a given Team (where the Team is specified by its name).
- Compatibility: iTop 2.0

```
SELECT Person AS p
  JOIN lnkPersonToTeam AS ll ON ll.person_id=p.id
  JOIN Team AS t ON ll.team_id=t.id
 WHERE t.name = "My team"
```

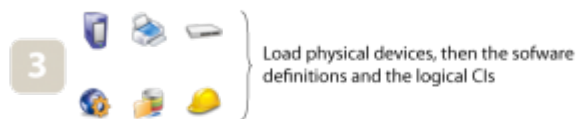
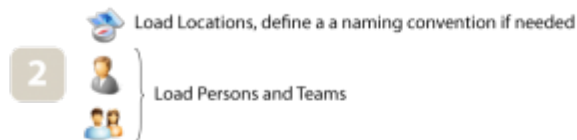
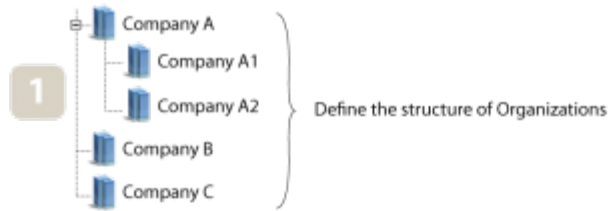
## iTop implementation guide

The purpose of this document is to describe, step by step, which iTop objects have to be created to implement iTop for your organization. For instance, in order to create a User Request ticket, you need to make sure that the caller for this request exists, that there is at least one contract documented for this customer defining the services delivered to this customer, etc. This document explains the order to be followed for creating the objects in iTop.

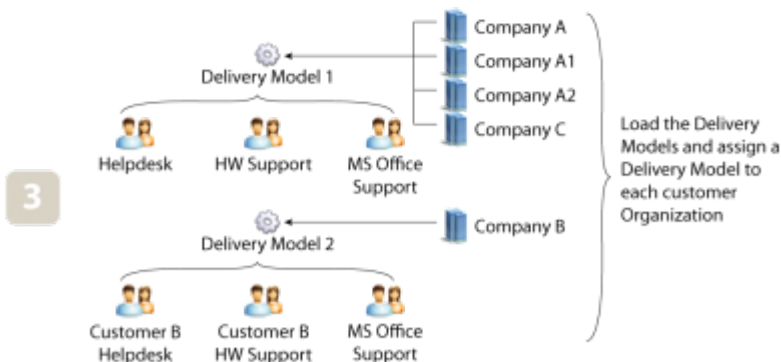
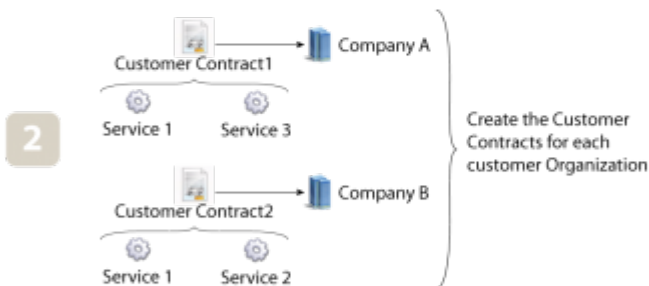
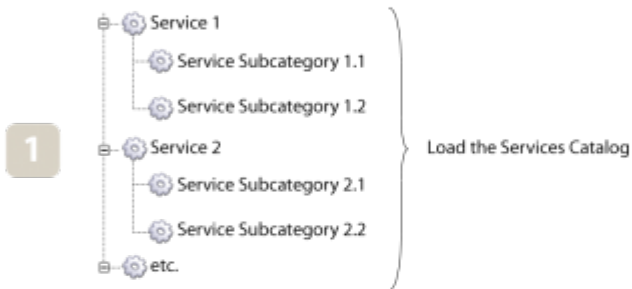
For creating a *clean* production environment it is better to start from an iTop instance installed *without* the sample data. However you can also install another instance of iTop *with* the sample data to have a quick look at the basic data needed to produce a fully functional instance.

The following schema summarizes the on-boarding process:

## Configuration Management



## Service Management



This document does not describe in details how to use all the features of iTop. For more details about the usage of iTop, refer to the “[iTop User Manual](#)” .

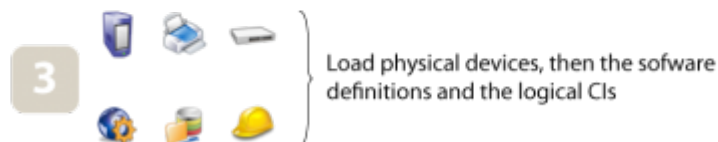
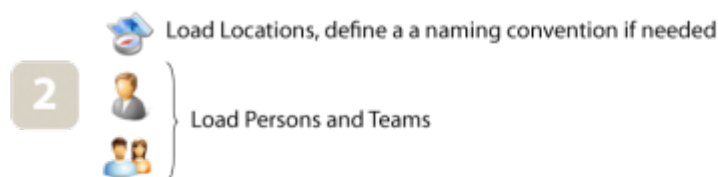
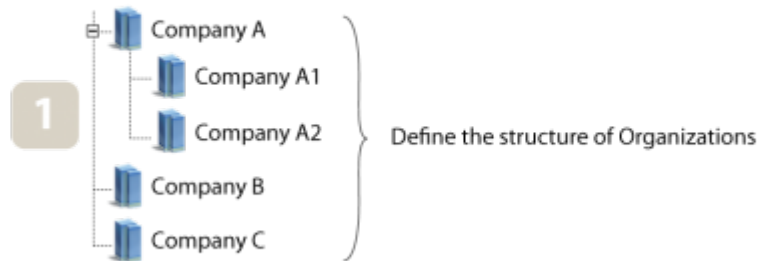
## Creating new objects in iTop

There are several ways to create new objects in iTop, depending on the type of object and whether you want to create the objects interactively, one by one, or in bulk mode. The steps to create each class of object from the menu is explained in details in the [Data Model Documentation](#), but there are two other ways that can be convenient for an administrator:

- **From CSV Import:** if you wish to create many instances of the same class of objects, it is often easier to import them from an existing data set. The CSV Import tool, under the “Data Administration” menu, is designed for this. Prepare your data in CSV format: as a text file, with one object per line and values separated by a fixed character (semicolon, comma or tab). Then let the CSV import wizard guide you into loading the file into iTop.
- **From the Universal Search:** to create a new object of any class, use the menu “Universal Search” in the “Admin Tools” section. Perform a search for objects of this class, then use the “Actions / New...” popup menu to create a new instance of the class.

## Configuration Management

## Configuration Management



## Creating organizations

When planning a deployment of iTop, the first decision to be made is about the structure of Organizations. In iTop, Organizations are used for two main purposes: the description of customers and providers entities and the partitioning of the data, from the security point of view. Almost all the objects loaded in iTop have a relation with an Organization, therefore it is important to create a proper structure of Organizations before loading other objects into iTop.

## Understanding customers and providers

In iTop, there is nothing such as a “customer” or a “provider”, there are only Organizations. Like in real life, whether an Organization is a customer or provider depends on the point of view. For example the Organization “Company A” can be a customer of “Company B” and at the same time a provider for “Company C”. The customer/provider relations in iTop are represented using Contracts. “Company A” is a customer of “Company B” if there is a Customer Contract with “Company B” as the provider and “Company A” as the customer.

## **What is the difference between Customer Contracts and Provider Contracts?**

A Provider Contract is a slightly simplified version of the Customer Contract, with two limitations:

- A Provider Contract is not related to any Service from the service catalog.
- The Service Level Agreement is documented as a free text field on Provider Contracts and therefore cannot be used for automated computations in iTop.

Provider Contracts are useful for documenting contracts with third party suppliers (called underpinning contracts in the ITIL terminology), for which no tickets will be tracked in iTop.

You can of course use Customer Contracts for describing the contractual relation with a third party supplier, but this means that you have to also document in iTop the service catalog of this supplier.

## **Organizations and access rights**

Apart from the customer/provider relations, another reason to create several Organizations in iTop is to restrict access to some areas of the managed data.

In iTop the rights to “read” (or display) objects from the database is defined on a per Organization basis. Each user is given (in the definition of her/his account) the rights to access a set of Organizations.

Organizations can be structured as a hierarchy. When this is the case, the access rights are inherited from the “Parent” Organization to the “Child” Organizations. For example, if “Company A” has two child Organizations: “Company A1” and “Company A2”, then if a user has the rights to access the objects in “Company A”, she/he will also be allowed to access the objects in “Company A1” and “Company A2”. On the other hand, a user who is allowed to access only “Company A1” will be allowed to access neither the objects in “Company A” nor those in “Company A2”.

The rights to “write” (i.e. create, modify or delete) objects are defined only by the profile(s) assigned to the user account. For example the profile “Support Agent” gives the rights to create or modify User Request tickets (but not to delete them).

This means that a user has the same access rights over all Organizations that she/he is allowed to access.

For example, in the current version of iTop, a user cannot have the rights to access the data of the Organizations “Company A” and “Company B” and the rights to create Servers only in “Organisation B”. If she/he is given the rights to create Servers, this will apply to both “Company A” and “Company B”.

## Creating Locations

The Locations are very useful for grouping object by geography. Even if the location attribute is not a mandatory field when you create a CI in the CMDB, it is strongly recommended to create Locations beforehand and then to track the locations of all CIs.

Carefully plan the creation of the Locations. Locations are difficult to identify (there is no commonly accepted unique identifier for a Location), if your company does not have one already, you may want to put in place a *naming convention* in order to avoid duplicate Locations in the CMDB.

## Shared Locations

In Enterprise environments, even though the split of roles and responsibilities are in favor of creating several sub Organizations, it is often needed to have “shared” locations among several Organizations to document “co-locations”. iTop does not provide – in its standard version – a way to actually “share” objects between Organizations. However, the Locations are “inherited” from parent Organizations to child Organizations in the same manner as the access rights. This means that a Person, a Server or a Network Device belonging to “Company A2” can be located on a Location owned by “Company A”.

## Creating Persons

The Persons are very important in iTop as they are used to define all the contacts and their responsibilities. A Person belongs to one and only one organization. A Person can be a member of one or more Team(s), and therefore should be created before trying to setup Teams.

Also, each user record is linked to a Person object. Therefore Persons must be created before loading user accounts into iTop. The user record

defines the access rights (and identification method), whereas the Person object defines the information about the contact: name, location, email address, telephone...

## **Creating Teams**

The teams are linked to several types of object, like contracts or tickets, in order to define responsibilities. Teams are also used as “workgroups” for assigning tickets. Teams used for assigning tickets must also have at least one member (the agent to assign the ticket to). The attribute “Role” on the link between a Team and a Person is not mandatory, so you can leave it empty, but it is useful to define the role of the Person in the Team (Team Leader, Manager...).

## **Devices and Software CIs**

Once the structure of the Organizations, the Locations and the contacts (Teams and Persons) have been loaded, you can start to populate the CMDB.

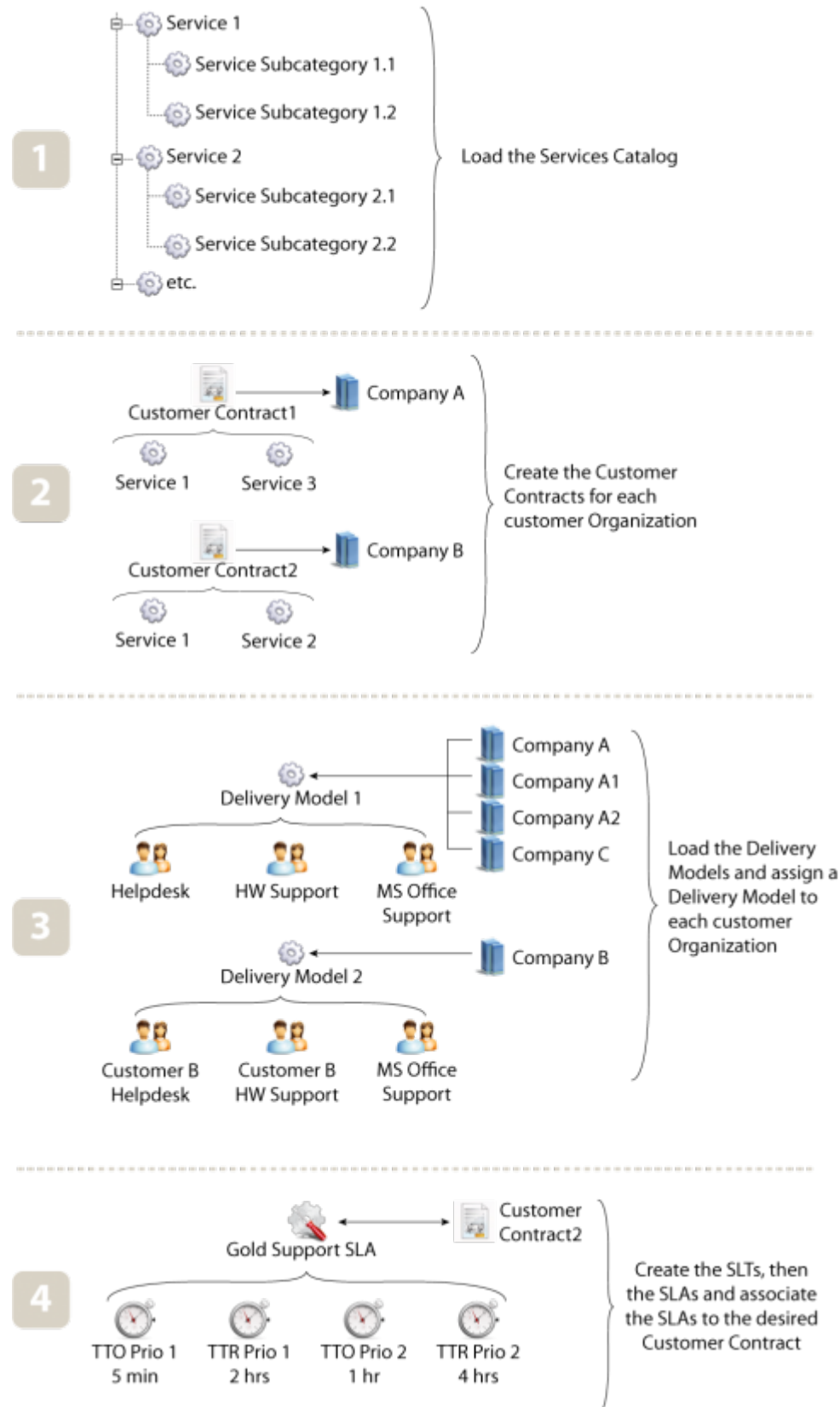
Since the software instances depend on the software types defined in the software catalog and are documented as installed on a particular host, you should start by documenting:

- The physical infrastructure: Servers, Network Devices, PCs, etc...
- The Software catalog, by creating the needed type of “Software” objects

## **Service Management**

Before managing tickets in iTop, the *services catalog*, the *Delivery Models* and the *contracts* must be defined.

## Service Management



## Services Catalog

The “Services Catalog” is the list of Services that are available from a given provider Organization. The Services Catalog is documented in iTop by creating Service objects, assigned to the given Organization (considered as the provider of the service). Services are organized in a two-level hierarchy, through the two classes of objects: **Service** and **Service Subcategory**. Create the top level Services before loading sub categories.

An optional third level “Service Family” can be used to group Services together, but this level is not used by the ticketing in iTop.

Once the service catalog (Services and Service Subcategories) is defined, create the Customer Contracts that will link each “customer” Organization to its “providers” by creating one Customer Contract per couple of provider/customer and linking the appropriate Services to the contract.

## Delivery Model

The Delivery Model is the object that defines which Team works for which customer. You can use a Delivery Model object to group together all the “support teams” for a given set of Services, or the support Teams dedicated to a particular customer. Each customer Organization must be assigned one, and only one, Delivery Model.

In the standard iTop 2.0 data model, there is no link between Teams and Services. The only limitation when assigning a ticket to a Team is that the Team must be part of the Delivery Model assigned to the Organization which is the customer of the ticket.

For example, if you have the following Teams:

- **Helpdesk team:** a Team that processes all helpdesk requests/calls.
- **MS Office Support Team:** a Team that processes all support requests about MS Office.
- **Hardware Support Team:** a Team that handles hardware calls (Replacements, new hardware orders)
- **Network Support Team:** a Team that handles network related issues
- **Customer B Helpdesk Team:** a helpdesk team dedicated to Customer B
- **Customer B Hardware Team:** a Team handling hardware calls for Customer B

You can then build two different Delivery Models:

- “Delivery Model 1” composed of:
  - Helpdesk Team
  - MS Office Support Team
  - Network Support Team
- “Delivery Model 2” composed of:
  - Customer B Helpdesk Team
  - Customer B Hardware Team
  - MS Office Team

The “Delivery Model 1” will be used by the Organizations “Customer A” , “Customer A1” , “Customer A2” and “Customer C” ; whereas “Delivery Model 2” will be used by “Customer B” .

## Service Level Agreements and Targets

The definition of Service Level Agreements (SLAs) and Service Level Targets (SLTs) are not mandatory to manage tickets in iTop, but without them iTop can neither compute deadlines for processing a ticket, nor escalate the ticket automatically.

In order to compute whether or not the expected Service Level Agreements are respected, iTop introduces two possible types of metrics called SLTs (Service Level Targets):

- **TTO (Time To Own):** the time between the creation of a ticket and its assignment to an Agent.
- **TTR (Time To Resolve):** the time between the creation of a ticket and its resolution (i.e. measured when the ticket enters the state “resolved”)

A SLT defines a duration associated with:

- A **metric**: either TTO or TTR
- A **type of ticket** (incident or user request)
- A **priority** (since tickets with higher priority should generally be processed more quickly)

A SLA is simply defined as a named group of SLTs (for example to distinguish between “Gold” and “Silver” service levels).

The definition of SLAs/SLTs have two effects in iTop:

- Notifications can be defined for any percentage of the “threshold” associated with one of the metrics (for example one can configure notifications to send an email to the agent working on a ticket when 50% of the Time To Resolve is reached and to the team leader when reaching 75%).

- When 100% of a metric is reached, the ticket is automatically set to a special “escalation” state (there are two specific states defined in the tickets’ life-cycle: Escalation TTO and Escalation TTR). Entering such a state can also be used to trigger specific notifications.

For example, one can define the following service level matrix:

| <b>Incidents – Priority<br/>High</b> | <b>Incidents – Priority<br/>Medium</b> | <b>Requests – Priority<br/>High</b> | <b>Requests – Priority<br/>Medium</b> |
|--------------------------------------|--|-------------------------------------|---------------------------------------|
| Time To Own: 5 min                   | Time To Own: 30 min                    | Time To Own: 30 min                 | Time To Own: 4 hours                  |
| Time To Resolve: 1 hour              | Time To Resolve: 4 hours               | Time To Resolve: 4 hours            | Time To Resolve: n/a                  |

This would lead to creating 4 SLTs, one for each row of the table. These 4 SLTs can be grouped under one SLA named “Gold Service Level”.

Finally SLAs can be associated to Customer Contracts in order to define the applicable metrics for the contract.

Your iTop instance is now ready to run. You may have a look at the [Configuration of Notifications](#) to setup email notifications for the tickets.