TechBridge Consultancy Services LLP

IT Service Management

tbITSM v.1.0 Admin Guide

v.1.0. |2021

Disclaimer

Company Information

URL: *http://www.tech-bridge.biz/*
Technical Inquiries: *sales@tech-bridge.biz*

Contact Information

326, Tower B3, Spaze iTech Park,
Sector-49, Sohna Road
Gurgoan-122018, Haryana

This documentation explains how to:

1. Navigate around tbITSM v.1.0.
2. Perform basics tasks such as searching for objects, creating or modifying objects
3. Customize the tbITSM v.1.0. user interface (lists, shortcuts, dashboards).

Even though the types of objects managed in tbITSM v.1.0. depend on the modules installed during setup, the general principles described in this document do not vary.

About tbITSM

This document describes release 1.0 of tbITSM. It is a robust web application that will help you to better support your IT. Development of tbITSM v.1.0. started in order to publish on the internet a completely open standard solution that would help enterprise to drive ITIL v3 best practices implementation.

At the early beginning of the project, the development team was focus on building the most complete CMDB (Configuration Management Data Base). One key objective was to make it as flexible as possible in order to allow administrator to add and remove configuration items from the data model and manage as many relationships as they want. The development team also designed a powerful state machine that allows defining life cycle for whatever configuration items in the CMDB.

Realizing that all concepts developed within the CMDB can be applied to all other ITIL best practices, the tbITSM v.1.0. decided to extend them to Incident Management, Change Management, Problem Management, Service Configuration Management, Service Request Management and Service Level Management modules. Then tbITSM v.1.0. became an IT operational portal that helps all IT management team to support their environment by:

- Documenting IT infrastructures and their relationships (servers, application, network…)
- Documenting all users service calls.
- Documenting IT incident and planned outages, as well as a known error database.
- Documenting all IT services and contracts with external providers.

tbITSM v.1.0. application can be used by different type of profiles:

- Help Desk
- IT support engineers (1st level, 2nd level, 3rd level …)
- IT service managers
- IT managers

tbITSM v.1.0. application is relying on Apache, MySQL and PHP, so it can run on whatever operating system supporting those applications. It had been tested already on Windows, Linux Debian and Red hat. As it is a web-based application you don’t need to install client on user PC. A simple web browser is enough to use it.

Features

Notifications:

This version introduces the capability to generate fully customizable email notifications. This is achieved via two new type of objects: triggers and actions. Triggers define when a particular notification must be sent. Actions define the recipients of the notification as well as the content of the email message.

Have a look at the "Notifications" menu item (admins only) for how to configure the notifications.

Note that the notifications shipped by default with the application have been set to 'disabled' so that no email get sent out in case you load the sample data that contains several tickets. To enable them go to the "Admin" menu "Notifications", then in the tab "Actions" open the action that you want to enable and modify its status to "production".

Documents:

The "document" object now contains a field to upload binary documents. The application automatically keeps track of the history of the document (i.e., each version is recorded in the database).

Such documents are stored as "blobs" in the database. Some type of documents can be displayed online directly (images, HTML, PDF...) some others can only be opened in a new window, or saved to the user's disk.

Truncated lists:

When the result of a query returns a long (limit is configurable) list of objects, the display is truncated to speed up the whole display, with a link to open the full list.

Contents

Glossary

S.No.	TERM	DEFINITION
1	**Acceptance**	Formal agreement that an IT Service, Process, Plan, or other Deliverable is complete, accurate, Reliable and meets its specified Requirements. Acceptance is usually preceded by Evaluation or Testing and is often required before proceeding to the next stage of a Project or Process
2	**Access Management**	The Process responsible for allowing Users to make use of IT Services, data, or other Assets. Access Management helps to protect the Confidentiality, Integrity and Availability of Assets by ensuring that only authorized Users are able to access or modify the Assets. Access Management is sometimes referred to as Rights Management or Identity Management.
3	**Account Manager**	A Role that is very similar to Business Relationship Manager, but includes more commercial aspects. Most commonly used when dealing with External Customers.
4	**Accounting**	The Process responsible for identifying actual Costs of delivering IT Services, comparing these with budgeted costs, and managing variance from the Budget
5	**Accredited**	Officially authorized to carry out a Role. For example, an Accredited body may be authorized to provide training or to conduct Audits.
6	**Agreed Service Time**	A synonym for Service Hours, commonly used in formal calculations of Availability
7	**Alert**	A warning that a threshold has been reached, something has changed, or a Failure has occurred. Alerts are often created and managed by System Management tools and are managed by the Event Management Process.
8	**Application**	Software that provides Functions that are required by an IT Service. Each Application may be part of more than one IT Services. An application runs on one or more Servers or Clients.
9	**Application Management**	The Function responsible for managing Applications throughout their Lifecycle.
10	**Application Service Provider (ASP)**	An External Service Provider that provides IT Services using Applications running at the Service Provider's premises. Users access the Applications by network connections to the Service Provider.
11	**Application Sizing**	The Activity responsible for understanding the Resource Requirements needed to support a new Application, or a major Change to an existing Application. Application Sizing helps to ensure that the IT Service can meet its agreed Service Level Targets for Capacity and Performance.
12	**Architecture**	The structure of a System or IT Service, including the Relationships of Components to each other and to the environment they are in. Architecture also includes the Standards and Guidelines which guide the design and evolution of the System.
13	**Assessment**	Inspection and analysis to check whether a Standard or set of Guidelines is being followed, that Records are accurate, or that Efficiency and Effectiveness targets are being met.

14	**Asset**	Any Resource or Capability. Assets of a Service Provider include anything that could contribute to the delivery of a Service. Assets can be one of the following types: Management, Organization, Process, Knowledge, People, Information, Applications, Infrastructure, and Financial Capital
15	**Asset Management**	Asset Management is the Process responsible for tracking and reporting the value and ownership of financial Assets throughout their Lifecycle. Asset Management is part of an overall Service Asset and Configuration Management Process
16	**Attribute**	A piece of information about a Configuration Item. Examples are name, location, Version number, and Cost. Attributes of CIs are recorded in the Configuration Management Database (CMDB)
17	**Audit**	Formal inspection and verification to check whether a Standard or set of Guidelines is being followed, that Records are accurate, or that Efficiency and Effectiveness targets are being met. An Audit may be carried out by internal or external groups
18	**Availability**	Ability of a Configuration Item or IT Service to perform its agreed Function when required. Availability is determined by Reliability, Maintainability, Serviceability, Performance, and Security. Availability is usually calculated as a percentage. This calculation is often based on Agreed Service Time and Downtime. It is Best Practice to calculate Availability using measurements of the Business output of the IT Service.
19	**Availability Management**	The Process responsible for defining, analyzing, Planning, measuring and improving all aspects of the Availability of IT Services. Availability Management is responsible for ensuring that all IT Infrastructure, Processes, Tools, Roles etc. are appropriate for the agreed Service Level Targets for Availability
20	**Availability Management Information System (AMIS)**	A virtual repository of all Availability Management data, usually stored in multiple physical locations
21	**Availability Plan**	A Plan to ensure that existing and future Availability Requirements for IT Services can be provided Cost Effectively
22	**Backup**	Copying data to protect against loss of Integrity or Availability of the original.
23	**Baseline**	A Benchmark used as a reference point. For example: • An ITSM Baseline can be used as a starting point to measure the effect of a Service Improvement Plan • A Performance Baseline can be used to measure changes in Performance over the lifetime of an IT Service • A Configuration Management Baseline can be used to enable the IT Infrastructure to be restored to a known Configuration if a Change or Release fails
24	**Best Practice**	Proven Activities or Processes that have been successfully used by multiple Organizations. ITIL is an example of Best Practice.
25	**Business Capacity Management**	In the context of ITSM, Business Capacity Management is the Activity responsible for understanding future Business Requirements for use in the

	(BCM)	Capacity Plan.
26	**Business Case**	Justification for a significant item of expenditure. Includes information about Costs, benefits, options, issues, Risks, and possible problems.
27	**Business Continuity Management (BCM)**	The Business Process responsible for managing Risks that could seriously impact the Business. BCM safeguards the interests of key stakeholders, reputation, brand and value creating activities. The BCM Process involves reducing Risks to an acceptable level and planning for the recovery of Business Processes should a disruption to the Business occur. BCM sets the Objectives, Scope and Requirements for IT Service Continuity Management.
28	**Business Continuity Plan (BCP)**	A Plan defining the steps required to Restore Business Processes following a disruption. The Plan will also identify the triggers for Invocation, people to be involved, communications etc. IT Service Continuity Plans form a significant part of Business Continuity Plans
29	**Business Impact Analysis (BIA)**	BIA is the Activity in Business Continuity Management that identifies Vital Business Functions and their dependencies. These dependencies may include Suppliers, people, other Business Processes, IT Services etc. BIA defines the recovery requirements for IT Services. These requirements include Recovery Time Objectives, Recovery Point Objectives and minimum Service Level Targets for each IT Service.
30	**Business Relationship Manager (BRM)**	A Role responsible for maintaining the Relationship with one or more Customers. This Role is often combined with the Service Level Manager Role.
31	**Business Unit**	A segment of the Business which has its own Plans, Metrics, income and Costs. Each Business Unit owns Assets and uses these to create value for Customers in the form of goods and Services.
32	**Capability**	The ability of an organization, person, Process, Application, Configuration Item or IT Service to carry out an Activity. Capabilities are intangible Assets of an Organization
33	**Capability Maturity Model (CMM)**	The Capability Maturity Model for Software is a model used to identify Best Practices to help increase Process Maturity. CMM was developed at the Software Engineering Institute (SEI) of Carnegie Mellon University. The SEI no longer maintains the SW-CMM model, its associated appraisal methods, or training materials.
34	**Capacity**	The maximum Throughput that a Configuration Item or IT Service can deliver whilst meeting agreed Service Level Targets. For some types of CI, Capacity may be the size or volume, for example a disk drive.
35	**Capacity Management**	The Process responsible for ensuring that the Capacity of IT Services and the IT Infrastructure is able to deliver agreed Service Level Targets in a Cost Effective and timely manner. Capacity Management considers all Resources required to deliver the IT Service, and plans for short-, medium- and long-term Business Requirements.
36	**Capacity Management**	A virtual repository of all Capacity Management data, usually stored in multiple physical locations

	Information System (CMIS)	
37	**Capacity Plan**	A Capacity Plan is used to manage the Resources required to deliver IT Services. The Plan contains scenarios for different predictions of Business demand, and costed options to deliver the agreed Service Level Targets.
38	**Category**	A named group of things that have something in common. Categories are used to group similar things together. For example, Cost Types are used to group similar types of Cost. Incident Categories are used to group similar types of Incidents, CI Types are used to group similar types of Configuration Item
39	**Certification**	Issuing a certificate to confirm Compliance to a Standard. Certification includes a formal Audit by an independent and Accredited body. The term Certification is also used to mean awarding a certificate to verify that a person has achieved a qualification
40	**Change**	The addition, modification or removal of anything that could have an effect on IT Services. The Scope should include all IT Services, Configuration Items, Processes, Documentation etc.
41	**Change Advisory Board (CAB)**	A group of people that advises the Change Manager in the Assessment, prioritization and scheduling of Changes. This board is usually made up of representatives from all areas within the IT Service Provider, the Business, and Third Parties such as Suppliers.
42	**Change Management**	The Process responsible for controlling the Lifecycle of all Changes. The primary objective of Change Management is to enable beneficial Changes to be made, with minimum disruption to IT Services.
43	**Change Model**	A repeatable way of dealing with a particular Category of Change. A Change Model defines specific pre-defined steps that will be followed for a Change of this Category. Change Models may be very simple, with no requirement for approval (e.g., Password Reset) or may be very complex with many steps that require approval (e.g., major software Release)
44	**Change Record**	A Record containing the details of a Change. Each Change Record documents the Lifecycle of a single Change. A Change Record is created for every Request for Change that is received, even those that are subsequently rejected. Change Records should reference the Configuration Items that are affected by the Change. Change Records are stored in the Configuration Management System.
45	**Change Schedule**	A Document that lists all approved Changes and their planned implementation dates. A Change Schedule is sometimes called a Forward Schedule of Change, even though it also contains information about Changes that have already been implemented
46	**CI Type**	A Category that is used to Classify CIs. The CI Type identifies the required Attributes and Relationships for a Configuration Record. Common CI Types include: hardware, Document, User etc.
47	**Classification**	The act of assigning a Category to something. Classification is used to ensure consistent management and reporting. CIs, Incidents, Problems, Changes etc. are usually classified.

48	**Compliance**	Ensuring that a Standard or set of Guidelines is followed, or that proper, consistent accounting or other practices are being employed.
49	**Component CI**	A Configuration Item that is part of an Assembly. For example, a CPU or Memory CI may be part of a Server CI
50	**Confidentiality**	A security principle that requires that data should only be accessed by authorized people.
51	**Configuration**	A generic term, used to describe a group of Configuration Items that work together to deliver an IT Service, or a recognizable part of an IT Service. Configuration is also used to describe the parameter settings for one or more CIs.
52	**Configuration Baseline**	A Baseline of a Configuration that has been formally agreed and is managed through the Change Management process. A Configuration Baseline is used as a basis for future Builds, Releases and Changes
53	**Configuration Control**	The Activity responsible for ensuring that adding, modifying or removing a CI is properly managed, for example by submitting a Request for Change or Service Request
54	**Configuration Item (CI)**	Any Component that needs to be managed in order to deliver an IT Service. Information about each CI is recorded in a Configuration Record within the Configuration Management System and is maintained throughout its Lifecycle by Configuration Management. CIs are under the control of Change Management. CIs typically include IT Services, hardware, software, buildings, people, and formal documentation such as Process documentation and SLAs.
55	**Configuration Management**	The Process responsible for maintaining information about Configuration Items required to deliver an IT Service, including their Relationships. This information is managed throughout the Lifecycle of the CI. Configuration Management is part of an overall Service Asset and Configuration Management Process.
56	**Configuration Management Database (CMDB)**	A database used to store Configuration Records throughout their Lifecycle. The Configuration Management System maintains one or more CMDBs, and each CMDB stores Attributes of CIs, and Relationships with other Cis
57	**Configuration Management System (CMS)**	A set of tools and databases that are used to manage an IT Service Provider's Configuration data. The CMS also includes information about Incidents, Problems, Known Errors, Changes and Releases; and may contain data about employees, Suppliers, locations, Business Units, Customers and Users. The CMS includes tools for collecting, storing, managing, updating, and presenting data about all Configuration Items and their Relationships. The CMS is maintained by Configuration Management and is used by all IT Service Management Processes.
58	**Continual Service Improvement (CSI)**	A stage in the Lifecycle of an IT Service and the title of one of the Core ITIL publications. Continual Service Improvement is responsible for managing improvements to IT Service Management Processes and IT Services. The Performance of the IT Service Provider is continually measured and

		improvements are made to Processes, IT Services and IT Infrastructure in order to increase Efficiency, Effectiveness, and Cost Effectiveness
59	**Control Processes**	The ISO/IEC 20000 Process group that includes Change Management and Configuration Management
60	**Cost Benefit Analysis**	An Activity that analyses and compares the Costs and the benefits involved in one or more alternative courses of action.
61	**Critical Success Factor (CSF)**	Something that must happen if a Process, Project, Plan, or IT Service is to succeed. KPIs are used to measure the achievement of each CSF. For example, a CSF of "protect IT Services when making Changes" could be measured by KPIs such as "percentage reduction of unsuccessful Changes", "percentage reduction in Changes causing Incidents" etc.
62	**Customer**	Someone who buys goods or Services. The Customer of an IT Service Provider is the person or group who defines and agrees the Service Level Targets. The term Customers is also sometimes informally used to mean Users, for example "this is a customer focused Organization"
63	**Dashboard**	A graphical representation of overall IT Service Performance and Availability. Dashboard images may be updated in real-time, and can also be included in management reports and web pages. Dashboards can be used to support Service Level Management, Event Management or Incident Diagnosis
64	**Definitive Media Library (DML)**	One or more locations in which the definitive and approved versions of all software Configuration Items are securely stored. The DML may also contain associated CIs such as licenses and documentation. The DML is a single logical storage area even if there are multiple locations. All software in the DML is under the control of Change and Release Management and is recorded in the Configuration Management System. Only software from the DML is acceptable for use in a Release.
65	**Demand Management**	Activities that understand and influence Customer demand for Services and the provision of Capacity to meet these demands. At a Strategic level Demand Management can involve analysis of Patterns of Business Activity and User Profiles. At a Tactical level it can involve use of Differential Charging to encourage Customers to use IT Services at less busy times.
66	**Detection**	A stage in the Incident Lifecycle. Detection results in the Incident becoming known to the Service Provider. Detection can be automatic, or can be the result of a User logging an Incident
67	**Differential Charging**	A technique used to support Demand Management by charging different amounts for the same IT Service Function at different times
68	**Direct Cost**	A cost of providing an IT Service which can be allocated in full to a specific Customer, Cost Centre, Project etc. For example, cost of providing non-shared servers or software licenses.
69	**Directory Service**	An application that manages information about IT Infrastructure available on a network, and corresponding User access Rights
70	**Downtime**	The time when a Configuration Item or IT Service is not Available during its Agreed Service Time. The Availability of an IT Service is often calculated from

		Agreed Service Time and Downtime.
71	**Effectiveness**	A measure of whether the Objectives of a Process, Service or Activity have been achieved. An Effective Process or Activity is one that achieves its agreed Objectives
72	**Efficiency**	A measure of whether the right number of resources have been used to deliver a Process, Service or Activity. An Efficient Process achieves its Objectives with the minimum amount of time, money, people or other resources.
73	**Emergency Change**	A Change that must be introduced as soon as possible. For example, to resolve a Major Incident or implement a Security patch. The Change Management Process will normally have a specific Procedure for handling Emergency Changes.
74	**Emergency Change Advisory Board (ECAB)**	A sub-set of the Change Advisory Board who make decisions about high impact Emergency Changes. Membership of the ECAB may be decided at the time a meeting is called, and depends on the nature of the Emergency Change.
75	**Error**	A design flaw or malfunction that causes a Failure of one or more Configuration Items or IT Services. A mistake made by a person or a faulty Process that impacts a CI or IT Service is also an Error.
76	**Escalation**	An Activity that obtains additional Resources when these are needed to meet Service Level Targets or Customer expectations. Escalation may be needed within any IT Service Management Process, but is most commonly associated with Incident Management, Problem Management and the management of Customer complaints. There are two types of Escalation, Functional Escalation and Hierarchic Escalation.
77	**Event**	A change of state which has significance for the management of a Configuration Item or IT Service. The term Event is also used to mean an Alert or notification created by any IT Service, Configuration Item or Monitoring tool. Events typically require IT Operations personnel to take actions, and often lead to Incidents being logged.
78	**Event Management**	The Process responsible for managing Events throughout their Lifecycle. Event Management is one of the main Activities of IT Operations.
79	**Exception Report**	A Document containing details of one or more KPIs or other important targets that have exceeded defined Thresholds. Examples include SLA targets being missed or about to be missed, and a Performance Metric indicating a potential Capacity problem.
80	**External Service Provider**	An IT Service Provider which is part of a different Organization to their Customer. An IT Service Provider may have both Internal Customers and External Customers
81	**Facilities Management**	The Function responsible for managing the physical Environment where the IT Infrastructure is located. Facilities Management includes all aspects of managing the physical Environment, for example power and cooling, building Access Management, and environmental Monitoring.
82	**Fault Tree Analysis (FTA)**	A technique that can be used to determine the chain of Events that leads to a Problem. Fault Tree Analysis represents a chain of Events using Boolean notation in a diagram.

83	**Financial Management**	The Function and Processes responsible for managing an IT Service Provider's Budgeting, Accounting and Charging Requirements
84	**Fit for Purpose**	An informal term used to describe a Process, Configuration Item, IT Service etc. that is capable of meeting its Objectives or Service Levels. Being Fit for Purpose requires suitable Design, implementation, Control and maintenance
85	**Fixed Cost**	A Cost that does not vary with IT Service usage. For example, the cost of Server hardware
86	**Follow the Sun**	A methodology for using Service Desks and Support Groups around the world to provide seamless 24 * 7 Service. Calls, Incidents, Problems and Service Requests are passed between groups in different time zones
87	**Function**	A team or group of people and the tools they use to carry out one or more Processes or Activities. For example, the Service Desk. The term Function also has two other meanings: - • An intended purpose of a Configuration Item, Person, Team, Process, or IT Service. For example, one Function of an Email Service may be to store and forward outgoing mails, one Function of a Business Process may be to dispatch goods to Customers. • To perform the intended purpose correctly, "The computer is Functioning"
88	**Gap Analysis**	An Activity which compares two sets of data and identifies the differences. Gap Analysis is commonly used to compare a set of Requirements with actual delivery
89	**Governance**	Ensuring that Policies and Strategy are actually implemented, and that required Processes are correctly followed. Governance includes defining Roles and responsibilities, measuring and reporting, and taking actions to resolve any issues identified
90	**Hierarchic Escalation**	Informing or involving more senior levels of management to assist in an Escalation
91	**Impact**	A measure of the effect of an Incident, Problem or Change on Business Processes. Impact is often based on how Service Levels will be affected. Impact and Urgency are used to assign Priority.
92	**Incident**	An unplanned interruption to an IT Service or a reduction in the Quality of an IT Service. Failure of a Configuration Item that has not yet impacted Service is also an Incident. For example, Failure of one disk from a mirror set.
93	**Incident Management**	The Process responsible for managing the Lifecycle of all Incidents. The primary Objective of Incident Management is to return the IT Service to Users as quickly as possible.
94	**Incident Record**	A Record containing the details of an Incident. Each Incident record documents the Lifecycle of a single Incident
95	**Indirect Cost**	A Cost of providing an IT Service which cannot be allocated in full to a specific Customer. For example, Cost of providing shared Servers or software licenses. Also known as Overhead
96	**Information Security**	The Process that ensures the Confidentiality, Integrity and Availability of an Organization's Assets, information, data and IT Services. Information Security

	Management (ISM)	Management usually forms part of an Organizational approach to Security Management which has a wider scope than the IT Service Provider, and includes handling of paper, building access, phone calls etc., for the entire Organization.
97	**Information Security Management System (ISMS)**	The framework of Policy, Processes, Standards, Guidelines and tools that ensures an organization can achieve its Information Security Management Objectives.
98	**Information Security Policy**	The Policy that governs the Organization's approach to Information Security Management.
99	**Information Technology (IT)**	The use of technology for the storage, communication or processing of information. The technology typically includes computers, telecommunications, Applications and other software. The information may include Business data, voice, images, video, etc. Information Technology is often used to support Business Processes through IT Services.
100	**Integrity**	A security principle that ensures data and Configuration Items are only modified by authorized personnel and Activities. Integrity considers all possible causes of modification, including software and hardware Failure, environmental Events, and human intervention.
101	**Internal Service Provider**	An IT Service Provider which is part of the same Organization as their Customer. An IT Service Provider may have both Internal Customers and External Customers
102	**International Organization for Standardization (ISO)**	The International Organization for Standardization (ISO) is the world's largest developer of Standards. ISO is a non-governmental organization which is a network of the national standards institutes of 156 countries
103	**ISO 9000**	A generic term that refers to a number of international Standards and Guidelines for Quality Management Systems
104	**ISO 9001**	An international Standard for Quality Management Systems
105	**ISO/IEC 20000**	ISO Specification and Code of Practice for IT Service Management. ISO/IEC 20000 is aligned with ITIL Best Practice
106	**ISO/IEC 27001**	ISO Specification for Information Security Management. The corresponding Code of Practice is ISO/IEC 17799
107	**IT Infrastructure**	All of the hardware, software, networks, facilities etc. that are required to Develop, Test, deliver, Monitor, Control or support IT Services. The term IT Infrastructure includes all of the Information Technology but not the associated people, Processes and documentation
108	**IT Operations**	Activities carried out by IT Operations Control, including Console Management, Job Scheduling, Backup and Restore, and Print and Output Management. IT Operations is also used as a synonym for Service Operation.
109	**IT Operations Control**	The Function responsible for Monitoring and Control of the IT Services and IT Infrastructure
110	**IT Operations Management**	The Function within an IT Service Provider which performs the daily Activities needed to manage IT Services and the supporting IT Infrastructure. IT

		Operations Management includes IT Operations Control and Facilities Management
111	**IT Service**	A Service provided to one or more Customers by an IT Service Provider. An IT Service is based on the use of Information Technology and supports the Customer's Business Processes. An IT Service is made up from a combination of people, Processes and technology and should be defined in a Service Level Agreement
112	**IT Service Continuity Management (ITSCM)**	The Process responsible for managing Risks that could seriously impact IT Services. ITSCM ensures that the IT Service Provider can always provide minimum agreed Service Levels, by reducing the Risk to an acceptable level and Planning for the Recovery of IT Services. ITSCM should be designed to support Business Continuity Management.
113	**IT Service Continuity Plan**	A Plan defining the steps required to Recover one or more IT Services. The Plan will also identify the triggers for Invocation, people to be involved, communications etc. The IT Service Continuity Plan should be part of a Business Continuity Plan.
114	**IT Service Management (ITSM)**	The implementation and management of Quality IT Services that meet the needs of the Business. IT Service Management is performed by IT Service Providers through an appropriate mix of people, Process and Information Technology.
115	**IT Service Management Forum (itSMF)**	The IT Service Management Forum is an independent Organization dedicated to promoting a professional approach to IT Service Management. The itSMF is a not-for-profit membership Organization with representation in many countries around the world. The itSMF and its membership contribute to the development of ITIL and associated IT Service Management Standards.
116	**ITIL**	A set of Best Practice guidance for IT Service Management. ITIL is owned by the OGC and consists of a series of publications giving guidance on the provision of Quality IT Services, and on the Processes and facilities needed to support them.
117	**Key Performance Indicator (KPI)**	A Metric that is used to help manage a Process, IT Service or Activity. Many Metrics may be measured, but only the most important of these are defined as KPIs and used to actively manage and report on the Process, IT Service or Activity. KPIs should be selected to ensure that Efficiency, Effectiveness, and Cost Effectiveness are all managed.
118	**Knowledge Base**	A logical database containing the data used by the Service Knowledge Management System.
119	**Knowledge Management**	The Process responsible for gathering, analyzing, storing and sharing knowledge and information within an organization. The primary purpose of Knowledge Management is to improve Efficiency by reducing the need to rediscover knowledge
120	**Known Error**	A Problem that has a documented Root Cause and a Workaround. Known Errors are created and managed throughout their Lifecycle by Problem Management. Known Errors may also be identified by Development or Suppliers
121	**Known Error**	A database containing all Known Error Records. This database is created by

	Database (KEDB)	Problem Management and used by Incident and Problem Management. The Known Error Database is part of the Service Knowledge Management System
122	**Known Error Record**	A Record containing the details of a Known Error. Each Known Error Record documents the Lifecycle of a Known Error, including the Status, Root Cause and Workaround. In some implementations a Known Error is documented using additional fields in a Problem Record.
123	**Lifecycle**	The various stages in the life of an IT Service, Configuration Item, Incident, Problem, Change etc. The Lifecycle defines the Categories for Status and the Status transitions that are permitted. For example: • The Lifecycle of an Application includes Requirements, Design, Build, Deploy, Operate, Optimize. • The Expanded Incident Lifecycle includes Detect, Respond, Diagnose, Repair, Recover, Restore. • The lifecycle of a Server may include: Ordered, Received, In Test, Live, Disposed etc.
124	**Maintainability**	A measure of how quickly and Effectively a Configuration Item or IT Service can be restored to normal working after a Failure. Maintainability is often measured and reported as MTRS. Maintainability is also used in the context of Software or IT Service Development to mean ability to be Changed or Repaired easily.
125	**Major Incident**	The highest Category of Impact for an Incident. A Major Incident results in significant disruption to the Business.
126	**Management of Risk (MoR)**	The OGC methodology for managing Risks. MoR includes all the Activities required to identify and control the exposure to Risk which may have an impact on the achievement of an Organization’s Business Objectives.
127	**Manual Workaround**	A Workaround that requires manual intervention. Manual Workaround is also used as the name of a Recovery Option in which The Business Process Operates without the use of IT Services. This is a temporary measure and is usually combined with another Recovery Option.
128	**Maturity Level**	A named level in a Maturity model such as the Carnegie Mellon Capability Maturity Model Integration
129	**Mean Time Between Failures (MTBF)**	A Metric for measuring and reporting Reliability. MTBF is the average time that a Configuration Item or IT Service can perform its agreed Function without interruption. This is measured from when the CI or IT Service starts working, until it next fails.
130	**Mean Time Between Service Incidents (MTBSI)**	A Metric used for measuring and reporting Reliability. MTBSI is the mean time from when a System or IT Service fails, until it next fails. MTBSI is equal to MTBF + MTRS
131	**Mean Time To Repair (MTTR)**	The average time taken to repair a Configuration Item or IT Service after a Failure. MTTR is measured from when the CI or IT Service fails until it is Repaired. MTTR does not include the time required to Recover or Restore. MTTR is sometimes incorrectly used to mean Mean Time to Restore Service
132	**Mean Time to Restore Service**	The average time taken to Restore a Configuration Item or IT Service after a Failure. MTRS is measured from when the CI or IT Service fails until it is fully

	(MTRS)	Restored and delivering its normal functionality
133	**Metric**	Something that is measured and reported to help manage a Process, IT Service or Activity.
134	**Mission Statement**	The Mission Statement of an Organization is a short but complete description of the overall purpose and intentions of that Organization. It states what is to be achieved, but not how this should be done.
135	**Notional Charging**	An approach to Charging for IT Services. Charges to Customers are calculated and Customers are informed of the charge, but no money is actually transferred. Notional Charging is sometimes introduced to ensure that Customers are aware of the Costs they incur, or as a stage during the introduction of real Charging.
136	**Office of Government Commerce (OGC)**	OGC owns the ITIL brand (copyright and trademark). OGC is a UK Government department that supports the delivery of the government's procurement agenda through its work in collaborative procurement and in raising levels of procurement skills and capability with departments. It also provides support for complex public sector projects.
137	**Operational Level Agreement (OLA)**	An Agreement between an IT Service Provider and another part of the same Organization. An OLA supports the IT Service Provider's delivery of IT Services to Customers. The OLA defines the goods or Services to be provided and the responsibilities of both parties. For example, there could be an OLA • between the IT Service Provider and a procurement department to obtain hardware in agreed times • between the Service Desk and a Support Group to provide Incident Resolution in agreed times.
138	**Outsourcing**	Using an External Service Provider to manage IT Services
139	**Pain Value Analysis**	A technique used to help identify the Business Impact of one or more Problems. A formula is used to calculate Pain Value based on the number of Users affected, the duration of the Downtime, the Impact on each User, and the cost to the Business (if known).
140	**Partnership**	A relationship between two Organizations which involves working closely together for common goals or mutual benefit. The IT Service Provider should have a Partnership with the Business, and with Third Parties who are critical to the delivery of IT Services.
141	**Pattern of Business Activity (PBA)**	A Workload profile of one or more Business Activities. Patterns of Business Activity are used to help the IT Service Provider understand and plan for different levels of Business Activity.
142	**Performance Management**	The Process responsible for day-to-day Capacity Management Activities. These include Monitoring, Threshold detection, Performance analysis and Tuning, and implementing Changes related to Performance and Capacity
143	**Pilot**	A limited Deployment of an IT Service, a Release or a Process to the Live Environment. A Pilot is used to reduce Risk and to gain User feedback and Acceptance
144	**Plan-Do-Check**	A four-stage cycle for Process management, attributed to Edward Deming. Plan-

	Act	Do-Check-Act is also called the Deming Cycle. PLAN: Design or revise Processes that support the IT Services. DO: Implement the Plan and manage the Processes. CHECK: Measure the Processes and IT Services, compare with Objectives and produce reports ACT: Plan and implement Changes to improve the Processes.
145	**Policy**	Formally documented management expectations and intentions. Policies are used to direct decisions, and to ensure consistent and appropriate development and implementation of Processes, Standards, Roles, Activities, IT Infrastructure etc.
146	**Post Implementation Review (PIR)**	A Review that takes place after a Change or a Project has been implemented. A PIR determines if the Change or Project was successful, and identifies opportunities for improvement.
147	**Priority**	A Category used to identify the relative importance of an Incident, Problem or Change. Priority is based on Impact and Urgency, and is used to identify required times for actions to be taken. For example, the SLA may state that Priority2 Incidents must be resolved within 12 hours
148	**Proactive Problem Management**	Part of the Problem Management Process. The Objective of Proactive Problem Management is to identify Problems that might otherwise be missed. Proactive Problem Management Analyses Incident Records, and uses data collected by other IT Service Management Processes to identify trends or significant Problems
149	**Problem**	A cause of one or more Incidents. The cause is not usually known at the time a Problem Record is created, and the Problem Management Process is responsible for further investigation
150	**Problem Management**	The Process responsible for managing the Lifecycle of all Problems. The primary Objectives of Problem Management are to prevent Incidents from happening, and to minimize the Impact of Incidents that cannot be prevented.
151	**Problem Record**	A Record containing the details of a Problem. Each Problem Record documents the Lifecycle of a single Problem.
152	**Procedure**	A Document containing steps that specify how to achieve an Activity. Procedures are defined as part of Processes.
153	**Process**	A structured set of Activities designed to accomplish a specific Objective. A Process takes one or more defined inputs and turns them into defined outputs. A Process may include any of the Roles, responsibilities, tools and management Controls required to reliably deliver the outputs. A Process may define Policies, Standards, Guidelines, Activities, and Work Instructions if they are needed
154	**Process Control**	The Activity of planning and regulating a Process, with the Objective of performing the Process in an Effective, Efficient, and consistent manner.
155	**Process Manager**	A Role responsible for Operational management of a Process. The Process Manager's responsibilities include Planning and co-ordination of all Activities required to carry out, monitor and report on the Process. There may be several Process Managers for one Process, for example regional Change Managers or IT

		Service Continuity Managers for each data center. The Process Manager Role is often assigned to the person who carries out the Process Owner Role, but the two Roles may be separate in larger Organizations.
156	**Process Owner**	A Role responsible for ensuring that a Process is Fit for Purpose. The Process Owner's responsibilities include sponsorship, Design, Change Management and continual improvement of the Process and its Metrics. This Role is often assigned to the same person who carries out the Process Manager Role, but the two Roles may be separate in larger Organizations.
157	**Project**	A temporary Organization, with people and other Assets required to achieve an Objective or other Outcome. Each Project has a Lifecycle that typically includes initiation, Planning, execution, Closure etc. Projects are usually managed using a formal methodology
158	**Quality Assurance (QA)**	The Process responsible for ensuring that the Quality of a product, Service or Process will provide its intended Value.
159	**Quality Management System (QMS)**	The set of Processes responsible for ensuring that all work carried out by an organization is of a suitable Quality to reliably meet Business Objectives or Service Levels.
160	**RACI**	A Model used to help define Roles and Responsibilities. RACI stands for Responsible, Accountable, Consulted and Informed
161	**Release**	A collection of hardware, software, documentation, Processes or other Components required to implement one or more approved Changes to IT Services. The contents of each Release are managed, Tested, and Deployed as a single entity
162	**Release and Deployment Management**	The Process responsible for both Release Management and Deployment
163	**Release Management**	The Process responsible for Planning, scheduling and controlling the movement of Releases to Test and Live Environments. The primary Objective of Release Management is to ensure that the integrity of the Live Environment is protected and that the correct Components are released. Release Management is part of the Release and Deployment Management Process
164	**Release Unit**	Components of an IT Service that are normally Released together. A Release Unit typically includes sufficient Components to perform a useful Function. For example, one Release Unit could be a Desktop PC, including Hardware, Software, Licenses, Documentation etc. A different Release Unit may be the complete Payroll Application, including IT Operations Procedures and User training
165	**Reliability**	A measure of how long a Configuration Item or IT Service can perform its agreed Function without interruption. Usually measured as MTBF or MTBSI. The term Reliability can also be used to state how likely it is that a Process, Function etc. will deliver its required outputs
166	**Request for Change (RFC)**	A formal proposal for a Change to be made. An RFC includes details of the proposed Change, and may be recorded on paper or electronically. The term RFC is often misused to mean a Change Record, or the Change itself.

167	**Request Fulfilment**	The Process responsible for managing the Lifecycle of all Service Requests.
168	**Requirement**	A formal statement of what is needed. For example, a Service Level Requirement, a Project Requirement or the required Deliverables for a Process
169	**Resilience**	The ability of a Configuration Item or IT Service to resist Failure or to Recover quickly following a Failure. For example, an armored cable will resist failure when put under stress.
170	**Resource**	A generic term that includes IT Infrastructure, people, money or anything else that might help to deliver an IT Service. Resources are considered to be Assets of an Organization
171	**Response Time**	A measure of the time taken to complete an operation or Transaction. Used in Capacity Management as a measure of IT Infrastructure Performance, and in Incident Management as a measure of the time taken to answer the phone, or to start Diagnosis.
172	**Restore**	Taking action to return an IT Service to the Users after Repair and Recovery from an Incident. This is the primary Objective of Incident Management.
173	**Retire**	Permanent removal of an IT Service, or other Configuration Item, from the Live Environment. Retired is a stage in the Lifecycle of many Configuration Items.
174	**Return on Investment (ROI)**	A measurement of the expected benefit of an investment. In the simplest sense it is the net profit of an investment divided by the net worth of the assets invested.
175	**Rights**	Entitlements, or permissions, granted to a User or Role. For example, the Right to modify particular data, or to authorize a Change.
176	**Risk**	A possible Event that could cause harm or loss, or affect the ability to achieve Objectives. A Risk is measured by the probability of a Threat, the Vulnerability of the Asset to that Threat, and the Impact it would have if it occurred.
177	**Risk Assessment**	The initial steps of Risk Management. Analyzing the value of Assets to the business, identifying Threats to those Assets, and evaluating how Vulnerable each Asset is to those Threats. Risk Assessment can be quantitative (based on numerical data) or qualitative
178	**Risk Management**	The Process responsible for identifying, assessing and controlling Risks
179	**Role**	A set of responsibilities, Activities and authorities granted to a person or team. A Role is defined in a Process. One person or team may have multiple Roles, for example the Roles of Configuration Manager and Change Manager may be carried out by a single person.
180	**Rollout**	Synonym for Deployment. Most often used to refer to complex or phased Deployments or Deployments to multiple locations.
181	**Root Cause**	The underlying or original cause of an Incident or Problem
182	**Root Cause Analysis (RCA)**	An Activity that identifies the Root Cause of an Incident or Problem. RCA typically concentrates on IT Infrastructure failures.
183	**Service**	A means of delivering value to Customers by facilitating Outcomes Customers want to achieve without the ownership of specific Costs and Risks.
184	**Service**	A set of criteria used to ensure that an IT Service meets its functionality and

	Acceptance Criteria (SAC)	Quality Requirements and that the IT Service Provider is ready to Operate the new IT Service when it has been Deployed.
185	**Service Asset and Configuration Management (SACM)**	The Process responsible for both Configuration Management and Asset Management.
186	**Service Capacity Management (SCM)**	The Activity responsible for understanding the Performance and Capacity of IT Services. The Resources used by each IT Service and the pattern of usage over time are collected, recorded, and analyzed for use in the Capacity Plan
187	**Service Catalogue**	A database or structured Document with information about all Live IT Services, including those available for Deployment. The Service Catalogue is the only part of the Service Portfolio published to Customers, and is used to support the sale and delivery of IT Services. The Service Catalogue includes information about deliverables, prices, contact points, ordering and request Processes.
188	**Service Continuity Management**	Synonym for IT Service Continuity Management
189	**Service Contract**	A Contract to deliver one or more IT Services. The term Service Contract is also used to mean any Agreement to deliver IT Services, whether this is a legal Contract or an SLA.
190	**Service Design**	A stage in the Lifecycle of an IT Service. Service Design includes a number of Processes and Functions and is the title of one of the Core ITIL publications
191	**Service Design Package**	Document(s) defining all aspects of an IT Service and its Requirements through each stage of its Lifecycle. A Service Design Package is produced for each new IT Service, major Change, or IT Service Retirement.
192	**Service Desk**	The Single Point of Contact between the Service Provider and the Users. A typical Service Desk manages Incidents and Service Requests, and also handles communication with the Users.
193	**Service Improvement Plan (SIP)**	A formal Plan to implement improvements to a Process or IT Service
194	**Service Knowledge Management System (SKMS)**	A set of tools and databases that are used to manage knowledge and information. The SKMS includes the Configuration Management System, as well as other tools and databases. The SKMS stores, manages, updates, and presents all information that an IT Service Provider needs to manage the full Lifecycle of IT Services.
195	**Service Level**	Measured and reported achievement against one or more Service Level Targets. The term Service Level is sometimes used informally to mean Service Level Target.
196	**Service Level Agreement (SLA)**	An Agreement between an IT Service Provider and a Customer. The SLA describes the IT Service, documents Service Level Targets, and specifies the responsibilities of the IT Service Provider and the Customer. A single SLA may cover multiple IT Services or multiple Customers
197	**Service Level**	The Process responsible for negotiating Service Level Agreements, and ensuring

	Management (SLM)	that these are met. SLM is responsible for ensuring that all IT Service Management Processes, Operational Level Agreements, and Underpinning Contracts, are appropriate for the agreed Service Level Targets. SLM monitors and reports on Service Levels, and holds regular Customer reviews.
198	**Service Level Package (SLP)**	A defined level of Utility and Warranty for a particular Service Package. Each SLP is designed to meet the needs of a particular Pattern of Business Activity
199	**Service Level Requirement (SLR)**	A Customer Requirement for an aspect of an IT Service. SLRs are based on Business Objectives and are used to negotiate agreed Service Level Targets.
200	**Service Management**	Service Management is a set of specialized organizational capabilities for providing value to customers in the form of services.
201	**Service Management Lifecycle**	An approach to IT Service Management that emphasizes the importance of coordination and Control across the various Functions, Processes, and Systems necessary to manage the full Lifecycle of IT Services. The Service Management Lifecycle approach considers the Strategy, Design, Transition, Operation and Continuous Improvement of IT Services.
202	**Service Manager**	A manager who is responsible for managing the end-to-end Lifecycle of one or more IT Services. The term Service Manager is also used to mean any manager within the IT Service Provider. Most commonly used to refer to a Business Relationship Manager, a Process Manager, an Account Manager or a senior manager with responsibility for IT Services overall.
203	**Service Operation**	A stage in the Lifecycle of an IT Service. Service Operation includes a number of Processes and Functions and is the title of one of the Core ITIL publications.
204	**Service Owner**	A Role which is accountable for the delivery of a specific IT Service.
205	**Service Package**	A detailed description of an IT Service that is available to be delivered to Customers. A Service Package includes a Service Level Package and one or more Core Services and Supporting Services
206	**Service Pipeline**	A database or structured Document listing all IT Services that are under consideration or Development, but are not yet available to Customers. The Service Pipeline provides a business view of possible future IT Services and is part of the Service Portfolio which is not normally published to Customers
207	**Service Portfolio**	The complete set of Services that are managed by a Service Provider. The Service Portfolio is used to manage the entire Lifecycle of all Services, and includes three Categories: Service Pipeline (proposed or in Development); Service Catalogue (Live or available for Deployment); and Retired Services.
208	**Service Portfolio Management (SPM)**	The Process responsible for managing the Service Portfolio. Service Portfolio Management considers Services in terms of the Business value that they provide.
209	**Service Potential**	The total possible value of the overall Capabilities and Resources of the IT Service Provider
210	**Service Provider**	An organization supplying Services to one or more Internal Customers or External Customers. Service Provider is often used as an abbreviation for IT Service Provider

211	**Service Request**	A request from a User for information, or advice, or for a Standard Change or for Access to an IT Service. For example, to reset a password, or to provide standard IT Services for a new User. Service Requests are usually handled by a Service Desk, and do not require an RFC to be submitted
212	**Service Strategy**	The title of one of the Core ITIL publications. Service Strategy establishes an overall Strategy for IT Services and for IT Service Management.
213	**Service Transition**	A stage in the Lifecycle of an IT Service. Service Transition includes a number of Processes and Functions and is the title of one of the Core ITIL publications.
214	**Service Utility**	The Functionality of an IT Service from the Customer's perspective. The Business value of an IT Service is created by the combination of Service Utility (what the Service does) and Service Warranty (how well it does it).
215	**Service Validation and Testing**	The Process responsible for Validation and Testing of a new or Changed IT Service. Service Validation and Testing ensures that the IT Service matches its Design Specification and will meet the needs of the Business
216	**Service Valuation**	A measurement of the total Cost of delivering an IT Service, and the total value to the Business of that IT Service. Service Valuation is used to help the Business and the IT Service Provider agree on the value of the IT Service
217	**Service Warranty**	Assurance that an IT Service will meet agreed Requirements. This may be a formal Agreement such as a Service Level Agreement or Contract, or may be a marketing message or brand image. The Business value of an IT Service is created by the combination of Service Utility (what the Service does) and Service Warranty (how well it does it)
218	**Serviceability**	The ability of a Third-Party Supplier to meet the terms of their Contract. This Contract will include agreed levels of Reliability, Maintainability or Availability for a Configuration Item.
219	**Single Point of Contact**	Providing a single consistent way to communicate with an organization or Business Unit. For example, a Single Point of Contact for an IT Service Provider is usually called a Service Desk.
220	**SLAM Chart**	A Service Level Agreement Monitoring Chart is used to help monitor and report achievements against Service Level Targets. A SLAM Chart is typically color coded to show whether each agreed Service Level Target has been met, missed, or nearly missed during each of the previous 12 months.
221	**Stakeholder**	All people who have an interest in an Organization, Project, IT Service etc. Stakeholders may be interested in the Activities, targets, Resources, or Deliverables. Stakeholders may include Customers, Partners, employees, shareholders, owners, etc.
222	**Standard**	A mandatory Requirement. Examples include ISO/IEC 20000 (an international Standard), an internal security Standard for Unix configuration, or a government Standard for how financial Records should be maintained. The term Standard is also used to refer to a Code of Practice or Specification published by a Standards organization such as ISO or BSI
223	**Standard Change**	A pre-approved Change that is low Risk, relatively common and follows a Procedure or Work Instruction. For example, password reset or provision of

		standard equipment to a new employee. RFCs are not required to implement a Standard Change, and they are logged and tracked using a different mechanism, such as a Service Request.
224	**Status**	The name of a required field in many types of Record. It shows the current stage in the Lifecycle of the associated Configuration Item, Incident, Problem etc.
225	**Status Accounting**	The Activity responsible for recording and reporting the Lifecycle of each Configuration Item.
226	**Strategic**	The highest of three levels of Planning and delivery (Strategic, Tactical, Operational). Strategic Activities include Objective setting and long-term Planning to achieve the overall Vision.
227	**Supplier**	A Third Party responsible for supplying goods or Services that are required to deliver IT services. Examples of suppliers include commodity hardware and software vendors, network and telecom providers, and Outsourcing Organizations
228	**Supplier and Contract Database (SCD)**	A database or structured Document used to manage Supplier Contracts throughout their Lifecycle. The SCD contains key Attributes of all Contracts with Suppliers, and should be part of the Service Knowledge Management System.
229	**Supplier Management**	The Process responsible for ensuring that all Contracts with Suppliers support the needs of the Business, and that all Suppliers meet their contractual commitments.
230	**System**	A number of related things that work together to achieve an overall Objective. For example: • A computer System including hardware, software and Applications. • A management System, including multiple Processes that are planned and managed together. For example, a Quality Management System. • A Database Management System or Operating System that includes many software modules that are designed to perform a set of related Functions.
231	**Tactical**	The middle of three levels of Planning and delivery (Strategic, Tactical, Operational). Tactical Activities include the medium-term Plans required to achieve specific Objectives, typically over a period of weeks to months
232	**Technical Management**	The Function responsible for providing technical skills in support of IT Services and management of the IT Infrastructure. Technical Management defines the Roles of Support Groups, as well as the tools, Processes and Procedures required.
233	**Third-line Support**	The third level in a hierarchy of Support Groups involved in the resolution of Incidents and investigation of Problems. Each level contains more specialist skills, or has more time or other Resources.
234	**Threat**	Anything that might exploit a Vulnerability. Any potential cause of an Incident can be considered to be a Threat. For example, a fire is a Threat that could exploit the Vulnerability of flammable floor coverings. This term is commonly used in Information Security Management and IT Service Continuity Management, but also applies to other areas such as Problem and Availability

		Management.
235	**Threshold**	The value of a Metric which should cause an Alert to be generated, or management action to be taken. For example, "Priority1 Incident not solved within 4 hours", "more than 5 soft disk errors in an hour", or "more than 10 failed changes in a month".
236	**Total Cost of Ownership (TCO)**	A methodology used to help make investment decisions. TCO assesses the full Lifecycle Cost of owning a Configuration Item, not just the initial Cost or purchase price.
237	**Trend Analysis**	Analysis of data to identify time related patterns. Trend Analysis is used in Problem Management to identify common Failures or fragile Configuration Items, and in Capacity Management as a Modelling tool to predict future behavior. It is also used as a management tool for identifying deficiencies in IT Service Management Processes.
238	**Tuning**	The Activity responsible for Planning Changes to make the most efficient use of Resources. Tuning is part of Performance Management, which also includes Performance Monitoring and implementation of the required Changes
239	**Type I Service Provider**	An Internal Service Provider that is embedded within a Business Unit. There may be several Type I Service Providers within an Organization.
240	**Type II Service Provider**	An Internal Service Provider that provides shared IT Services to more than one Business Unit.
241	**Type III Service Provider**	A Service Provider that provides IT Services to External Customers.
242	**Underpinning Contract (UC)**	A Contract between an IT Service Provider and a Third Party. The Third Party provides goods or Services that support delivery of an IT Service to a Customer. The Underpinning Contract defines targets and responsibilities that are required to meet agreed Service Level Targets in an SLA.
243	**Urgency**	A measure of how long it will be until an Incident, Problem or Change has a significant Impact on the Business. For example, a high Impact Incident may have low Urgency, if the Impact will not affect the Business until the end of the financial year. Impact and Urgency are used to assign Priority.
244	**User**	A person who uses the IT Service on a day-to-day basis. Users are distinct from Customers, as some Customers do not use the IT Service directly
245	**User Profile (UP)**	A pattern of User demand for IT Services. Each User Profile includes one or more Patterns of Business Activity.
246	**Utility**	Functionality offered by a Product or Service to meet a particular need. Utility is often summarized as "what it does".
247	**Value on Investment (VOI)**	A measurement of the expected benefit of an investment. VOI considers both financial and intangible benefits.
248	**Variable Cost**	A Cost that depends on how much the IT Service is used, how many products are produced, the number and type of Users, or something else that cannot be fixed in advance.
249	**Vision**	A description of what the Organization intends to become in the future. A Vision is created by senior management and is used to help influence Culture and

		Strategic Planning.
250	**Vulnerability**	A weakness that could be exploited by a Threat. For example, an open firewall port, a password that is never changed, or a flammable carpet. A missing Control is also considered to be a Vulnerability.
251	**Warranty**	A promise or guarantee that a product or Service will meet its agreed Requirements.
252	**Work Instruction**	A Document containing detailed instructions that specify exactly what steps to follow to carry out an Activity. A Work Instruction contains much more detail than a Procedure and is only created if very detailed instructions are needed
253	**Workaround**	Reducing or eliminating the Impact of an Incident or Problem for which a full Resolution is not yet available. For example, by restarting a failed Configuration Item. Workarounds for Problems are documented in Known Error Records. Workarounds for Incidents that do not have associated Problem Records are documented in the Incident Record
254	**Workload**	The Resources required to deliver an identifiable part of an IT Service. Workloads may be Categorized by Users, groups of Users, or Functions within the IT Service. This is used to assist in analyzing and managing the Capacity, Performance and Utilization of Configuration Items and IT Services. The term Workload is sometimes used as a synonym for Throughput.

User Description

S.No.	Profile	Description
1	**Administrator**	Has the rights on everything (bypassing any control)
2	**Change Approver**	Person who could be impacted by some changes
3	**Change Implementor**	Person executing the changes
4	**Change Supervisor**	Person responsible for the overall change execution
5	**Configuration Manager**	Person in charge of the documentation of the managed Cis
6	**Document author**	Any person who could contribute to documentation
7	**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.
8	**Problem Manager**	Person analyzing and solving the current problems
9	**Service Desk Agent**	Person in charge of creating incident reports
10	**Service Manager**	Person responsible for the service delivered to the [internal] customer
11	**Support Agent**	Person analyzing and solving the current incidents

1. IT Service Management

The IT Service Management (ITSM) solution provides scalable workflows to manage and deliver IT services to your users all through a single cloud-based platform.

The ITSM solution can help increase your agents' productivity, resolve issues quickly, and improve user satisfaction. Also, powered by platform native AI, you can quickly accelerate technology changes and view recommended actions for incoming tickets or requests and drive self-service and automation through enterprise chatbot technology. The NOW Platform also provides users access to ITSM via mobile or web-portal interfaces.

1.1. Features

Enhance the service experience
Automate support for common requests with virtual agents powered by natural language understanding (NLU). Engage users with natural, human conversation to provide consistent good customer service experiences.

Consolidate IT services
Rapidly consolidate existing tools to a single system of action in the cloud.

Improve IT productivity
Use machine learning algorithms and virtual agents to automate routine tasks and increase the productivity of agents.

Gain visibility into processes and services
Get full visibility into any process or service with built-in dashboards and analytics providing real-time, actionable information to improve service quality.

Provide mobile connectivity
Enable employees to find answers and get work done from a modern mobile application.

1.2. Modules of tbITSM v.1.0

1. Service Configuration Management
2. Service Request Management
3. Incident Management
4. Change Management
5. Problem Management
6. Knowledge Management
7. Project Management
8. Knowledge Management

2. Introduction to tbITSM v.1.0. Service Desk Module

tbITSM v.1.0. Service Desk is a fully featured ticket management system which allows enterprises to implement strong and effective work-flows for internal and customer-facing support departments. The support ticketing management system allows help desk staff to keep track of tickets raised by users, attend to them, reassign them to appropriate department or organizations, generate reports and more. The Staff Panel allows the support personnel to view and manage tickets, generate reports, add new users, add and edit organizations, reassign tickets and more.

2.1. Key Features

Dashboard – At-a-glance overview of ticket activity over time. The charts and tables on the dashboard allow you to quickly view the overall status of your operation. The dashboard also provides a starting point from which you can deep-dive into more detailed areas.

Customer Web Portal – No account registration is required for users to submit tickets. Service Desk allow send-users to view their ticket thread by logging in with their email address and ticket number.

Ticket Creation – Allows users to raise tickets in multiple ways – via email, over phone to staff members and web portal as a guest or as a registered user.

Service Level Agreements (SLA) – Easily define service level agreements and configure ticket due dates and warnings to match.

Help Topics - Automates the process of ticket allocation to specified department for faster response. Configurable in the web portal, help topics when combined with custom forms can be used to gather more information from users to assign tickets to related department. Help Topics can be configured with a basic workflow, allowing automated re-assignment of tickets to different departments/staff at successive stages, with different SLAs.

Custom Forms and Fields – Create custom forms designed to gather the precise data your support department needs from end-users in order to solve an issue. Configurable lists can be added to custom forms to make selections more issue specific. Custom Forms and Fields used in help topics for web tickets can be used to automate the process of ticket assignment to correct department and staff member to cut down response time.

Ticket Filters – Ticket filter rules which ensure tickets matching certain criteria are assigned to specific departments/staff members or have default actions applied to them. Ticket filter actions include reject ticket, auto-assign, canned response and more.

Ticket Assignment and Transfer – Tickets can easily be reassigned to different departments and staff members as required. Internal logs are created to keep track of ticket transfers.

Auto-Responder – Configure automatic replies sent to user for new ticket creation. Personalize the auto-responders by inserting variables such as first name and more.

Knowledgebase – Post solutions to frequently asked questions (FAQs) and create articles to solve problems. Knowledgebase items can be internal or publicly viewable and can help with ticket resolution times and can reduce the number of tickets submitted. Resolved tickets also can be posted in Knowledgebase and linked with help topics.

Projects - Staff members can create Gannt charts to plan and track progress on company projects. Individual tasks and sub-tasks can be assigned completion percentages, time spent and billable time. Service desk tickets can also be bound to any task, meaning any updates in the ticket regarding completion, time-spent and billable time will be dynamically reflected in the parent task.

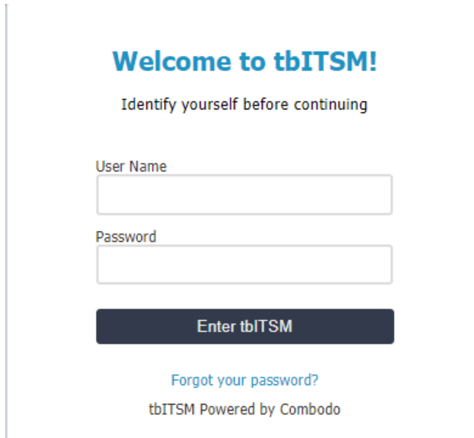
Time Sheets - Time sheets allow administrators to evaluate the performance of the staff and track their working hours. Staff Members can create and submit time sheets in the 'My Time Sheets' interface of the staff panel. Admins can review and approve time sheets in the admin interface by clicking 'Staff > Time Sheets'.

2.2. Connecting to tbITSM v.1.0

Log in screen

The first screen displayed by tbITSM v.1.0 is the login screen, Welcome to tbITSM v.1.0. Each tbITSM v.1.0. user must be identified before accessing the application.

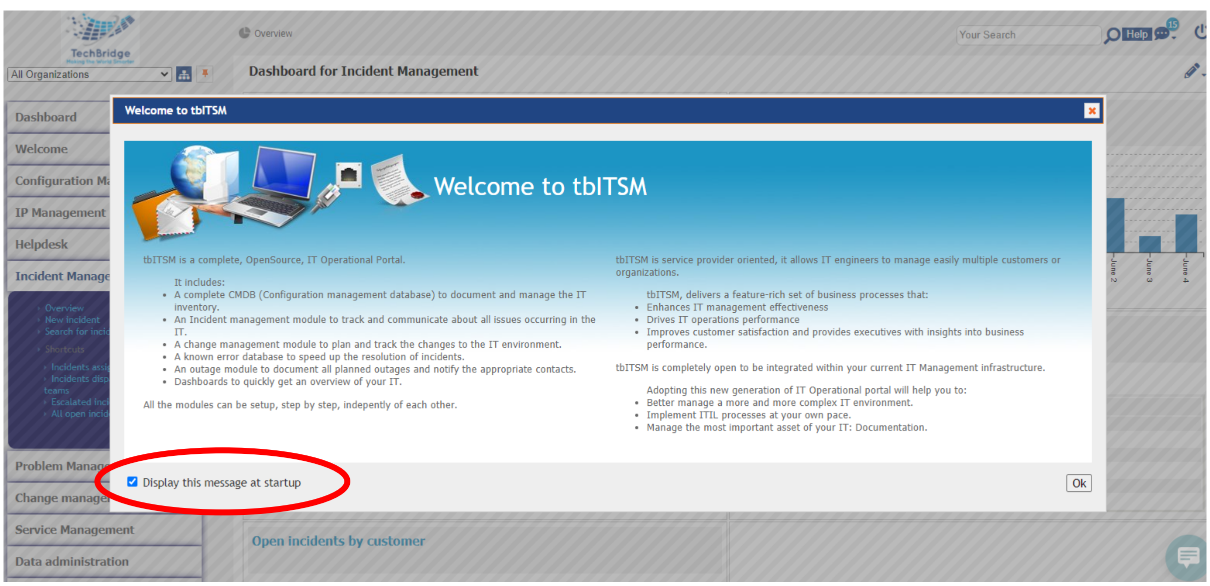
Enter your User Name and Password given by your administrator, and click on Enter tbITSM v.1.0.



2.2.1. First connection to tbITSM v.1.0

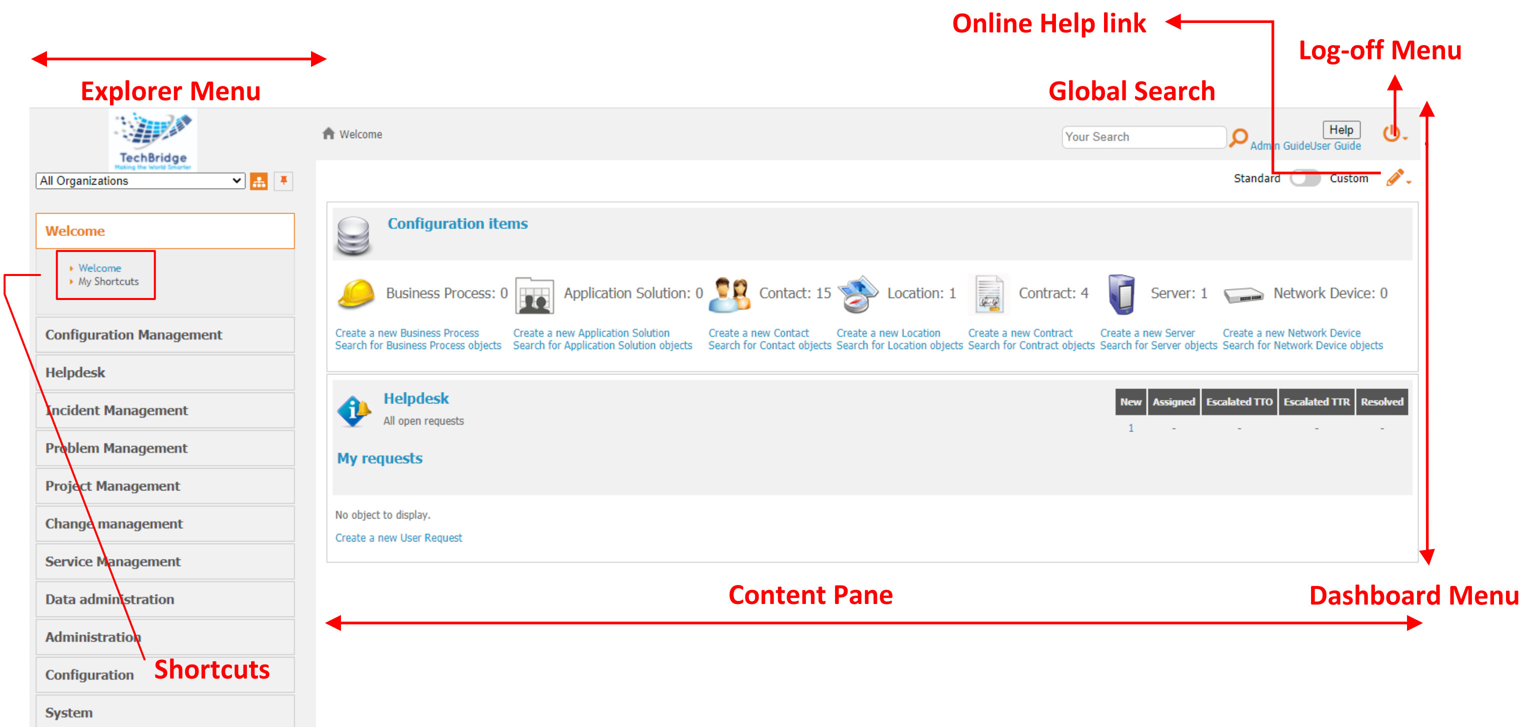
The first time you connect to tbITSM v.1.0, the following Welcome dialog is displayed:

To skip this dialog and go straight to the main screen of the application the next time you connect to tbITSM v.1.0., uncheck **Display this message at startup**, before clicking on the **Ok** button to close the dialog.



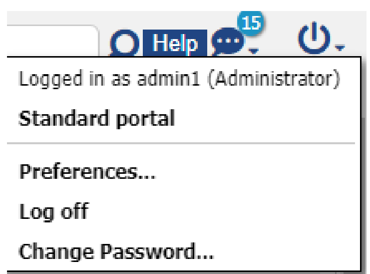
2.2.2. The Main Screen

The picture below shows the main screen of the application, it contains all the standard elements of the tbITSM v.1.0. screens:



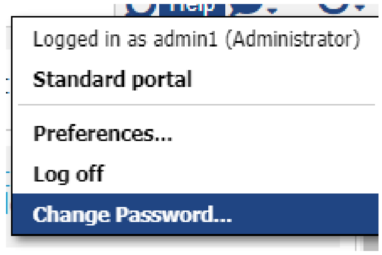
2.2.3. The Log-off Screen

At the top-right of the page, a popup menu is accessible by clicking on the On/off icon:

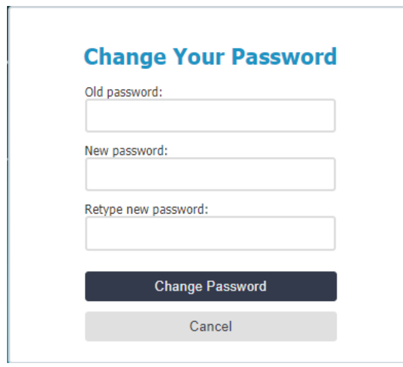


- The first line of the popup displays the name of the user currently logged-in. If this user is an administrator for the application, this is also shown on this line.
- Other lines in bold typeface are Actions. Clicking on them will launch the action corresponding to this menu item.
- Standard Portal is proposed if the user is allowed to switch from the Console to the Portal.
- Preferences is for managing your user preferences in the application (language, favorite organizations, etc.)
- Log off is for terminating your session with tbITSM v.1.0.
- Change Password is for changing your tbITSM v.1.0. password

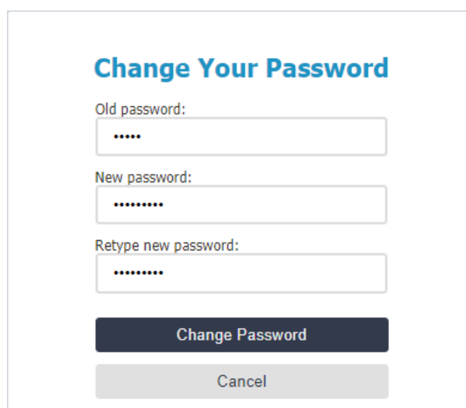
2.2.4. Changing your Password



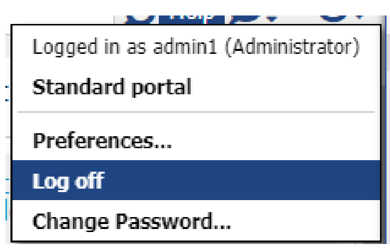
Click on the **Change Password** menu item, to display the change password form:



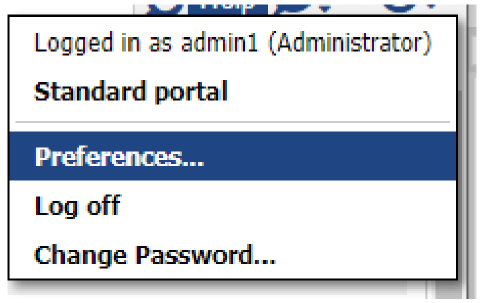
Type the Old password in the Old password field. Next, in the New password field below, type the new password. Retype the new password in the third field. When this is done, click **Change Password** to validate your change.



2.2.5. Disconnecting from tbITSM v.1.0.

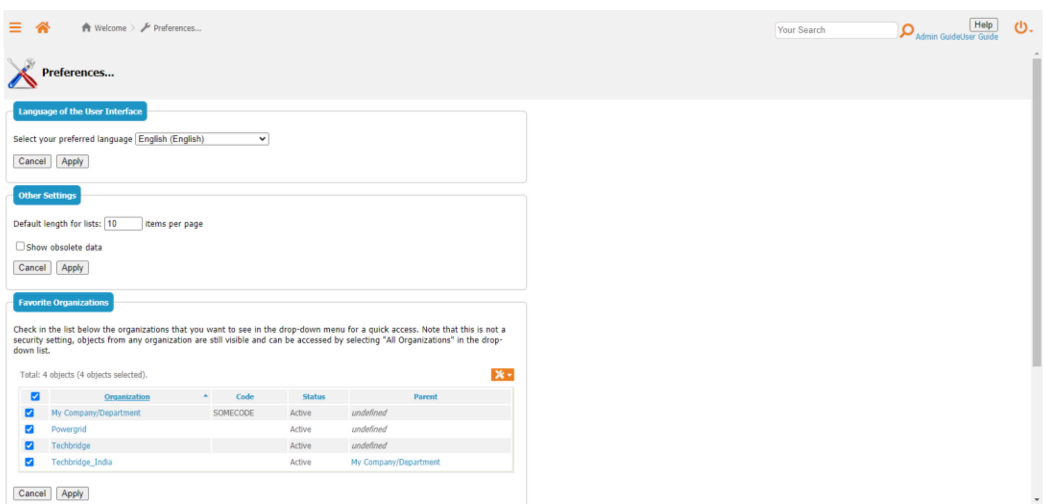


2.2.6. Managing Your Preferences



The Preferences menu allows each user to configure:

- The language for the tbITSM v.1.0. user interface
- The default length for displaying any list of search results
- The option Show obsolete data unchecked by default, allow to display obsolete objects in drop-down lists and search results
- The list of favorite Organizations to be displayed in the context menu
- The Personal shortcuts created by the user



- **Languages**
  To change the language of the user interface, select the desired language from the drop-down list and click on Apply.

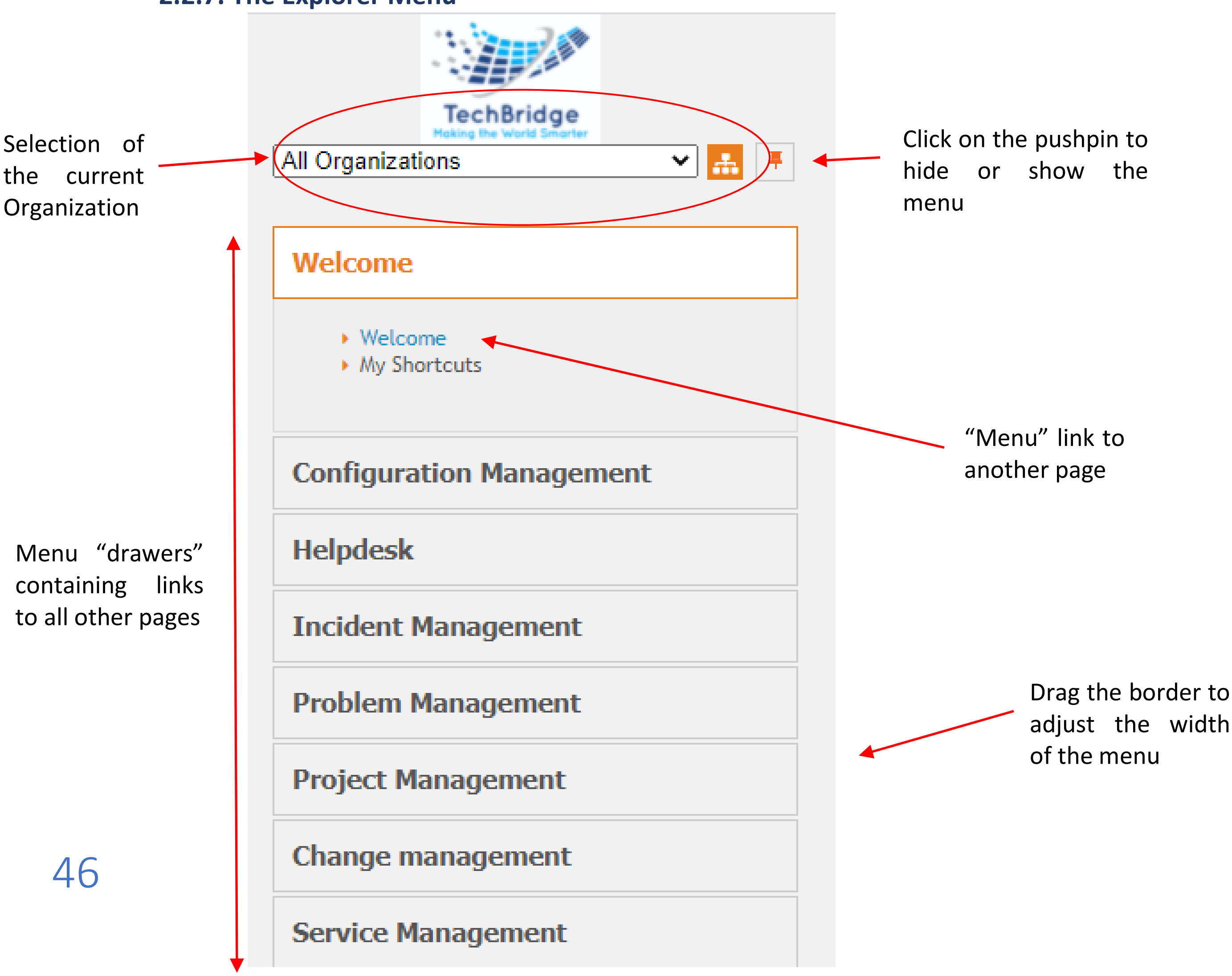
- **Other Settings**
  The Default length for lists is used to limit the display size of the lists of objects; for example, when performing a search. If the number of elements in the list exceeds this value, the list will display the objects by page. Each page is configured by default to display the default length number of objects. Displaying more objects per page may be more convenient but will slow down the display.

This length is also used to limit the history of CSV imports (truncated view).
To modify the default value, change the number then click Apply.

- **Favorite Organizations**
  In large companies, not all users are interested by all organizations documented in tbITSM v.1.0. (for example, if a given user works only for a given set of customers). This setting let each user tailor the list of organizations that appear in the context menu at the top-left of their tbITSM v.1.0. pages, in order to ease the navigation in the application.

- **Shortcuts**
  Use this to rename or delete user-created shortcuts. The shortcuts appear in the main menu immediately below the Welcome menu. To rename a shortcut, check the checkbox in front of its name and click on Rename. To delete shortcuts, check the checkboxes in front of their name and click on Delete (A popup confirmation message is displayed)

2.2.7. The Explorer Menu



In order to optimize the space on screen, the size of the menu may be adjusted by dragging its right border. You can also completely hide the menu by clicking on the pushpin icon at the top-right. Once the menu is hidden, move the cursor of the mouse next to the left border of the page to make the menu appear again. Click on the *pushpin* again to pin it on the page, or let the menu disappear again when you move the mouse out of it.

2.2.8. Organizations

Many of the objects managed in tbITSM v.1.0. belong to an Organization. Organizations are the main means to arrange objects in tbITSM v.1.0. For example, the access control is based on Organizations: a given user may be allowed to access only a given set of Organizations. Moreover, the Organizations Selection menu at the top of the menu allows the end-user to limit the current display to the context of the given Organization. Once an Organization is selected, this “context” information is kept from page to page. To reset the context and display again all tbITSM v.1.0. objects, simply select “All Organizations” in the drop-down list.

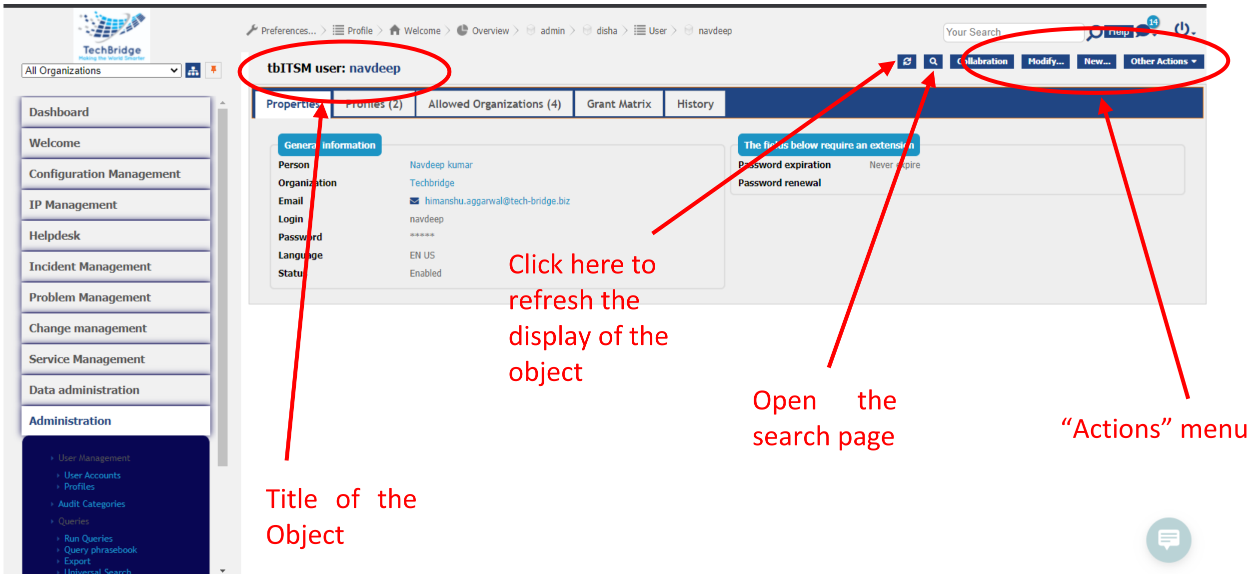
2.2.9. Menu Drawers

The main menu organizes the tbITSM v.1.0. features into several *drawers*, one per process (or per area of interest). Click on the ***Title of a drawer*** to toggle the display of its contents. Only one *drawer* is open at a time.
Inside each **drawer**, a number of links give access to the various features of tbITSM v.1.0.

2.3. Display and Modification of an Object

2.3.1. Display

The details of an object are always displayed in the same manner in tbITSM v.1.0., as shown below:



The title area contains the icon representing the given class (or type) of the object followed by the class (**Person** in this case), followed by the actual name of the object.

The blue **Search** tab (at the top of the page above the title), toggles the display of the search form for this class of object.

The actions allowed for this object are listed as orange buttons (at the top-right of the page). The most frequent actions are displayed as buttons. The remaining actions are accessible via the popup menu **Other Actions**.

The **Properties** tab displays the actual properties of the object.

The **History** tab displays the history of all modifications performed on this object (*who* modified *what* and *when*).

Depending on the class of the object displayed (*Tickets* for example), an extra **Notifications** tab may be available, showing all the notifications sent in response to events on this object.

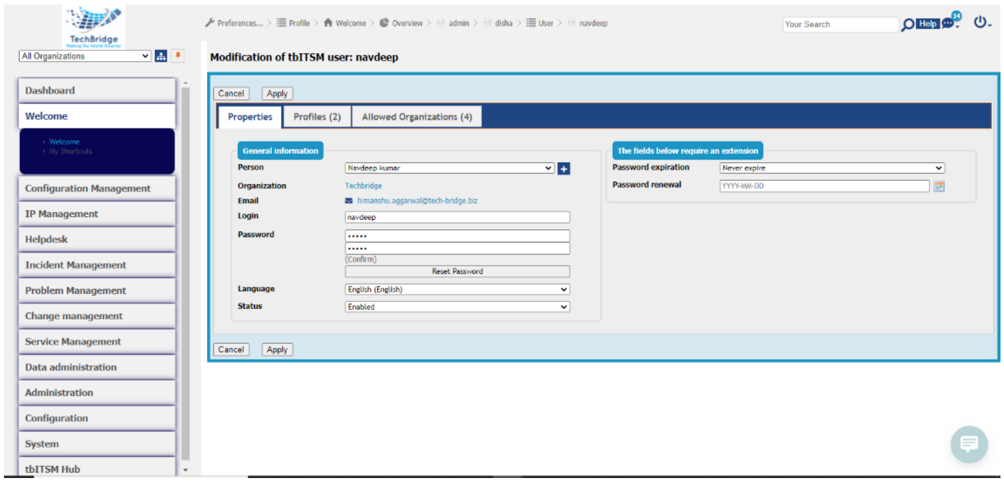
The other tabs display the relationships between the selected object and other objects in the database.

For example, the tab **Teams** displays all the teams for which the selected person is a member of. The **Tickets** tab displays the tickets for which the selected person is the *requester*.

2.3.2. Modification of an Object

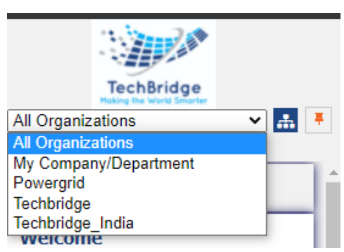
To modify an object, click on the ***Modify*** action, from the details of the object (The same action is also available on a list containing only one object).

The form used to modify an object is quite similar to its details. The only difference is that the fields are editable. For example, the form to modify a *Person* is displayed as shown below:



Note the small red exclamation mark icon (❗) next to the First Name. This icon appears when a mandatory field is left blank, or when a field does not follow the expected format for the field (for example for an email address). The modifications to the object can only be saved (using the Apply button) if all the fields are properly filled.

For some of the fields, the value is not a free-text but instead must be picked from a drop-down list:

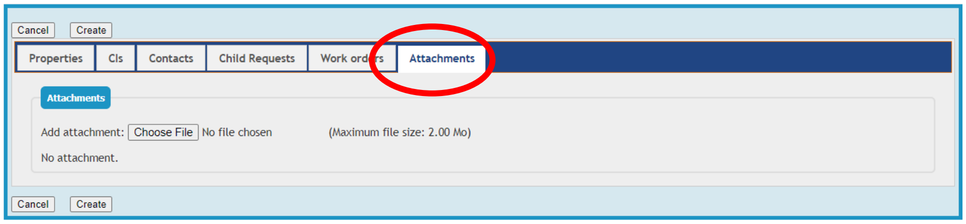


The tree view button ( ) is available for values that are organized as a hierarchy.

2.3.3. Handling Tickets Attachments

Attachments allow to upload a document (any file or image) into tbITSM v.1.0. and "attach" it to a given object in tbITSM v.1.0. This is very similar to email attachments.

To manage the attachments to a Ticket, click on the **Attachments** tab while creating or modifying a Ticket:



To upload a new attachment, click on the ***Browse*** button and select the desired file. Note that the maximum size for a single uploaded file is displayed next to the **Browse** button. If a file is bigger than this value, it cannot be uploaded into tbITSM v.1.0. This maximum size depends on several settings on the server and can be changed only by an administrator.

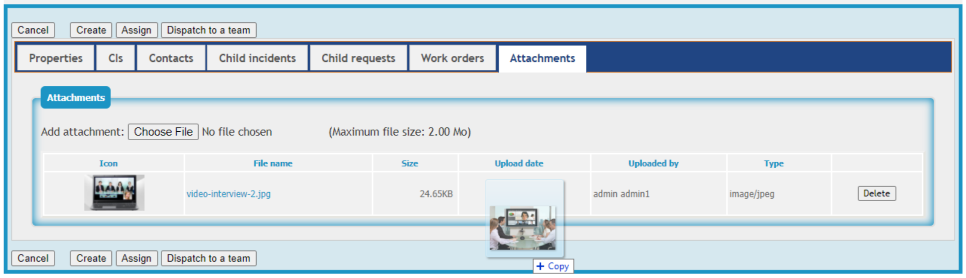
Once attachments have been uploaded, they are listed in the tab, with one icon per file. The name of the attachment is a hyperlink to download it.

Move the mouse above an icon to display the "Delete" button. Clicking on ***Delete*** will then remove the attachment from the ticket.

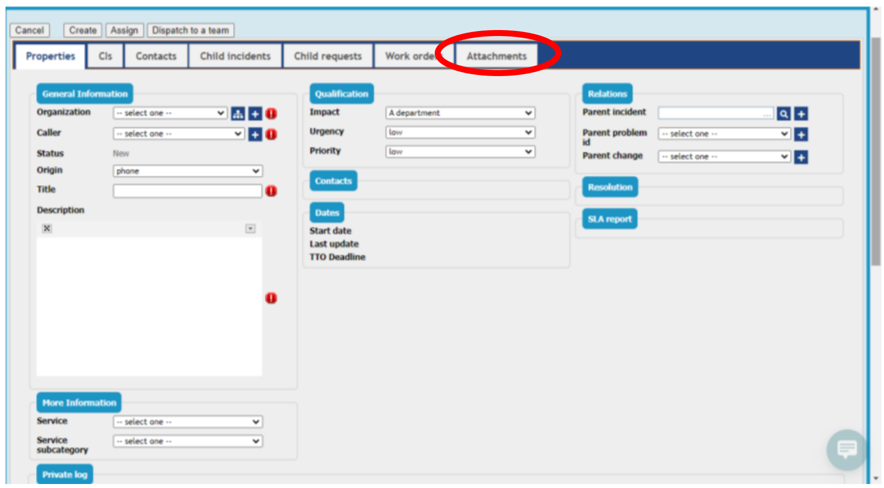
When the attachments are an image, then a preview will be shown anytime the mouse moves over the attachment icon.

2.3.5. Drag and Drop

Attachments can also be added by performing a drag drop of the files to attach, directly over the browser's window. When files are dragged over the "Attachments" tab, a blue shadow appears around the area where attached files are normally displayed:



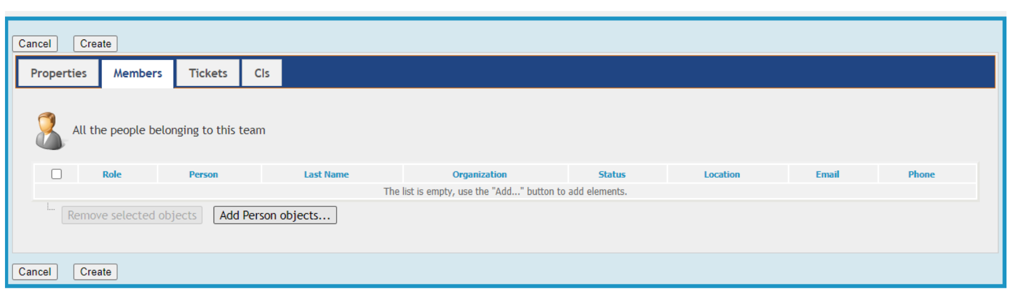
If the "Attachments" tab is not the active one when dragging, files can still be attached to the ticket and the tab itself shows the blue shadow during the drag operation:



2.3.6. Managing Object Relations

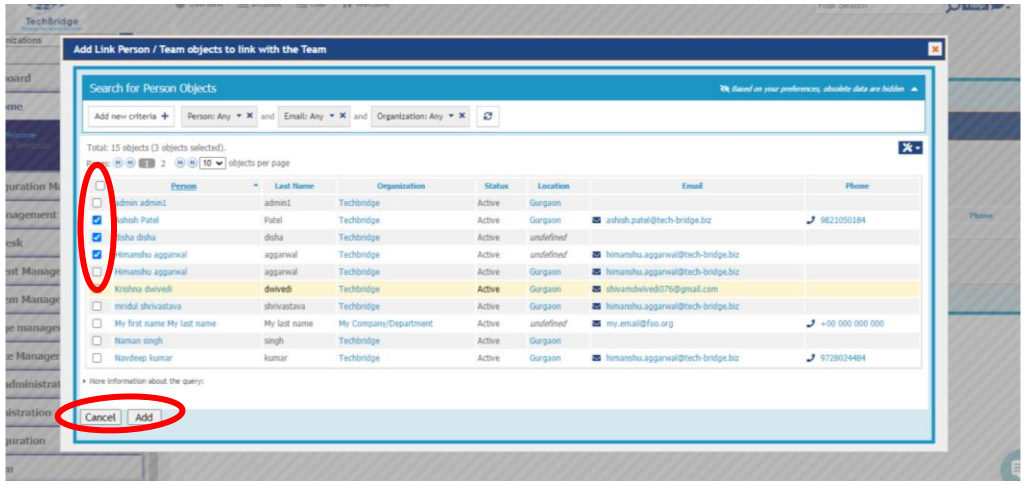
In tbITSM v.1.0. most of the tabs displayed on the details of an object represent relations between objects. All these relations are managed in the same manner.

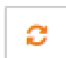
For example, the form to manage the members of a "Team" looks like as shown below:



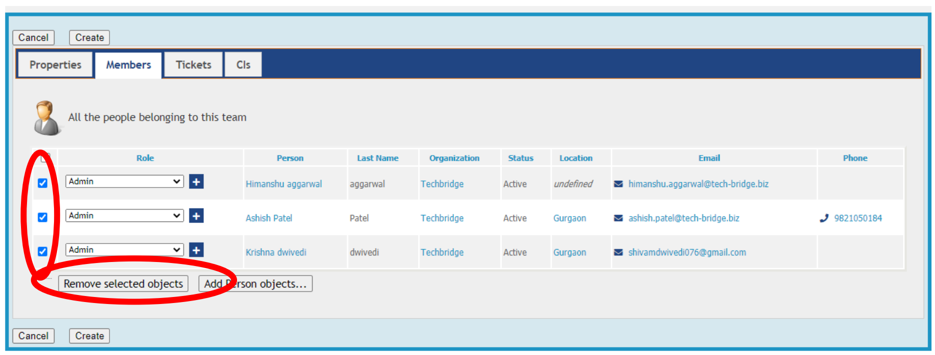
In order to add members to the team, click on the "Add Persons…" button at the bottom. These displays - as a popup - the search form to select

Persons. Check the desired persons in the list and click “Add” to go back to the main form.

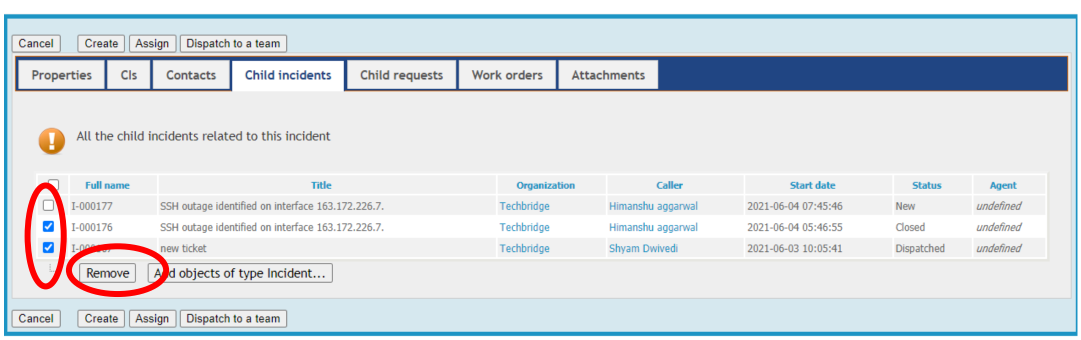


You can narrow the search using the criteria at the top, then click on the  button to refresh the list.

To remove objects from the relation, check the corresponding line in the form, then click on the “Remove selected objects” button at the bottom.

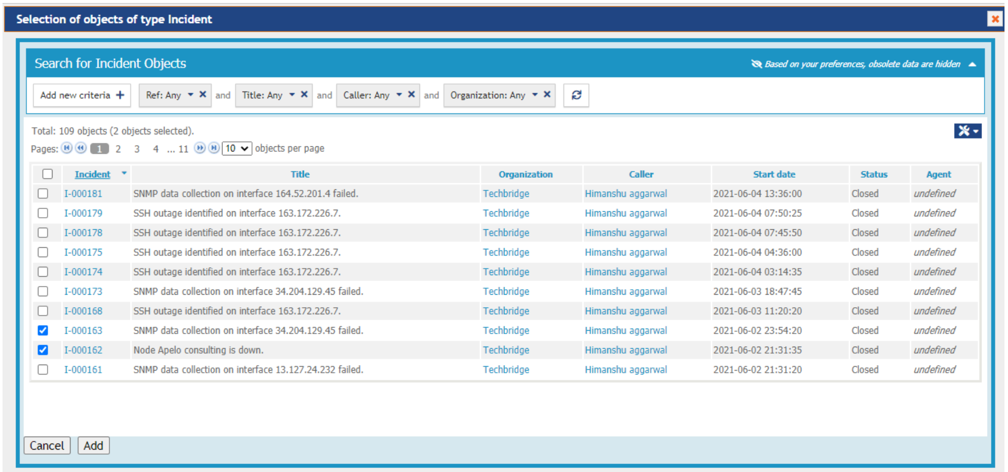


On Tickets, a slightly different type of relation is available where the related objects are other tickets.



To detach related tickets from the current ticket, just select them by checking the corresponding checkbox and click on “Remove”.

You can use the “Add objects of type…” button to relate other Tickets to the current Ticket. When clicking on the button, a popup dialog is displayed with the standard search form at the top and a list with check boxes at the bottom.



You can use the  button to refresh the list after changing the filtering criteria at the top. Make your choice by checking the corresponding checkboxes, then press the “Add” button to dismiss the dialog.

2.4. Actions

The actions are used to modify objects in tbITSM v.1.0. Actions can be executed:

- from a list of objects, in which case the selected action is applied to all the objects in the list
- from the details of an object
- The actions buttons are only displayed to users with sufficient rights to execute them.

The standard actions in tbITSM v.1.0. are:

2.4.1. New

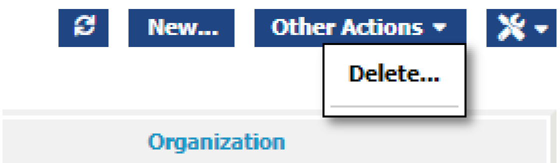
This action is used to create a new object of the same type as the object currently displayed. When clicking on “Create…”, the creation form of the object is displayed.

2.4.2. Modify

This action displays the form to update an object. Both creation and modification forms are very similar, except that when modifying an object, the values in the form are prefilled with the current values of the object being modified.

2.4.3. Delete

The **Delete** action is located in the **Other Actions** popup menu:



This action is used to permanently delete an object from tbITSM v.1.0. In order to guarantee the consistency and integrity of the data in tbITSM v.1.0, deleting an object may also affect other objects. Depending on the rules defined in the data model, other objects may be modified or deleted automatically, or the deletion may not be possible until some depending object is deleted manually.

In any case, a confirmation screen listing all the impacted objects, is displayed.

Deletion of 13 objects of class User

126 objects/links are referencing some of the objects to be deleted

To ensure Database integrity, any reference should be further eliminated

Class	Object	Consequence
tbITSM user	admin	
tbITSM user	admin1	
tbITSM user	AshishPatel	
tbITSM user	disha	
tbITSM user	gunjan	
tbITSM user	himanshu aggarwal	
tbITSM user	krishna	
tbITSM user	naman	
tbITSM user	navdeep	
tbITSM user	rajoli	
tbITSM user	sahil	
tbITSM user	shashi	
tbITSM user	shyam	
User Preferences	1	
User Preferences	13	
User Preferences	2	
User Preferences	6	

For example, if you want to delete the whole organization, the following confirmation screen is displayed:

Deletion of My Company/Department

9 objects/links are referencing My Company/Department

To ensure Database integrity, any reference should be further eliminated

Class	Object	Consequence
Organization	My Company/Department	
Person	My first name My last name	Must be deleted manually, but this is not feasible: Person::1 Must be deleted manually
Team	MyTeam	Must be deleted manually, but this is not feasible: Team::2 Must be deleted manually
Incident	I-000001	
Problem	P-000003	
Problem	P-000004	
User organizations	Link between navdeep and My Company/Department	
User organizations	Link between krishna and My Company/Department	
Organization	Techbridge_India	will be automatically updated (reset: Parent)
Known Error	knownerror1	will be automatically updated (reset: Related Problem)

Please perform the manual operations listed above prior to requesting the deletion of this object

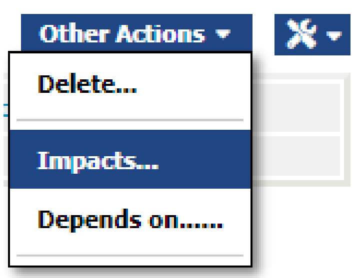
<< Back   Delete !

In this particular case, tbITSM v.1.0. detects that there are several Persons belonging to this Organization. Since the definition of the data model requires that a Person *must* belong to an Organization and that the deletion of a Person is *not automatic*, the **Demo** Organization cannot be deleted while there are Persons belonging to it.

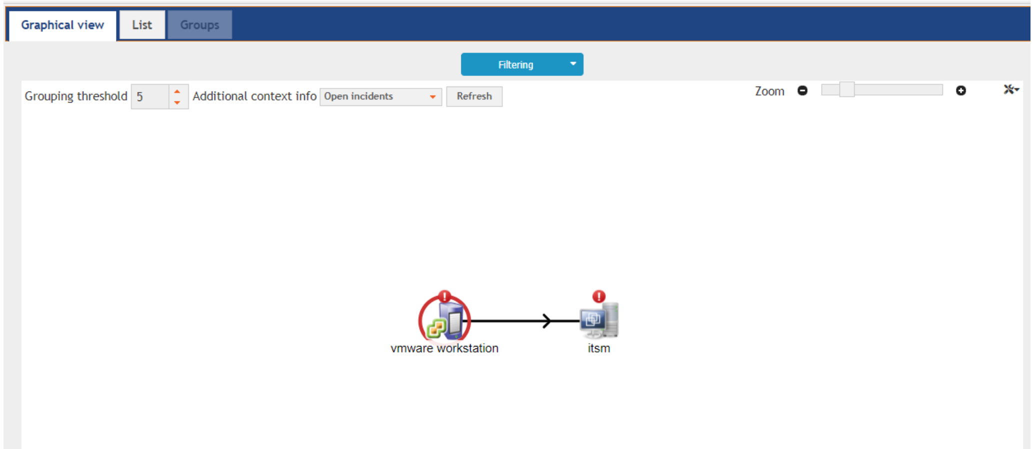
This is why the **Delete!** button is *grayed out*. In order to delete the **Demo** Organization, the Persons must be processed first, either by deleting them or by assigning them to a different Organization.

2.4.4. Relations

Depending on the type of object displayed, additional Relations menus are available. For example, for most Configuration Items, two relations are defined: “Impacts” and “Depends On”. In this case, two extra menus are available in the Other Actions popup menu.



When selecting such an action, the page showing the relation is displayed:

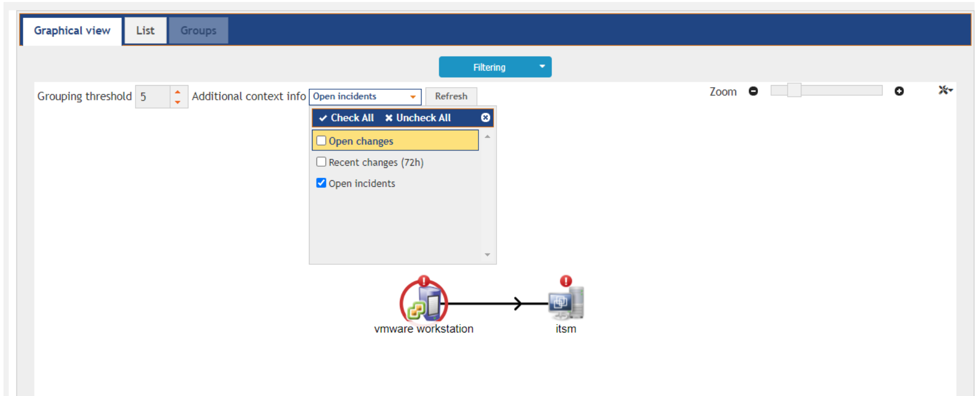


- You can use the checkboxes with the icons at the top, to filter the display (just press **Refresh** after updating the selection).
- Move the mouse over an object to display a tooltip showing additional information (plus a link to its details)
- Click and drag onto an object to re-arrange the chart
- Click and drag in the background to move the whole chart
- The + and - buttons, as Well as the slider at the top right of the chart (and the mouse wheel) are used to adjust the zoom level
- The Grouping threshold defines the minimum number of similar objects (same type, same status) for creating a group which replaces the individual objects in the chart, in order to simplify the chart. When changing this value, click **Refresh** to re-compute the chart with the new grouping level.

2.4.5. Context

The Additional context drop-down list defines how Tickets are taken into account when computing and displaying the chart (just press **Refresh** after updating the selection).

When an object in the chart is linked to (at least) one of the context tickets, a small icon representing the type of the ticket (Incident, Change) is displayed next to the icon of the object. The list of tickets linked to a given object is available in the tooltip for this object.



- Open changes context corresponds to 'Changes' which are neither closed nor rejected, are generating an outage and do impact some of the displayed CIs (Impact! = 'Not impacted')

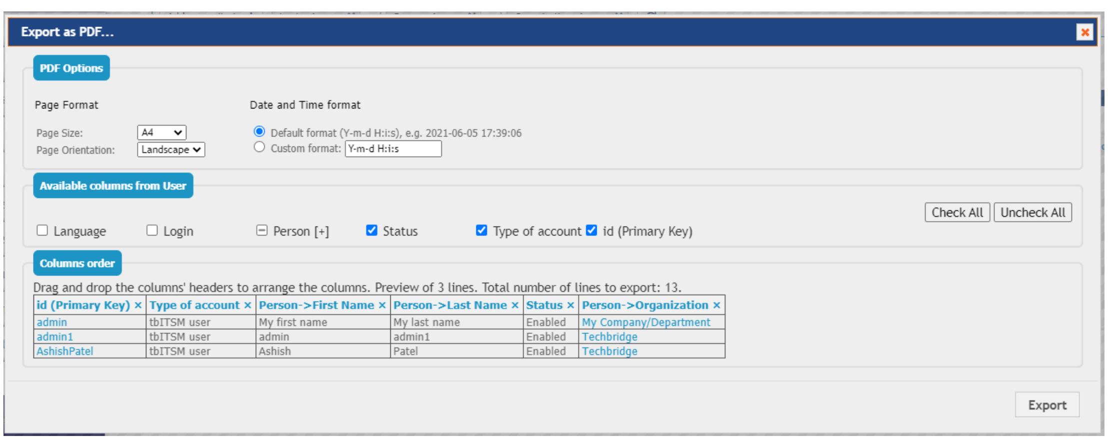
- Recent changes (72h) context corresponds to 'Changes which are closed since less than 3 days, have generated an outage and did impact some of the CIs displayed.

- Open incidents context corresponds to Incidents which are neither closed nor resolved and do impact some of the CIs displayed.

2.4.6. Exporting the chart as a PDF Document

To export the chart as a PDF document, use the Toolkit menu at the top-right of the chart: 

This displays the following dialog box:



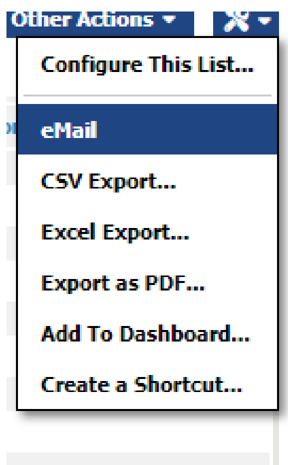
The export generates a printable PDF document where the first page contains the chart (as laid out on the screen), with a legend on the left and an optional comment area at the bottom. The list of objects can be added as a set of tables (one per class of object) on additional pages.

The following options are available:

- **Page format:** the dimensions of the PDF document: either A3, A4 or Letter.
- **Page orientation:** either horizontally (Landscape) or vertically (Portrait)
- **Title:** a title for the document, to be added as a header at the top of each page
- **Comment:** A free text area that will be integrated (if not empty) on the first page of the document, below the chart.
- **Include the list of objects:** if checked, the list of the objects in the chart will be added as a big table as additional pages in the PDF document.

2.4.7. E-Mail

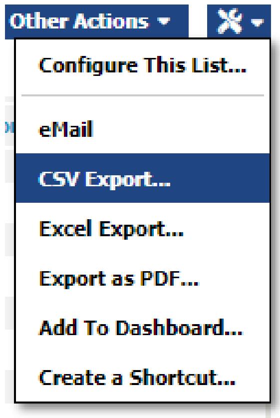
The email action is located in the Toolkit popup menu:



This action creates a new email message containing a link to the selected object in tbITSM v.1.0. The email is created and sent from the client mail program configured on the end-user's computer.

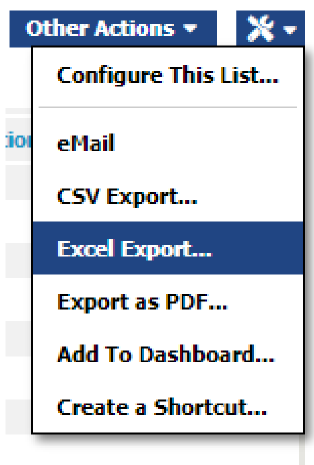
The recipient of the message will receive an URL link to navigate to the details of the object, but she/he must have an tbITSM v.1.0. account to view the information, since she/he will be prompted to authenticate for viewing the page.

2.4.8. CSV Export



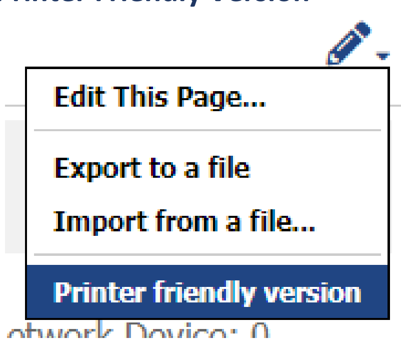
The CSV export of one object is similar to the CSV export of a list, except that there is only one line of data in the resulting export.

2.4.9. Excel Export



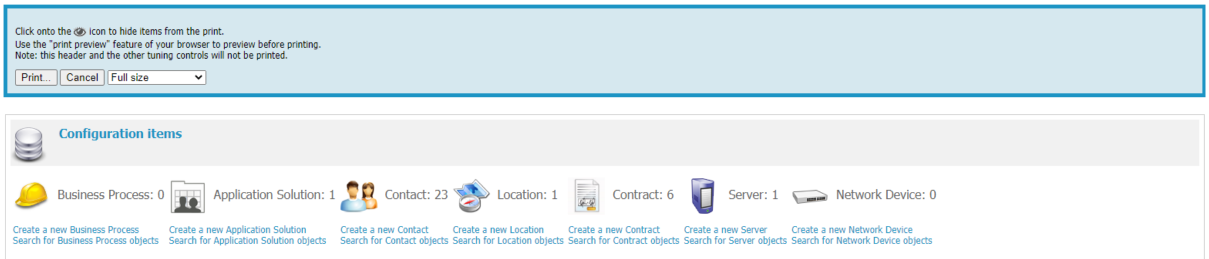
The Excel export of one object is similar to the Excel export of a list, except that there is only one row of data in the resulting export.

2.4.10. Printer Friendly Version



This action opens, in another tab of your browser, a **Printer Friendly** version of the details of the current object. This version of the **details** is optimized for printing: all the tabs are listed as sections inside the page, most of the surrounding elements (menu, search bar…) are removed.

Moreover, the page can also be configured by hiding/showing some of its sections before printing, by clicking on the **Eye** icons.



2.4.11. Ticket Specific Actions

Some objects managed in tbITSM v.1.0. follow a specific life-cycle that correspond to a specific process. This is the case, for example, for the different tickets managed in tbITSM v.1.0. (incident tickets, change management tickets, etc.). For these objects, additional actions are available, corresponding to process execution workflow.

The actions available on each object depend on:

- The life-cycle of the object (each type of object may have a different life-cycle)
- The current state of the object (not all actions are possible in all states)
- The user-rights as defined by her/his profile(s): some actions are restricted to a given profile.

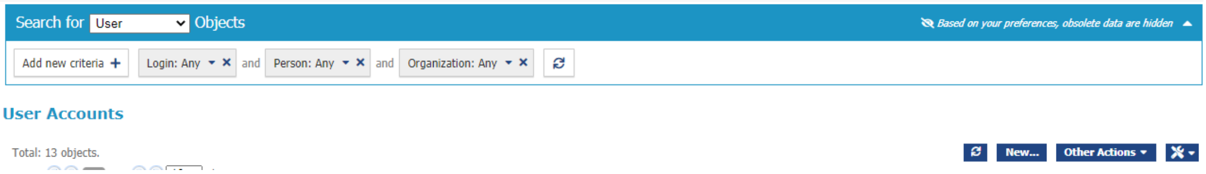
2.5. Search

A search icon (

) is present on the details of most objects in tbITSM v.1.0. Click on this icon to switch to a multi-criteria search form.

All the criterion of the search form are combined using an **AND operator**. This means that when searching for Tickets, if you specify **Demo** as the Organization and **High** as the priority, the search results will list all the Tickets of the

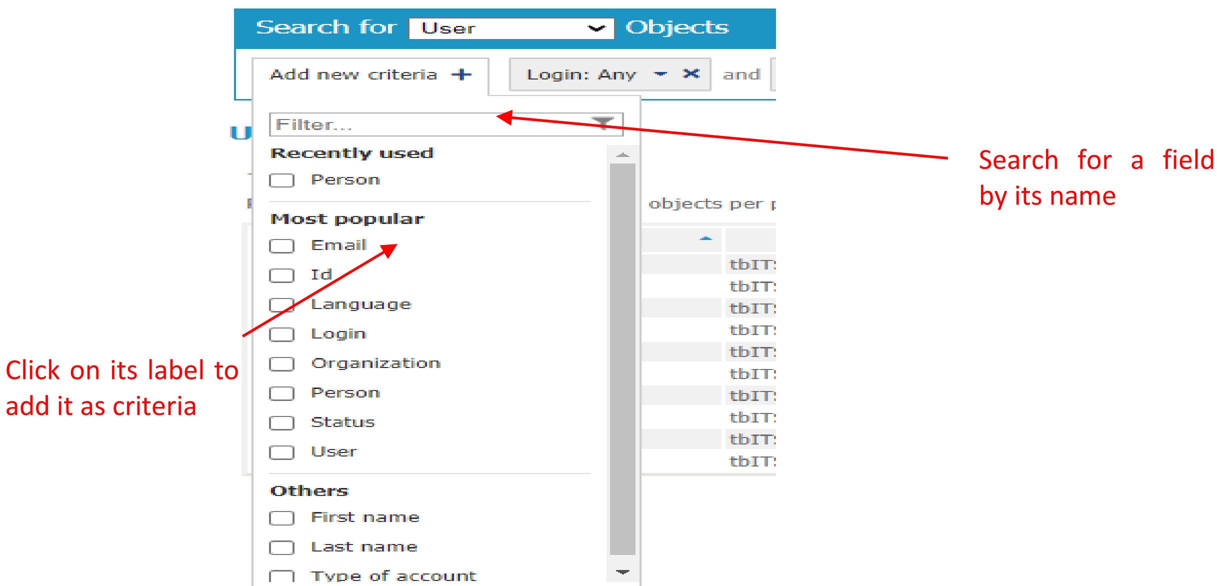
Organization **Demo** which have a **High** priority (Organization = Demo **and** Priority = High).



2.6. Cinematic

2.6.1. Adding Criteria

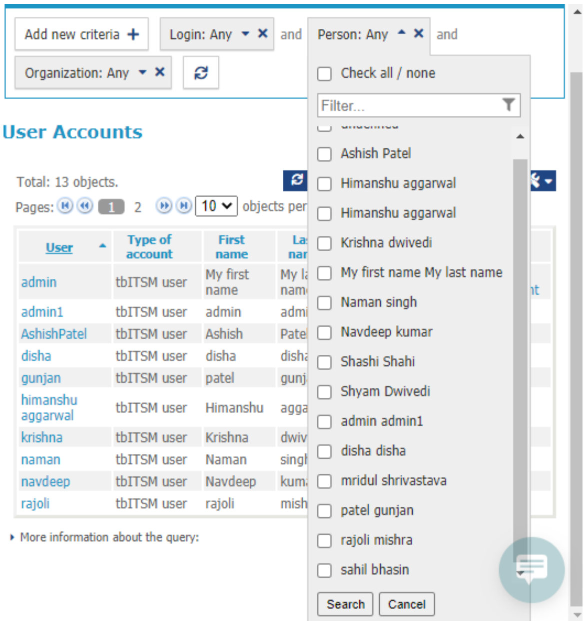
Most search will come with predefined search criterion, but you may want to change them or add new ones



- Open the *Add criteria* box
- You can filter the list by entering the beginning of an attribute
- You can click on an attribute name; it will add it as a filtering criterion and close that box
- Or you can click on the checkbox, it will select it but leave the box open for selecting more criteria, in that case next click even on name will just checked it. Just press a Apply button at the very end of that box.

2.6.2. Setting a Criteria

Open a criteria box. The proposed options will depend on the type of attribute.

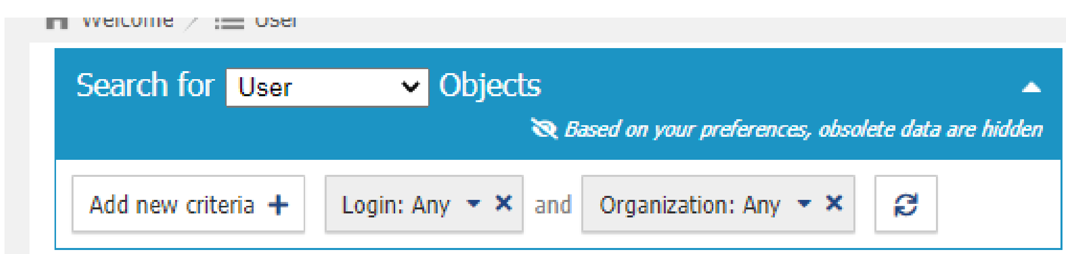


Select an option by:

- entering data in the text area,
- selecting a radio button
- checking one or more choices
- If you want more options, click on *more*
- Once you have set your option, click on button
- *Search* to execute the search
- *Apply* to save your criteria option (This option is only available when "auto-submit" is dis-activated for performance reason)
- *Cancel* to lose your option and entries anywhere in the screen to execute the default action *search* or *Apply*

2.6.3. Removing a Criteria

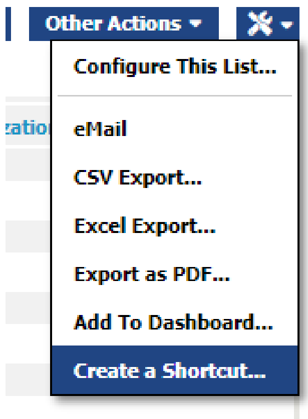
You can remove a criterion by clicking on the red x icon. Note that this removal will in general refresh the list automatically.



By default, a criterion comes with *any*, which means it does not filter the results, so removing it as no effect on the resulting list.

2.6.4. Saving a Search

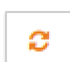
You can save a search with all its criterion, using the *Create a Shortcut…* action on the resulting list.



It will create a shortcut and when you will open it later, all set criterion will be there.

A criterion with *any* is not stored, so is lost, but this does not change the results of your search.

2.6.5. Re-run a search

You can force the execution of the search again with the search button (

).

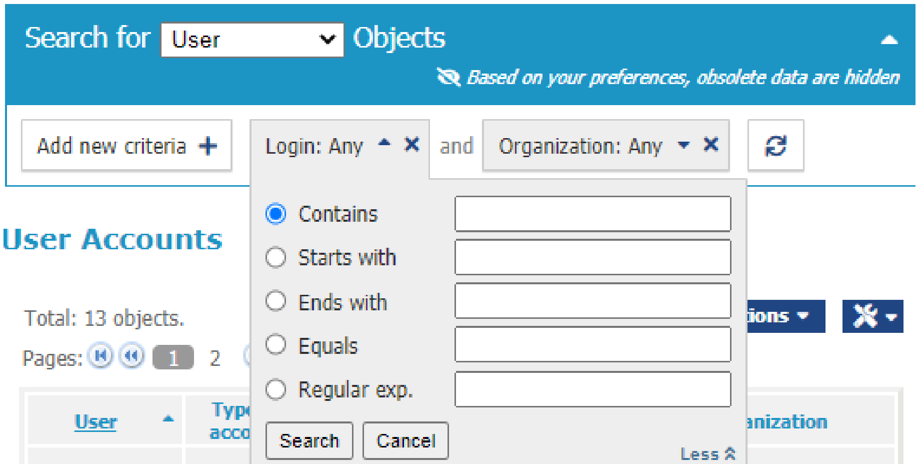
In general, it is useless because any change to a criterion or removing a criterion will refresh the list, but if another object matching the criteria has been created or modified since the search was executed, then the refresh is useful. Another situation where it is needed is when the search for this particular class, due to performance issue, is not automatic.

2.7. Search Capabilities

Here are the search capabilities in details per type of Attribute

2.7.1. Text

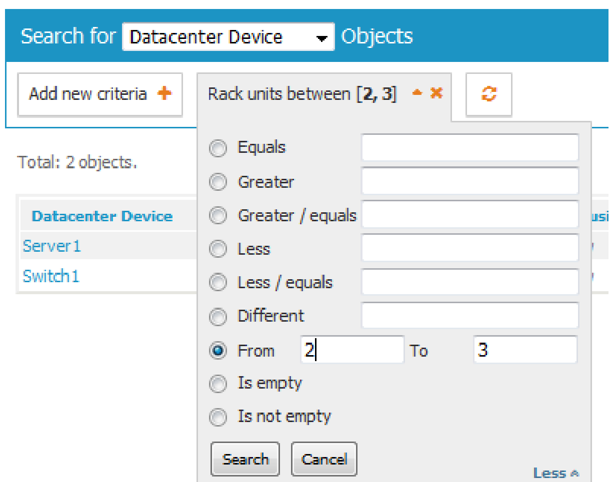
This applies to simple text, html and case-log.



You can even use regular expression (regex).

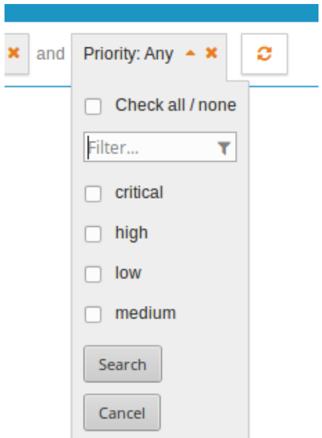
2.7.2. Numbers

You filter on numbers: above, under, between, …



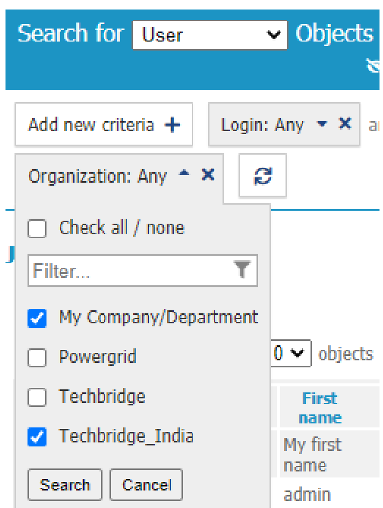
2.7.3. List of Choices

It offers the full list of available values and you can select them: all, none, one or multiple.

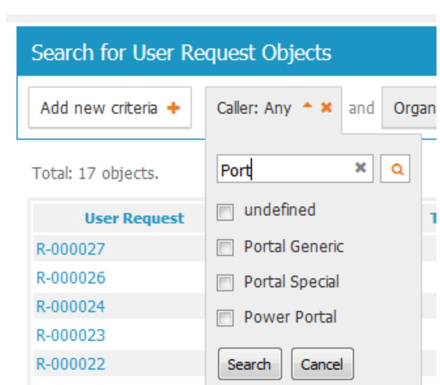


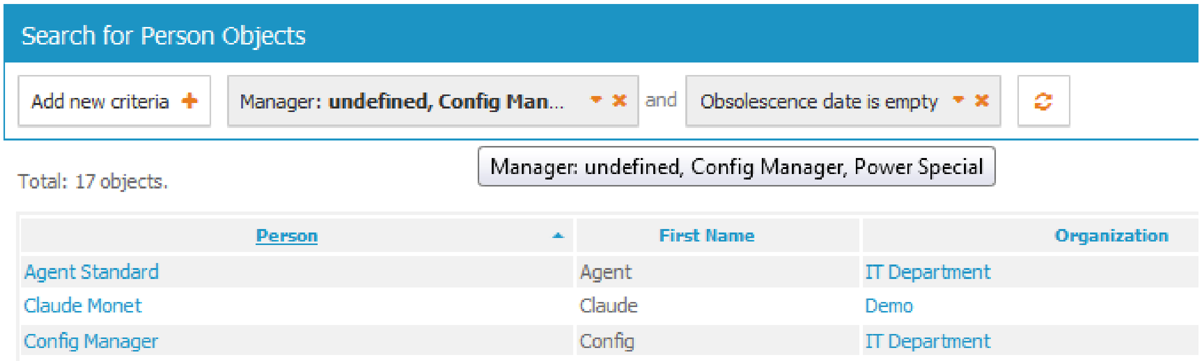
2.7.4. Objects

Depending on the number of objects, it will either propose the full list of objects or an auto-complete mode to retrieve those you need, as you can see in the image, you can select some which are kept while you search for others.

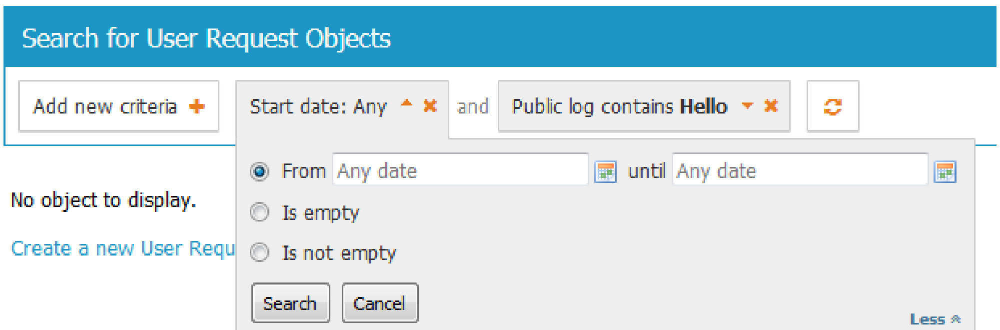


You can combine real values and *undefined* which means not documented (*=0 in OQL*)





2.7.5. Date

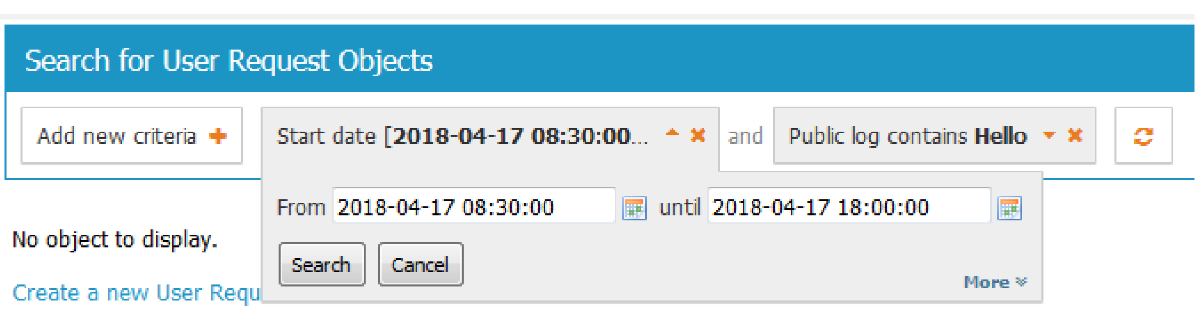


tbITSM v.1.0. uses by default those formats for representing dates and time:

Date: first the year is displayed with 4 digits, followed by a dash, followed by the month expressed with 2 digits, followed by a dash and finally the day with 2 digits. (In short: YYYY-MM-DD)

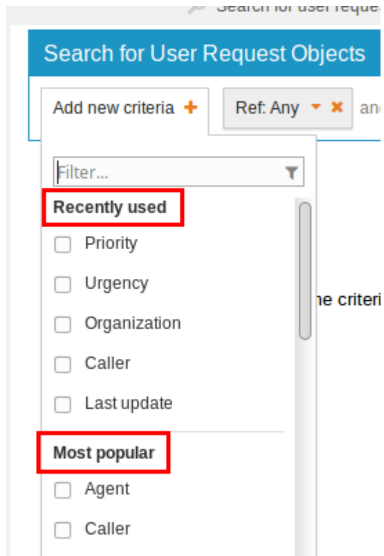
Date and time: start with a date using above format, followed by a space, then the hours with 2 digits (24-hour format) followed by a colon (:), followed by the minutes with two digits followed by a colon (:) and the seconds with two digits. (In short: YYYY-MM-DD hh: mm: ss).

Your administrator, can have defined other format and that can depend on your language.



2.7.6. Recently Used

When selecting search criteria, the first list of up to 5 names are the last criterion that you have used for this class of objects. They are ordered by time, last used being first. They will be there even after closing the browser, as they are stored in your user preference.



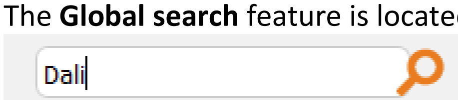
Most Popular are those defined by your administrator as more useful than others. They are in alphabetic order.

Others are the rest of the object's attributes, which are less expected to be used as search criteria, but may if you need.

All names listed in *Recently used* are also listed in one of the other lists.

2.8. Global Search

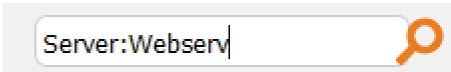
The **Global search** feature is located at the top right of all tbITSM v.1.0. pages.



The global search searches for the given string of text inside all the objects in tbITSM v.1.0. To launch the search, **enter the text** to search for and click on the ***magnifying glass icon*** or press the ***ENTER*** key of your keyboard.

The search results are displayed by groups: all the matching objects of the same class are grouped together in one table.

Since the global search searches in all the objects of the tbITSM v.1.0. database, it can become quite slow when the number of objects increases. However, you can speed-up the search by limiting its scope to a given class of objects. The syntax to specify the scope of the global search is as follows: class_name: text_to_search.



Searching for **Server: webserv** will search for all server objects containing the text "webserv" in any of their fields.

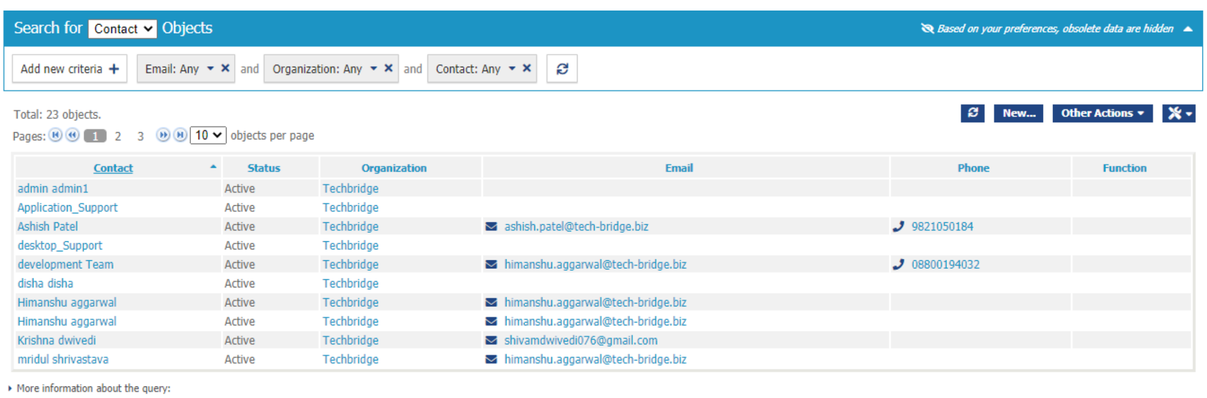
To limit scope, you can also specify X as a class that has subclasses. For example:

Searching for **Contact: dali** will search for all Teams and Persons containing dali in any of their fields.

2.9. Managing Lists

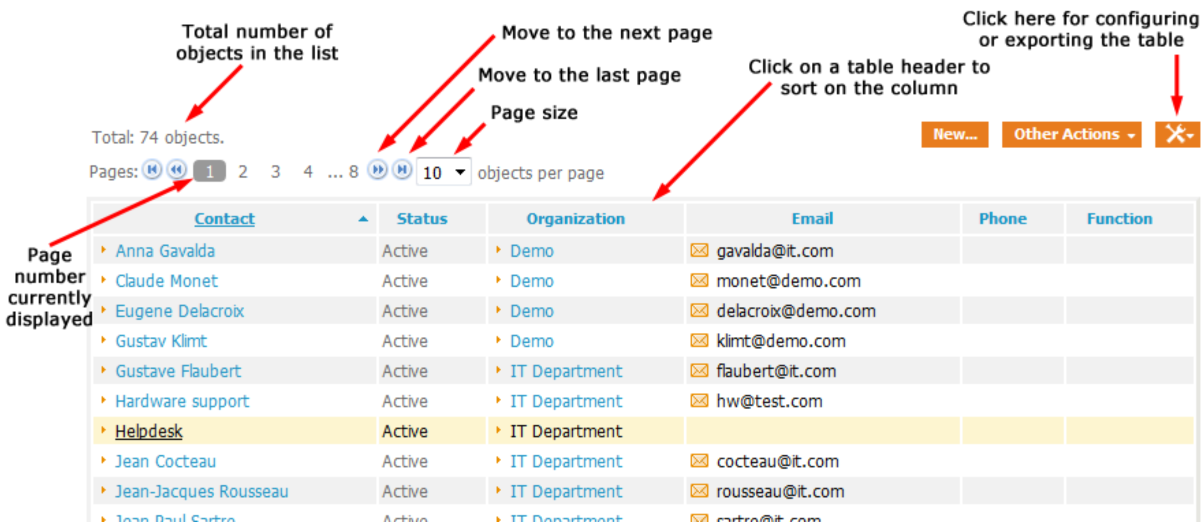
In tbITSM v.1.0, lists are quite common. They can be found in menu items which point to lists, or the search results, as Well as the global search. Once a list of objects is displayed, the operations that can be performed on the list are always the same.

For example, below is a list of Contacts:



Contact	Status	Organization	Email	Phone	Function
admin admin1	Active	Techbridge			
Application_Support	Active	Techbridge			
Ashish Patel	Active	Techbridge	ashish.patel@tech-bridge.biz	9821050184	
desktop_Support	Active	Techbridge			
development Team	Active	Techbridge	himanshu.aggarwal@tech-bridge.biz	08800194032	
disha disha	Active	Techbridge			
Himanshu aggarwal	Active	Techbridge	himanshu.aggarwal@tech-bridge.biz		
Himanshu aggarwal	Active	Techbridge	himanshu.aggarwal@tech-bridge.biz		
Krishna dwivedi	Active	Techbridge	shivamdwivedi076@gmail.com		
mridul shrivastava	Active	Techbridge	himanshu.aggarwal@tech-bridge.biz		

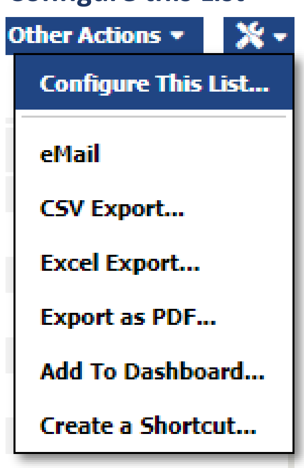
If a list contains more than a (configurable) number of elements, it is displayed as a *paginated* list:



2.10. Toolkit Menu

When a list of objects is displayed in tbITSM v.1.0, a certain number of operations can be performed on the list itself (without modifying the objects in tbITSM v.1.0). These operations are executed through the toolkit popup menu at the top-right of the list.

2.10.1. Configure this List

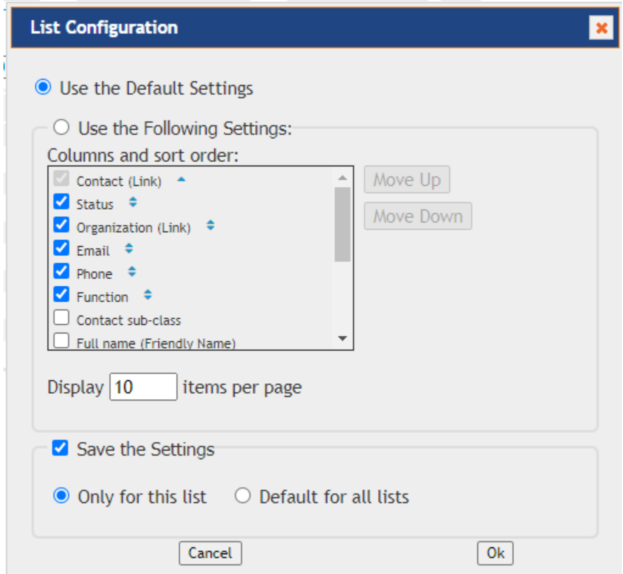


The first possible operation consists of configuring the appearance of the list. You can choose:

- Which columns are displayed and in which order?
- How the list is sorted
- How many items will be displayed at the same time (see paginated lists above)

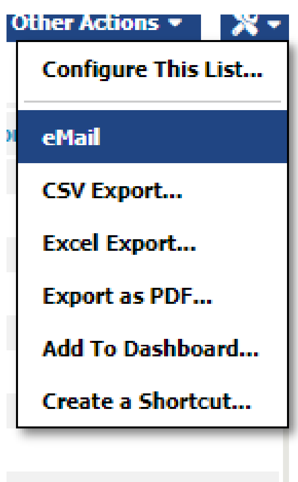
- Which list this configuration will apply to: (Only to this specific list, or to all lists which do not already have a specific configuration).

All this configuration is done via the popup dialog:



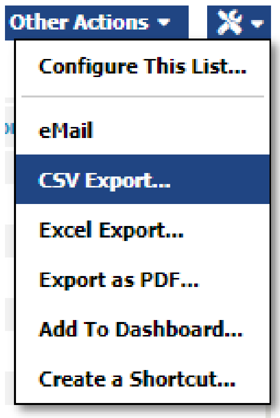
- Advanced case: If your list displays multiple classes of objects (e.g., SELECT a, b FROM Person AS a JOIN UserRequest AS b…), then you cannot save the configuration Only for this list, but when you save it for Default for all lists, it will only impact lists which do select exactly the same classes with the same alias, for e.g., if another list is based on that query (SELECT a2, b FROM Person AS a2 JOIN UserRequest AS b…), then the configuration saved on the first list will not apply.

2.10.2. E-Mail



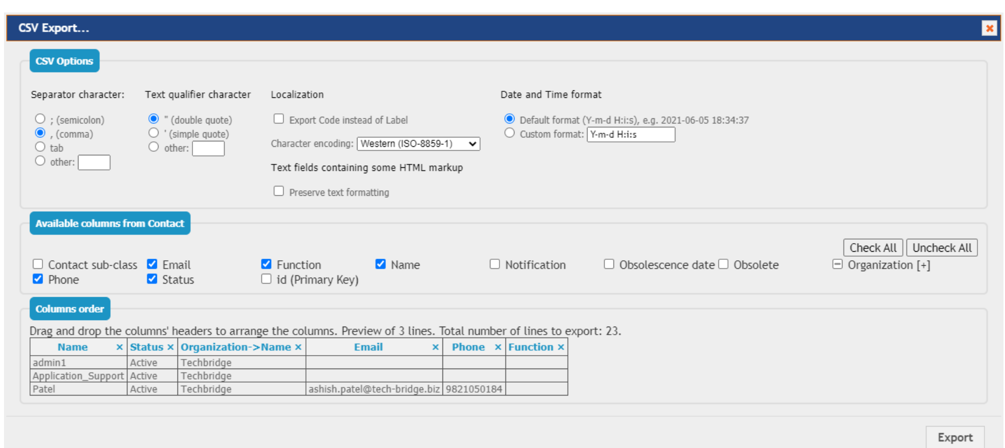
This action is similar to the **email** action on the details of an object, except that the hyperlink inserted into the message points to the specified list of objects.

2.10.3. CSV Export



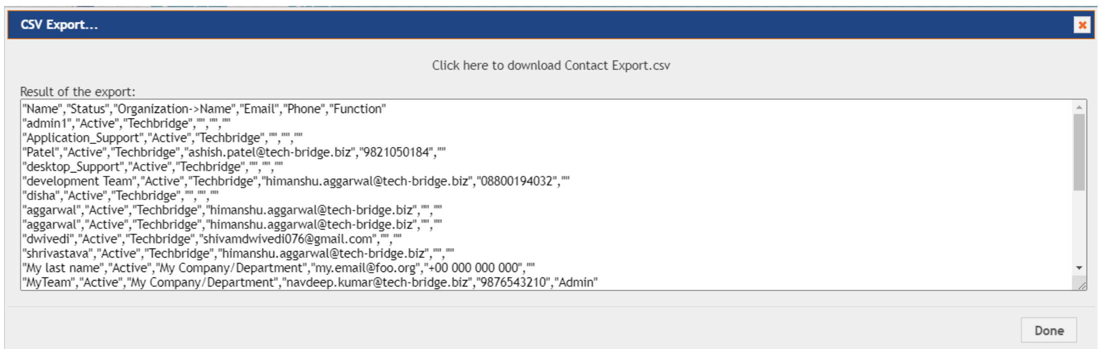
The CSV export action displays a configuration dialog which allows to:

- Select the format options: separator character for the CSV (comma, semicolon, tab or any other character)
- Specify a text qualifier character (to be placed around text strings inside the CSV)
- Pick a character encoding
- Choose, for enumerated fields (like statuses) whether the values must be localized (i.e., use the values as they appear on-screen) or use the internal (language independent) codes.
- Pick the fields to be included in the export
- Arrange the order of the columns



When your export is properly configured, click ***Export*** to build it. If the export contains a large amount of data, (more than 1000 lines) a progress bar is displayed.

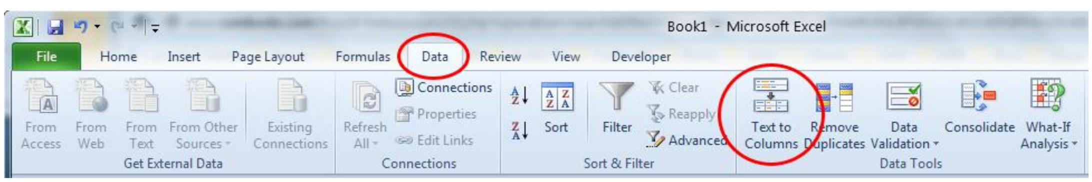
When the export is complete, you can either click on the link ***Click here to download*** at the top of the dialog box to download the result as a .csv file, or copy and paste the export as plain text from the text area:



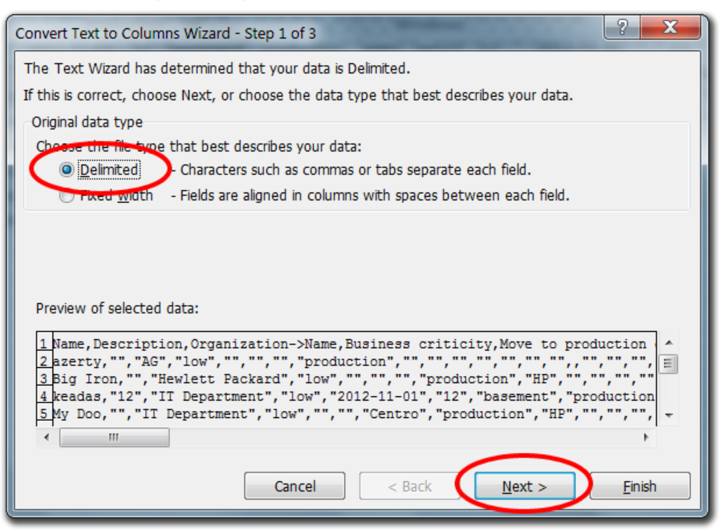
How to paste CSV data in Excel 2007/2010?

If you use copy/paste to import CSV data into Excel, all the data is put in one column only. Fortunately, this is easy to change.

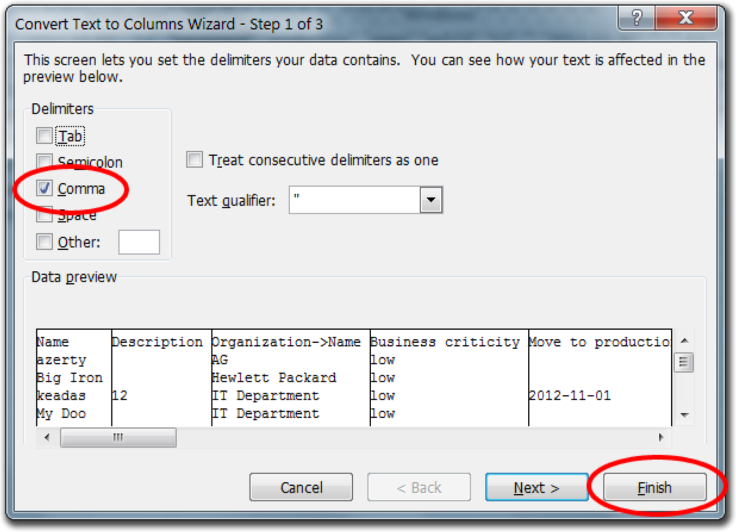
In the Data tab of Excel, click on the **Text To Columns** icon:



The following dialog wizard appears:



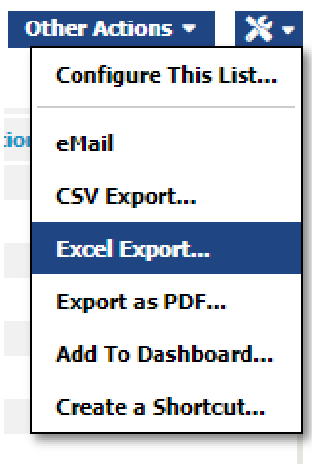
Select Delimited and click **Next >**.



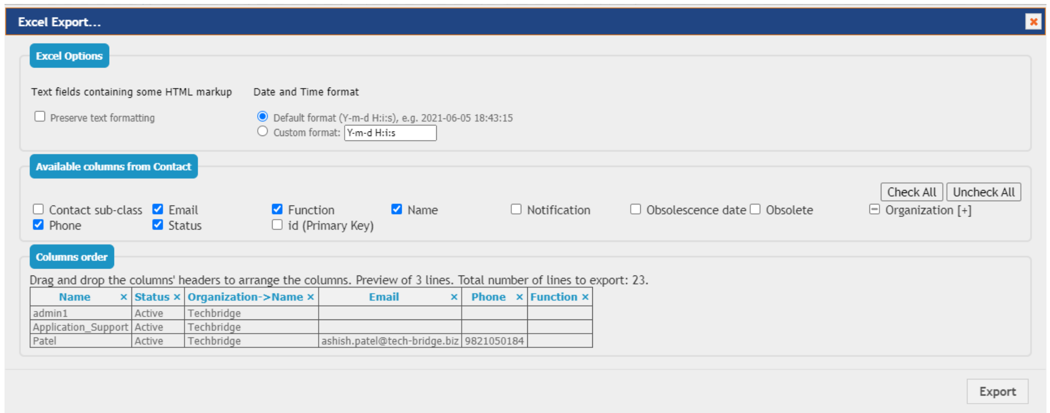
Select Comma and click **Finish**.

2.10.4. Excel Export

Starting with tbITSM v.1.0, a direct export to Excel (XLSX format) is available from the menu:

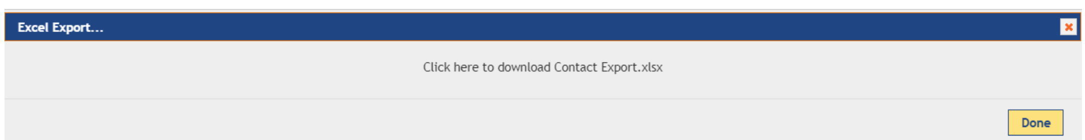


The following dialog box is displayed:



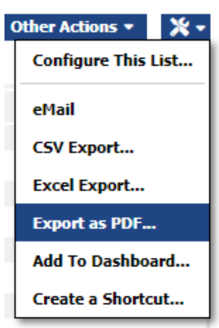
The Excel export action displays a configuration dialog which allows to:

- Pick the columns to be included in the export
- Arrange the order of the columns
- When you click on **Export**, the Excel file is built. If the list of objects to export is large (>1000), a progress bar is displayed. Once the export is complete, you can download the Excel file by clicking on the link ***Click here to download*** at the top of the dialog box:



2.10.5. PDF Export

Available in tbITSM v.1.0, a new menu item Export as PDF is available:

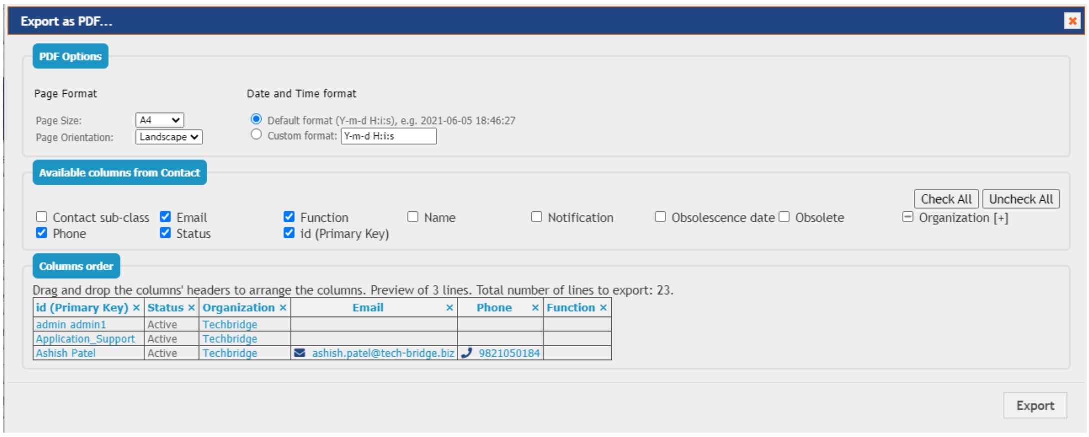


This menu generates a PDF document with one big table containing one row per object in the list and one column for each selected field.

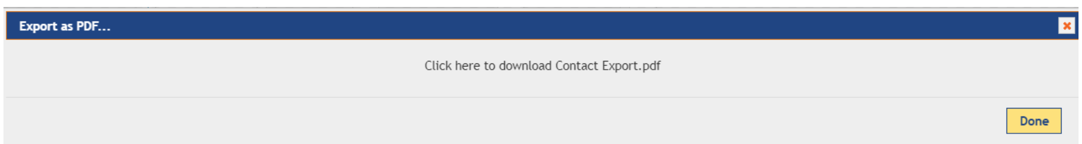
The PDF export action displays a configuration dialog which allows to:

- Select the page size for rendering the PDF: either A3, A4 or Letter
- Select the orientation of the page: Portrait or Landscape
- Pick the columns to be included in the export

- Arrange the order of the columns

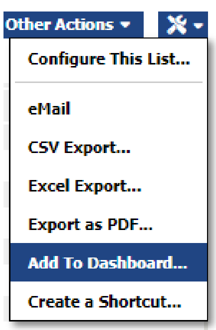


When you click on **Export**, the PDF document is created. If the list of objects to be exported is large (>1000), a progress bar is displayed. Once the export is complete, you can download the PDF file by clicking on the link ***Click here to download*** at the top of the dialog box:

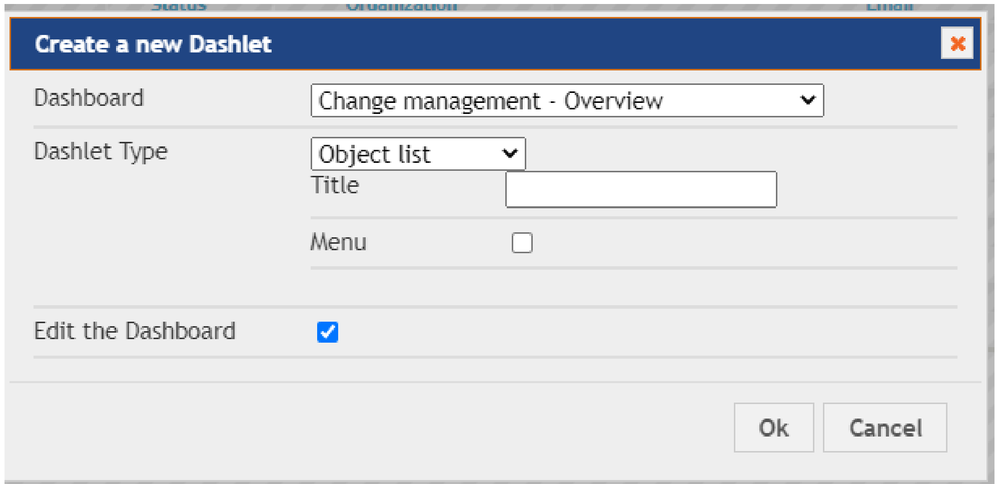


2.10.6. Add to Dashboard

Every result displayed as a list can be added to one of the editable dashboards. Click on the menu **Add to Dashboard**:



Then select the dashboard to which the list will be added and the type of "Dashlet" to use:

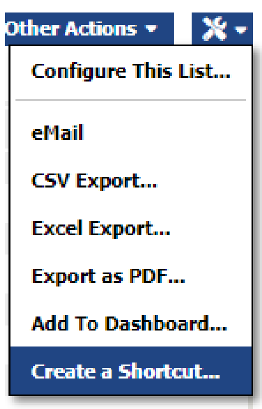


A list can be added to a dashboard either as:

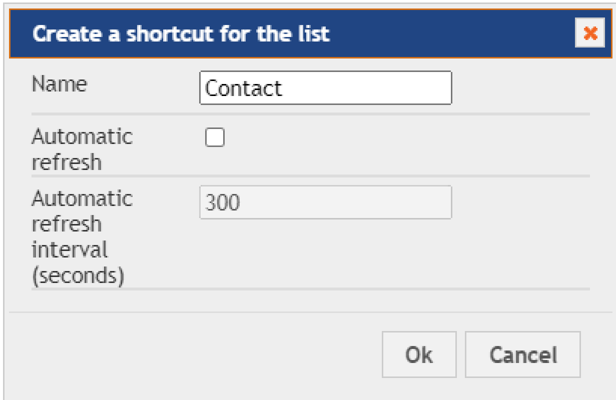
- A plain list of objects
- A pie chart (By grouping the objects on a given field)
- A bar chart (By grouping the objects on a given field)
- A table with the count of objects, grouped on a given field

When the user presses Ok, the new dashlet is added at the end of the selected dashboard (by adding a new cell after the last used cell). If Edit the dashboard is checked, then the modified dashboard is immediately opened in the dashboard editor.

2.10.7. Create a Shortcut



When prompted, enter the name for your new shortcut (this is the name that will appear under **My Shortcuts** in the menu), and click ***Ok***.

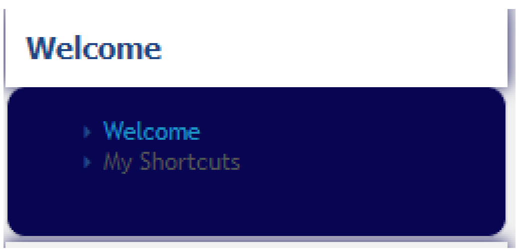


The new shortcut is then added to the menu.

2.11. Shortcuts

Shortcuts are used to navigate to a particular list of objects in tbITSM v.1.0. You can think of it as a kind of dynamic bookmark, since the **bookmarked** list is defined by its search criteria.

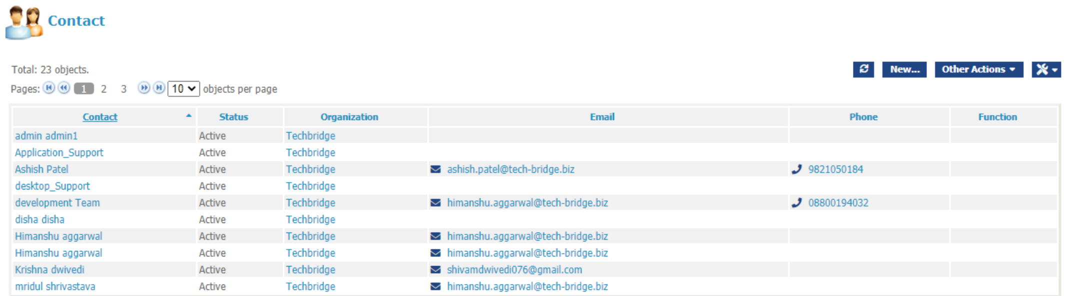
Shortcuts appear as extra menu items under the **My Shortcuts** menu in the **Welcome** drawer of the explorer menu on the left of tbITSM v.1.0. pages.



2.11.1. Creating a New Shortcut

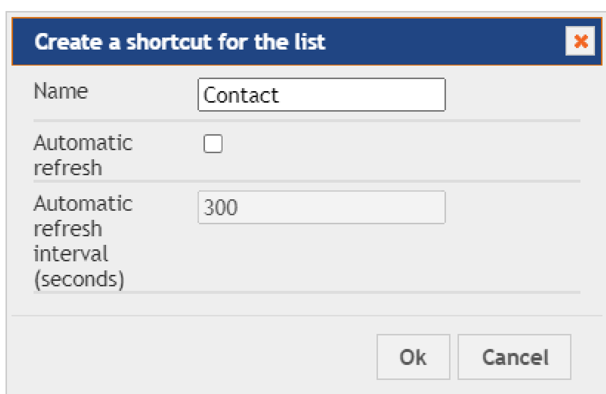
To create a new shortcut, start by searching the objects you want to **bookmark**.

For example, imagine that you want to have an easy access to all persons tagged as **On-Site Contact**. Search for contacts having **on-site in their** function**:**



Then click on the toolkit **popup menu and select** Create a Shortcut**:**

When prompted, enter the name for your new shortcut (this is the name that will appear under My Shortcuts in the menu), enable the automatic refresh if you wish (minimum allowed: 5 seconds), and click *Ok*.

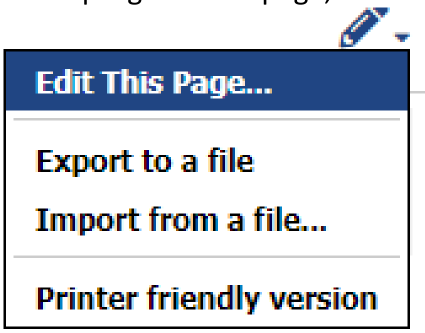


The new shortcut is then added to the menu.

2.12. Editing Dashboards

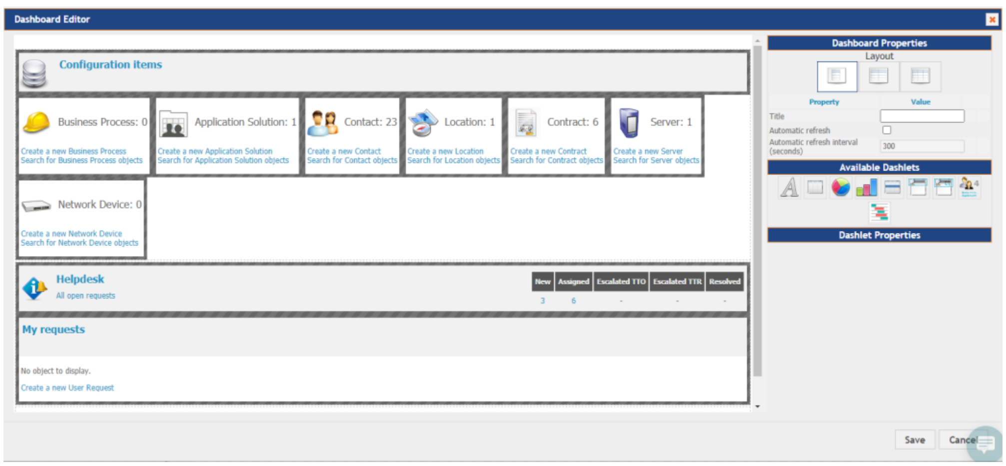
Some of the pages in tbITSM v.1.0. are dashboards. Dashboards are used to display information about different tbITSM v.1.0. objects on the same page. Most of the overview pages, as Well as the Welcome page are dashboards. In tbITSM v.1.0, each user can configure their own dashboards by editing the existing dashboard pages. It is not possible to create new dashboard pages.

When a page is an editable dashboard, the small Pen drop-down menu appears at the top-right of the page, next to the Log-off menu.



To edit the current page, select Edit this page.

After a few seconds a popup dialog appears containing the dashboard editor dialog:



A dashboard is made of building blocks named after dashlets.

A dashboard itself arranges the display of the dashlets it contains. A dashboard has only three properties:

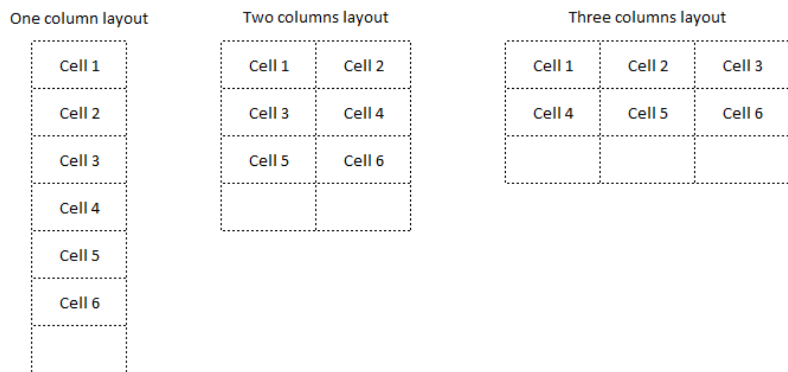
- A layout (one, two or three columns)
- An optional title
- The automatic refresh setting (disabled, or refresh interval given in seconds)

The layout determines how the dashlets are arranged on the page: it determines a grid, with each dashlet being assigned to one cell of the grid. When the layout is changed, the cells (as Well as the dashlets inside the cells) are re-organized to produce the desired display.

To add a new dashlet to the dashboard, drag one of the dashlet icons into the desired area of the dashboard, then adjust the properties on the right to finalize your dashboard.

When you are done with editing, click on ***Save*** to save your modifications and go back to the tbITSM v.1.0. page.

To undo all modifications performed since clicking on **Edit this page**, click on ***Cancel***.



To edit the properties of a dashlet, click on it in the left part of the dialog. An orange border appears around the selected dashlet, and the properties of the dashlets are displayed on the right part of the dialog. To modify the appearance of the dashlet, simply modify the properties on the right. The dashlet will change to reflect your modifications. To remove the selected dashlet from the dashboard, click on the **red cross** at the top-right of the selected dashlet.

To add a new dashlet to the dashboard, drag one of the dashlet icons into the desired area of the dashboard, then adjust the properties on the right to finalize your dashboard.

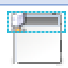
When you are done with editing, click on ***Save*** to save your modifications and go back to the tbITSM v.1.0. page.

To undo all modifications performed since clicking on **Edit this page**, click on ***Cancel***.

2.12.1. Dashlet Types

The following types of dashlets are available to compose your dashboard:

Icon	Name	Description
	Text	A free text area. Use this dashlet to add a title or a comment to a page
	Object List	A list of tbITSM v.1.0. objects, displayed as a table, like any search

		result
	**Pie Chart**	A list of tbITSM v.1.0. objects, grouped on the given field and displayed as a pie chart. Instead of the count, various functions can be used (sum, average, min and max) on specified fields. The ordering direction can also be specified and the max number of results.
	**Bar Chart**	A list of tbITSM v.1.0. objects, grouped on the given field and displayed as a bar chart. Instead of the count, various functions can be used (sum, average, min and max) on specified fields. The ordering direction can also be specified and the max number of results.
	**Group By**	A list of tbITSM v.1.0. objects, grouped on the given field and displayed as a table. Instead of the count, various functions can be used (sum, average, min and max) on specified fields. The ordering direction can also be specified and the max number of results.
	**Header**	A grey banner with an icon on the left and a title.
	**Header with counts**	A grey banner with an icon on the left, title and a table showing the count of objects grouped by a given field.
	**Badge**	An icon with a text showing the total number of objects of the given class and two links: 'create a new…' and 'search for…'

2.12.2. Dynamic Dashlet

If the Dashboard is a *Menu* dashboard, then you can use the 2 first methods

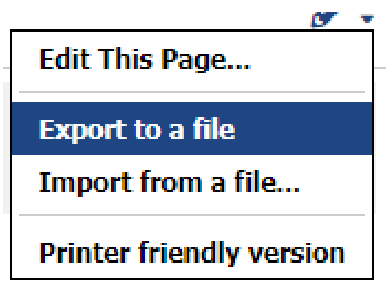
- **Based on Current user**
  : current_contact_id: This is the id of the contact linked to the connected user, it's 0 if no contact is associated to the user. It allows to display the Tickets for which the current user is the agent for e.g.
  SELECT Ticket WHERE agent_id =: current_contact_id

- **Based on Time**
  In the below OQL example, we display Tickets created in the past 30 days and not yet closed
  SELECT Ticket
  WHERE creation_date > DATE_FORMAT (DATE_SUB (NOW (), INTERVAL 30 DAY),'%Y-%m-%d 00:00:00')
  AND STATUS! = 'closed'

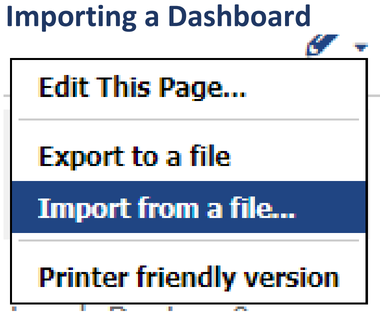
- **Based on Current Object**
  Only available for Dashboard Attribute defined on an Object, you can filter your OQL using any field of the current Object.
  For example, on Organization, you can use placeholders like: this->*att_code* where *att_code* can be any field of the Organization, including the id.
  SELECT UserRequest WHERE org_id=: this->id

2.12.3. Exporting a Dashboard

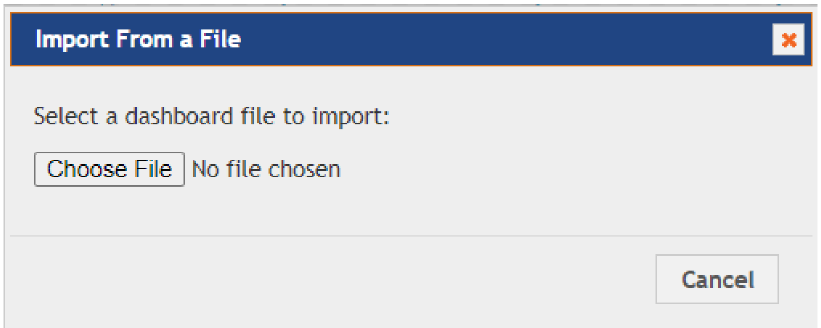
A dashboard definition can be exported as an XML file, for archiving or sharing it with other users. To export a dashboard, click on the menu item ***Export to file*** from the **Pen** popup menu. When prompted, enter the download location (Where the exported dashboard will be saved to on your computer).



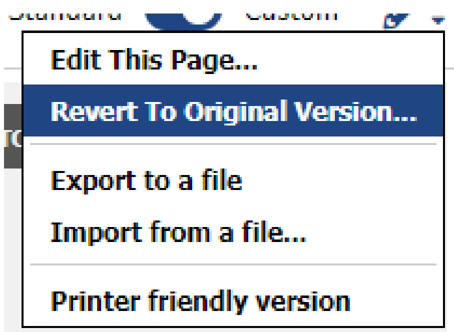
2.12.4. Importing a Dashboard



To import a dashboard definition from an XML file, click on ***Import from file*** and select the XML file to upload from your computer.



2.12.5. Discarding the customizations



At any time, the customization of a dashboard can be discarded in order to revert to the original definitions (shared by all users). Simply click on *Revert to Original Version* and acknowledge the confirmation popup dialog.

2.13. Bulk Modifications

There are two ways to perform bulk modifications in tbITSM v.1.0:

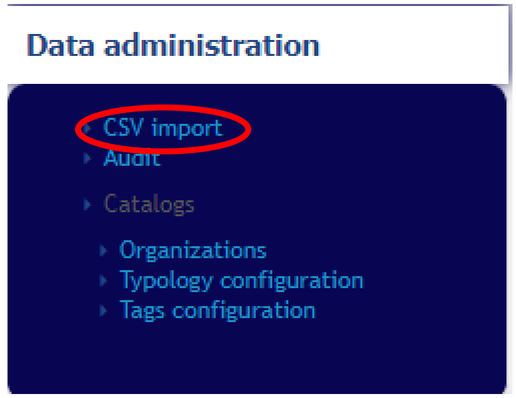
- The Interactive CSV Import: this is useful for re-importing a set of exported and modified data. Fix Me! missing date format capabilities.

- The Modify action on a list of objects: this is useful for data clean-up (normalization) when you want to align values within a set of objects, or for daily operations when you need to modify many objects in the same manner.

2.13.1. Interactive CSV Import

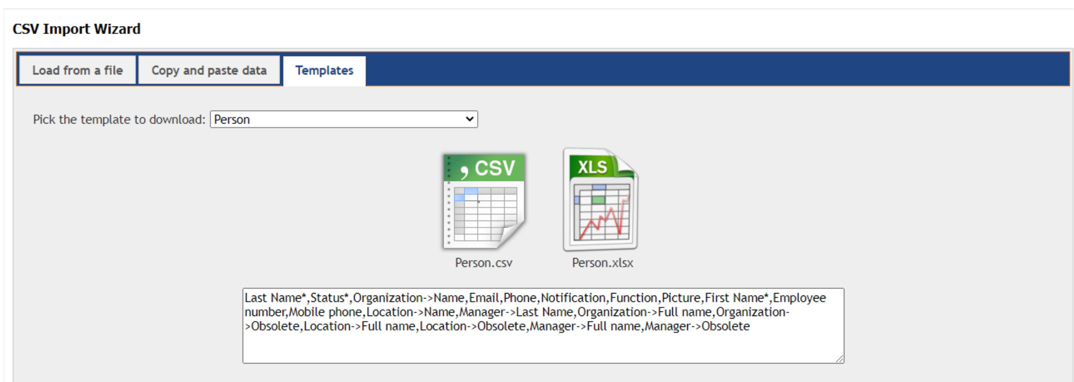
tbITSM v.1.0 provides a powerful CSV Import feature to assist end-users (and administrators) in massively creating or updating objects in tbITSM v.1.0.

To launch the CSV Import Wizard, click on the **CSV Import** menu in the Data Administration section.



2.13.2. CSV Templates

tbITSM v.1.0. provides CSV templates for creating/importing any class of objects. You can either download or copy/paste the template from the third tab **Templates** in the CSV Import wizard.



Select the desired class from the drop-down list to generate the template. Then either click on the ***big Excel icon*** to download the template, or copy/paste the text from the text area below the icon.

2.13.3. Expected format for Values

In order to load some type of attributes, you need to provide their value in a particular format.

- **Date**
  Date format of the imported data can be specified during the import, but all dates in your source file must be using the same format. This format can be either the default one defined on your tbITSM v.1.0. or any that you specify.

- **Text with HTML**
  You can provide the raw HTML with tags.
  Example: <p>Bold
Yes</p>

- **Image/File**
  Some classes contain an image or a file document as one of their fields. For example, the Person class, contains a Picture field which holds the picture of this person. When editing a Person interactively, the user can upload an image from her computer to provide the picture of the person. However, when performing a CSV import, it is not possible to “upload” such a file or to put its content inside the CSV file itself, since the CSV format does not support this.

  In such a case the CSV file must provide - in the appropriate CSV column - an URL to upload the file from. This URL must be accessible from the tbITSM v.1.0. server and return the expected image (or file document). If the upload fails or if the format of the uploaded document does not match the expected format for the field (for example if the URL does not return an image for the *picture* field), the import/update will fail for this field (and the whole line of the CSV will be rejected).

Limitations:

- The URL must point directly to the image to upload (redirections are not supported).
- The URL must be accessible from the tbITSM v.1.0. web server (beware some public websites may deny access to a PHP script - they check the HTTP User Agent string).
- The URL must not require authentication, since the upload is done by the tbITSM v.1.0. server itself.
- URLs pointing to tbITSM v.1.0. itself (like the URLs provided by CSV/Excel export) are supported. But only if they point to the same tbITSM v.1.0. instance (in such a case the import will parse the provided URL and will read the document from the tbITSM v.1.0. database).
- If you use an administrator account to perform the CSV import, you can specify a path to a file located on the tbITSM v.1.0. server itself (but this works only for administrators).

2.13.4. External Object

For loading an attribute which is an external key to another object, you can provide any combination of attributes which are declared as *reconciliation attributes* for the remote class.

Example: if you were to load UserRequest and set the Caller (caller_id), you could provide any combination of those columns to identify the caller:

- Caller->Email
- Caller->Organization name
- Caller->First Name
- Caller->Last Name
- Caller->Full Name
- Caller->Employee number

TagSet

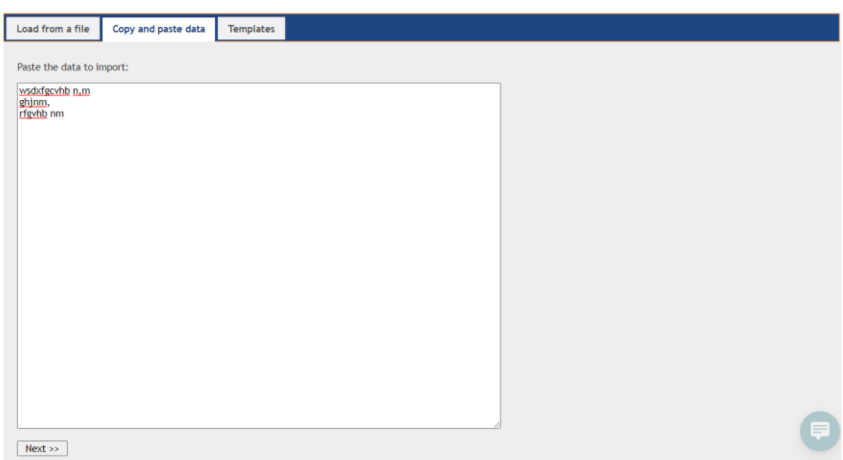
For loading a TagSet attribute, the tags have to be imported with their label separated by a pipe |. There must be a pipe at the beginning and at the end.

2.13.5. Uploading Data

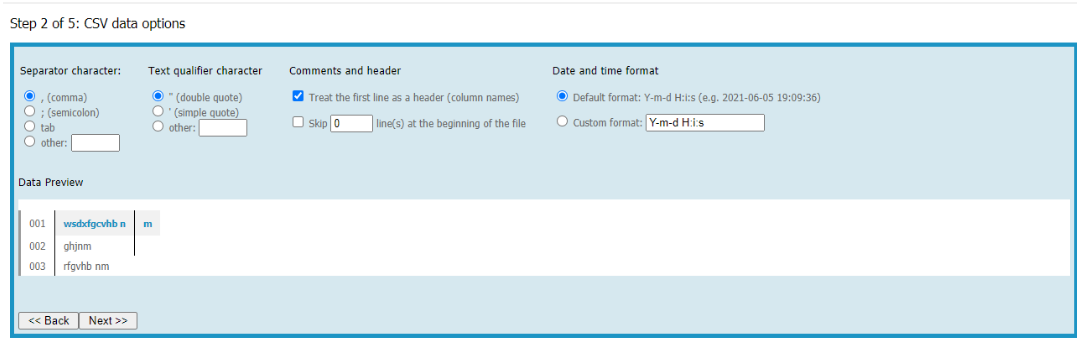
The first step of the CSV import consists of uploading the data to the server. You can either upload a file from your computer or directly copy/paste the data in the second tab of the wizard.

To upload the file from your computer, use the file upload form on the first tab of the wizard. Make sure that you also select the appropriate character set that was used for saving the file.

If you choose to copy/paste the data, use the second tab **Copy and paste data** of the wizard:



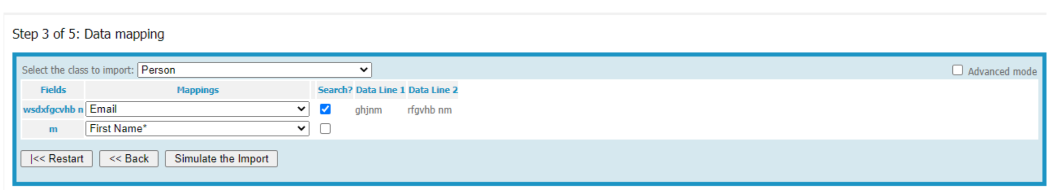
Then click ***Next »*** to proceed with the next step of the wizard.



The following parameters must be adjusted during this step:

Parameter	Meaning
Separator Character	This is the character used for separating the different columns inside the CSV file. If the columns are properly displayed and separated in the preview at the bottom of the page, then the current choice is correct. (The typical choice for a CSV file is the comma)
Text Qualifier Character	Sometimes the values inside the cells contain the separator character. Such fields are then enclosed by the **text qualifier** character to prevent the separator character from being wrongly interpreted in the middle of a cell. (A typical value is either a simple or double quote)
Treat the first line as a header	Check this box if the first line of the file contains a header (column names) instead of actual data. Note that on the preview at the bottom of the page, the header line is displayed with a different color.
Skip x lines at the beginning	Some CSV files (typically CSV files generated by hand-made scripts) contain some comments at the beginning of the file. Use this setting to tell tbITSM v.1.0. to ignore as many lines as needed to skip the comments. The skipped lines are removed from the preview at the bottom of the page.

Once the data looks correct in the preview, click on the **Next »** button to move to the next step:



The third step of the Import Wizard defines the mapping between the column of the CSV file and the fields of the object in tbITSM v.1.0. The first operation consists of selecting the class of objects to import. This determines the list of fields available for the mapping.

Fields mapping

If the CSV file has been originally created using a template or an tbITSM v.1.0. CSV export, the header provided in the CSV should automatically give you the correct mapping.

If this is not the case, you have to create this mapping manually.

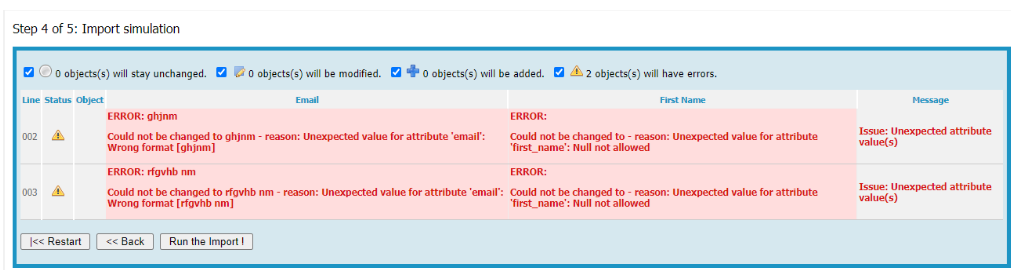
Search Criteria

The CSV Import Wizard can be used to either create or update objects in tbITSM v.1.0. The wizard uses the *search criteria* defined at this step to determine whether each line of the CSV file corresponds to an existing object in tbITSM v.1.0. or not. If the object exists, then it will be updated, otherwise it will be created. Typically, in tbITSM v.1.0. search forms, all selected *search criteria* are combined using the AND operator. (i.e., an object must match each value of the Search criteria to be considered as a match).

Once the mapping and the search criteria looks correct, proceed to the simulation of the import by clicking on the **Simulate Import**: button.

Simulate Import

After a few seconds, the following summary screen is displayed:


Step 4 of 5: Import simulation

0 objects(s) will stay unchanged. 0 objects(s) will be modified. 0 objects(s) will be added. 2 objects(s) will have errors.

Line	Status	Object	Email	First Name	Message
002			ERROR: ghjnm Could not be changed to ghjnm - reason: Unexpected value for attribute 'email': Wrong format [ghjnm]	ERROR: Could not be changed to - reason: Unexpected value for attribute 'first_name': Null not allowed	Issue: Unexpected attribute value(s)
003			ERROR: rfgvhb nm Could not be changed to rfgvhb nm - reason: Unexpected value for attribute 'email': Wrong format [rfgvhb nm]	ERROR: Could not be changed to - reason: Unexpected value for attribute 'first_name': Null not allowed	Issue: Unexpected attribute value(s)

|<< Restart << Back Run the Import !

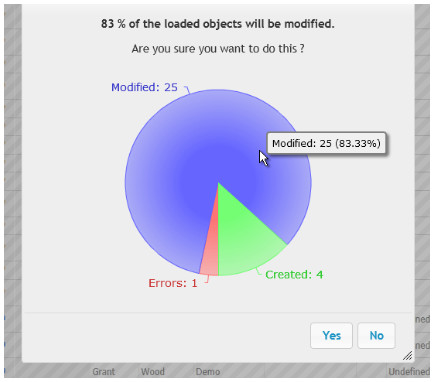
Note the number of objects to be created and the number of objects to be updated, as Well as the number of errors. You can use the checkboxes at the top of the page to filter the display for a quicker analysis of the results.

The lines in errors are highlighted with a red background. Errors will not prevent the import from running, but the corresponding lines will be rejected.

If the results from the simulation look weird, maybe the Search criteria aren't correct. then press **Back** button to change them and Simulate again. If you need to modify the data, click on ***|<< Restart*** (the **Copy and paste data** tab will contain a copy of the uploaded data). If the simulation looks correct, click ***Run the Import!*** to perform the actual import.

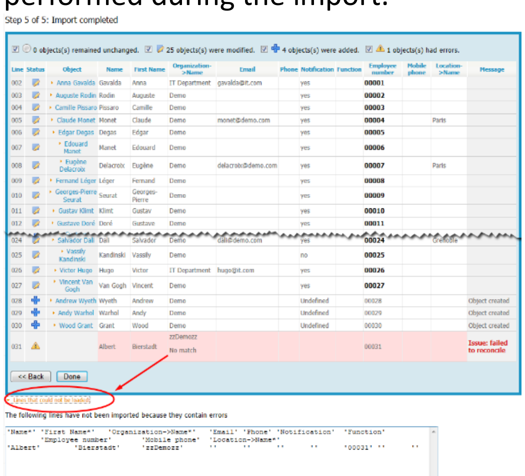
Run Import

If the number of objects imported and the percentage of objects to be created or updated exceeds a certain level, a confirmation dialog is displayed:



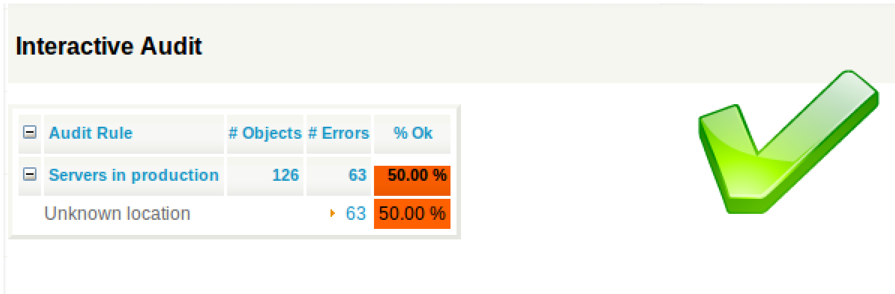
Click ***Yes*** to close the dialog and proceed with the data import.

The last screen of the wizard displays the summary of the operations performed during the import.



2.14. Understanding Audits

Audit is an tbITSM v.1.0. feature designed to check the consistency of information stored in the tbITSM v.1.0. database. The audit is used to answer questions such as: “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?”



The verifications are named **audit rules**.

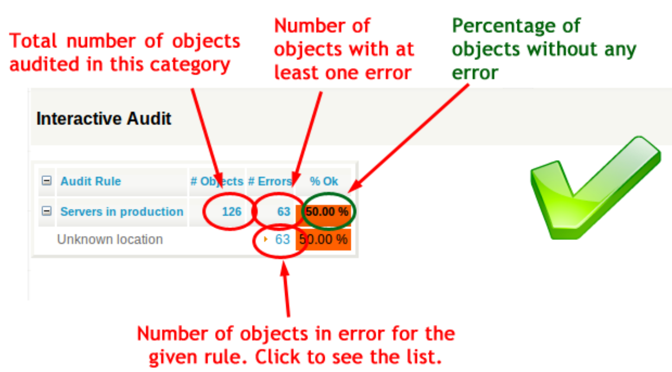
Audit rules are grouped into **audit categories**. A category defines the list of objects (the scope) that must be compliant with the rules. This scope is computed dynamically. In other words, the list of objects to check is evaluated when the audit page is being displayed.

An **audit rule** defines the actual constraint that needs to be checked for each object in the scope. For instance, “Device must be attached to at least one active support contract.”.

To add or modify an **audit category** or an **audit rule**, ask your tbITSM v.1.0. administrators. Only administrators are allowed to manage audit categories and rules.

2.14.1. 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 rules in this category)
- The percentage of objects without any error in this category
- Then 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

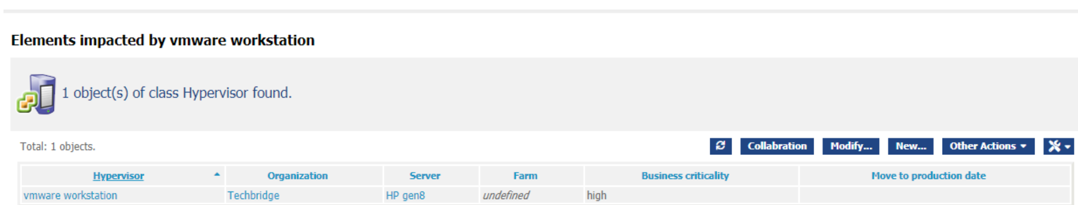
2.14.2. Obsolete Object

Data Model now includes an obsolescence notion defined per class of object, to improve user experience by highlighting / hiding obsolete objects.

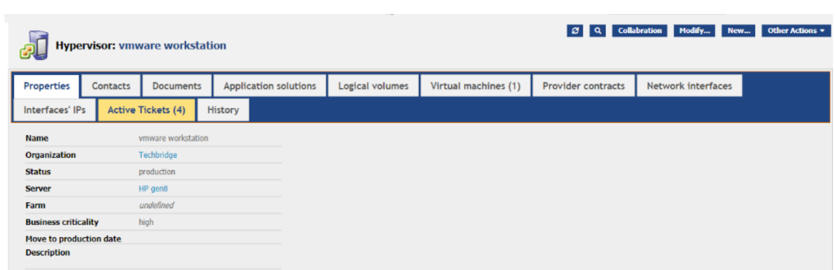
2.14.3. Highlight Obsolete Objects

Object defined as obsolete are flagged in the console to make this aspect obvious, by graphical icon and tag.

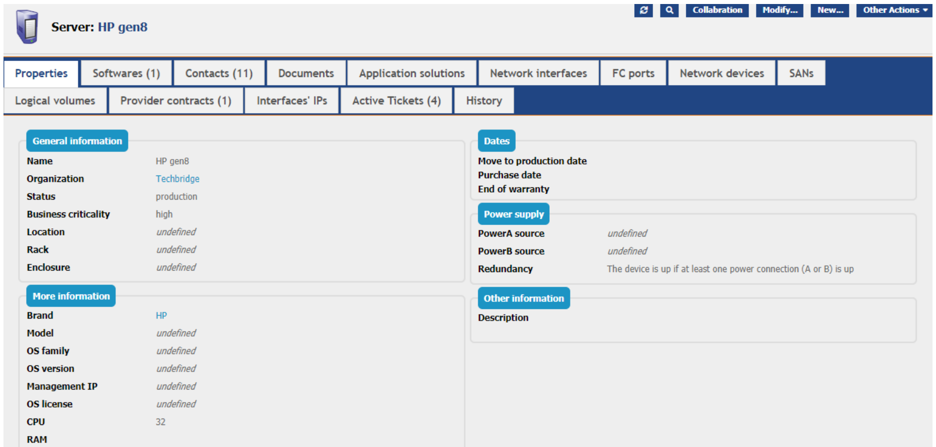
On a list:



On an object:



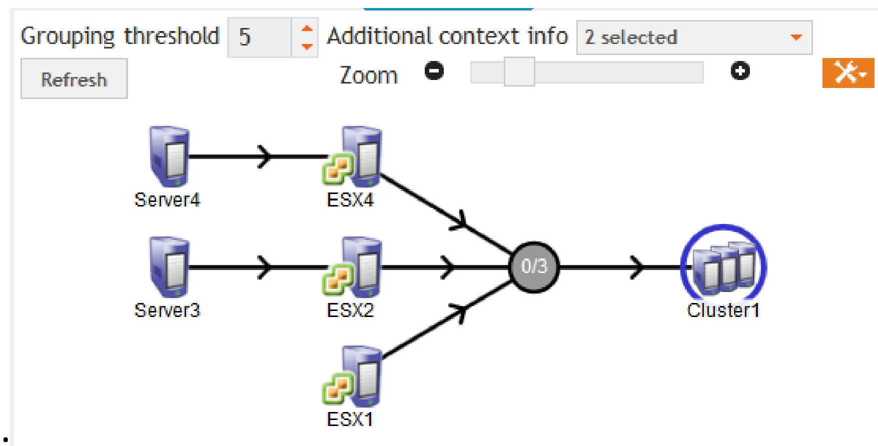
On a list of linked objects:



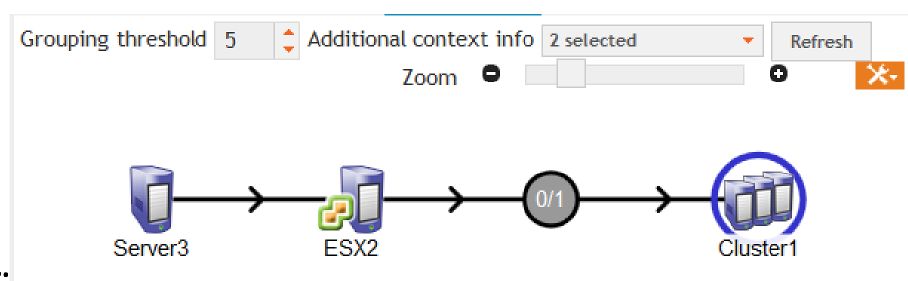
2.14.4. Improved Impact Analysis

Obsolete objects are ignored by Impact Analysis

Before:



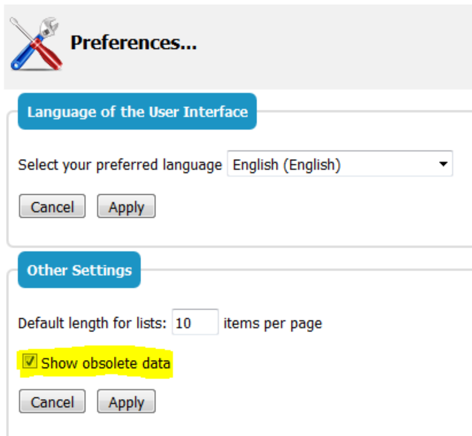
After:



2.14.5. Hide Obsolete Objects

Based on user preference, objects which are defined as obsolete, are removed from **displayed lists** in the tbITSM v.1.0. console.

An **obsolete object details** is displayed regardless of user preference.



The same Farm as above displayed for a user, requesting to not display obsolete objects.

Modify Mode

- In details screen, on a drop-down list: obsolete objects will not be proposed.
- In a relation tab, obsolete objects will not be displayed, so cannot be removed.
- In a relation tab, obsolete objects cannot be added.

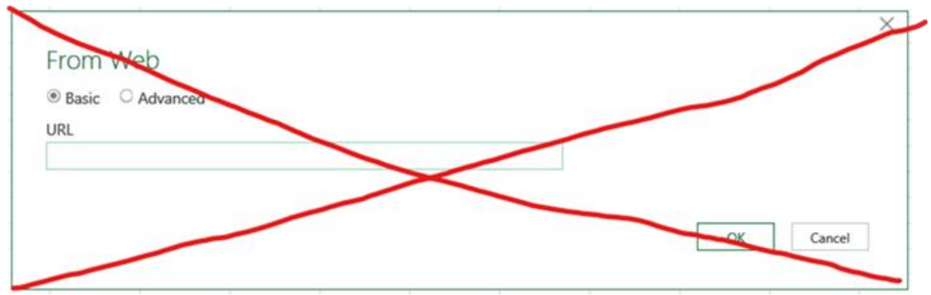
Other Display Difference

- Shortcut display a list without obsolete objects
- Dashlets do not include obsolete objects
- Global Search do not return obsolete objects
- Bulk modify will not modify obsolete objects

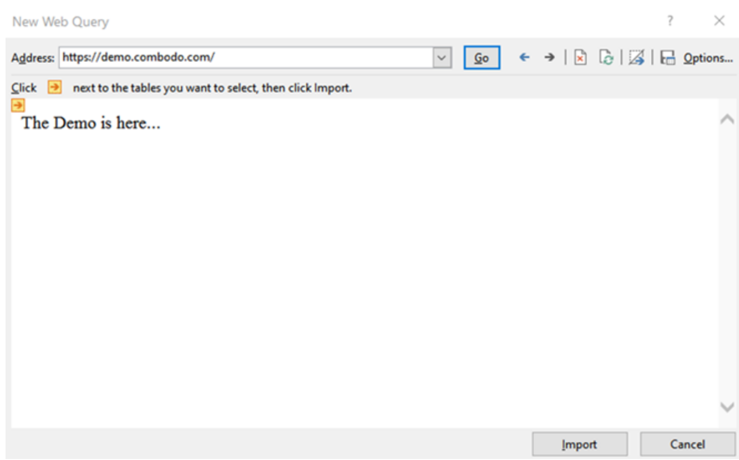
2.15. Reports in Excel

2.15.1. Make sure that Excel is properly configured

Click on the menu item “Data” / “From Web”. If you get a dialog similar to the one below (PowerQuery) this is not good.

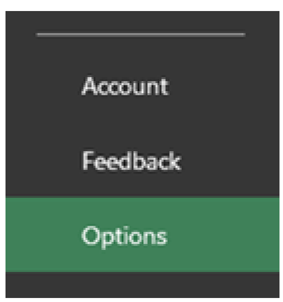


Instead, you should get a dialog similar to this:



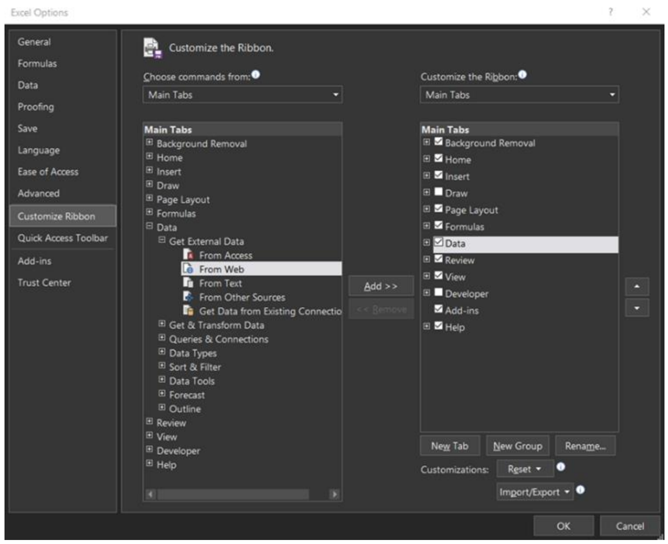
To use the old GetExternalDataFromWeb wizard, follow these steps:

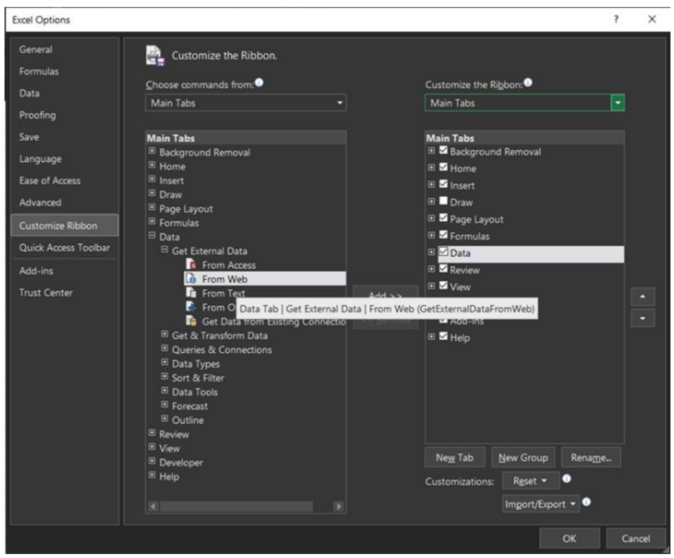
Go to the File / Options menu



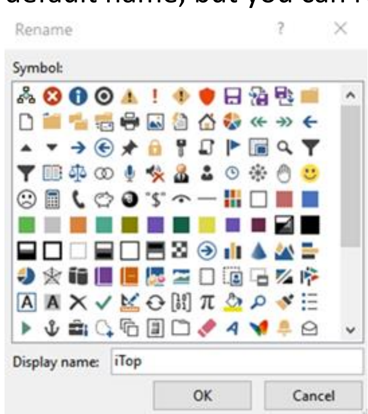
Click on the “Customize Ribbon” menu on the left:

In the drop-down list at the top-left, select “Main tabs”, in the tree displayed, open the item “Data” and select the menu item “From Web”

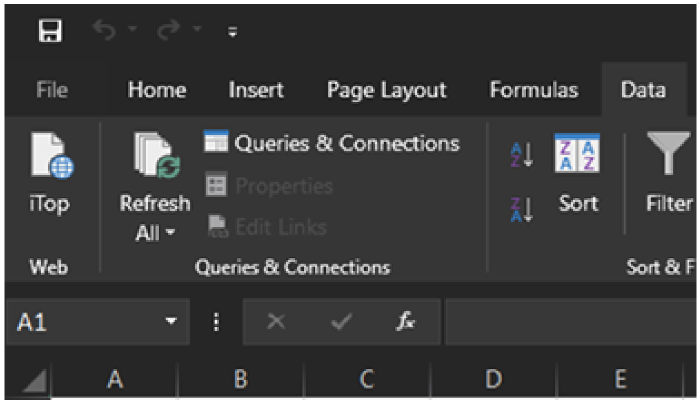




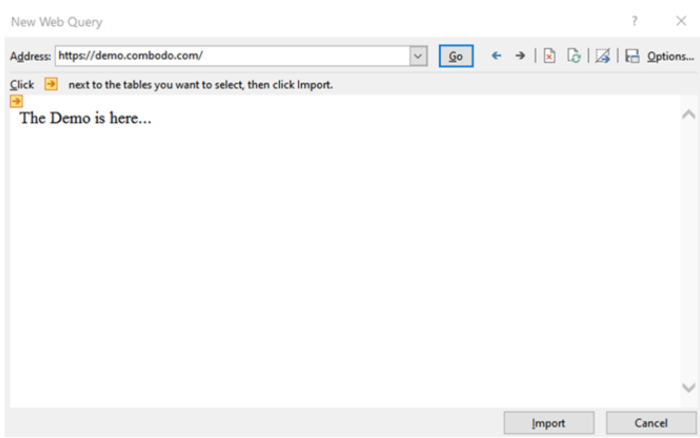
To add the command to your Ribbon you must create a new "Group" (using the button at the bottom of the dialog box). The group is named with a default name, but you can rename it if you like:



Similarly, you can rename the menu and choose an icon for it. Once your menu has been added to the ribbon, you should see something similar to this:

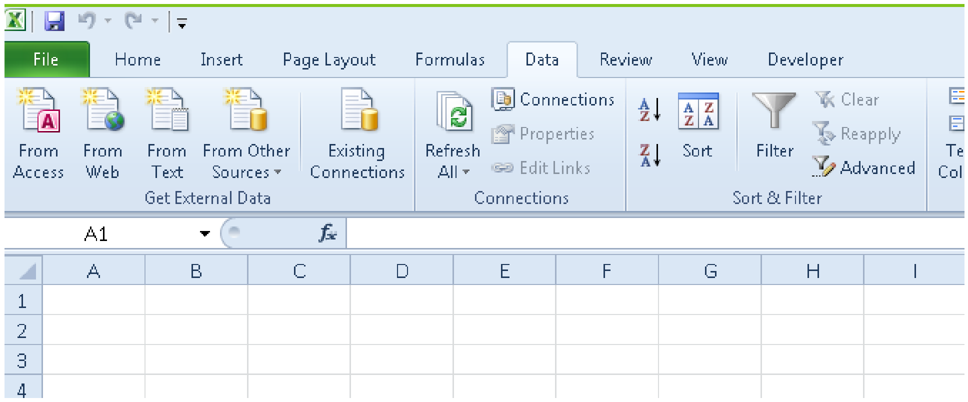


When you click on the custom menu, the “External Data From Web” wizard should appear:

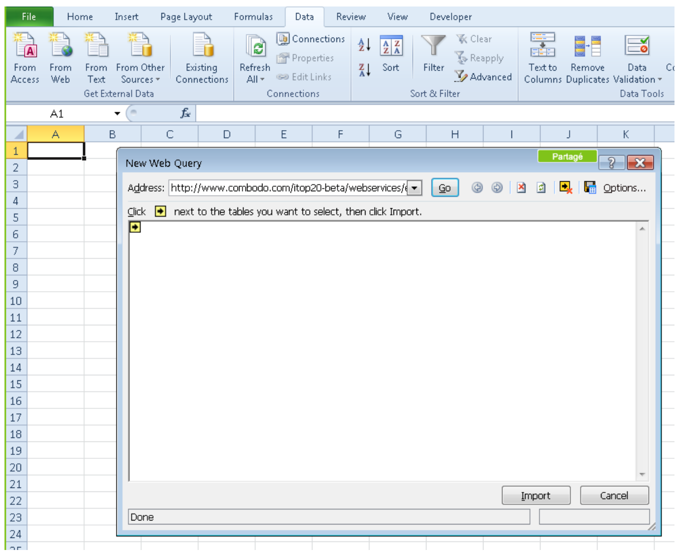


2.15.2. Importing data from tbITSM v.1.0.

Administrators can store predefined queries in tbITSM v.1.0. (menu “Administration / Query Phrasebook”). These queries can then be used to create reports in Excel. A query is identified by a number. It is called in Excel using a URL provided by your tbITSM v.1.0. administrator. For example, the URL below stored in the query phrasebook, returns a list of User Request. To use this query in Excel, click on the button ***From Web*** in the Excel tab **Data**

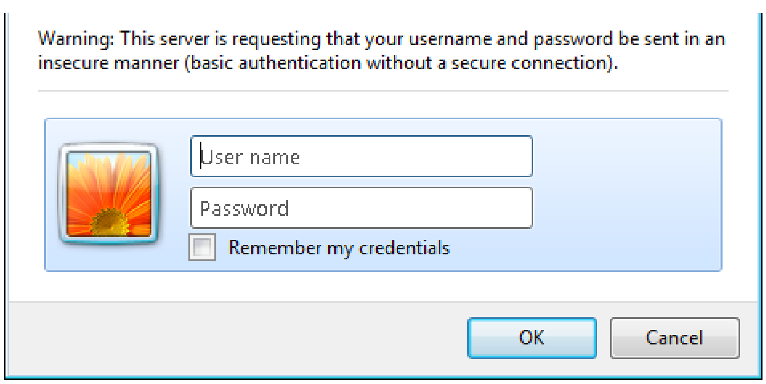


A window is displayed to enter the URL to be called:

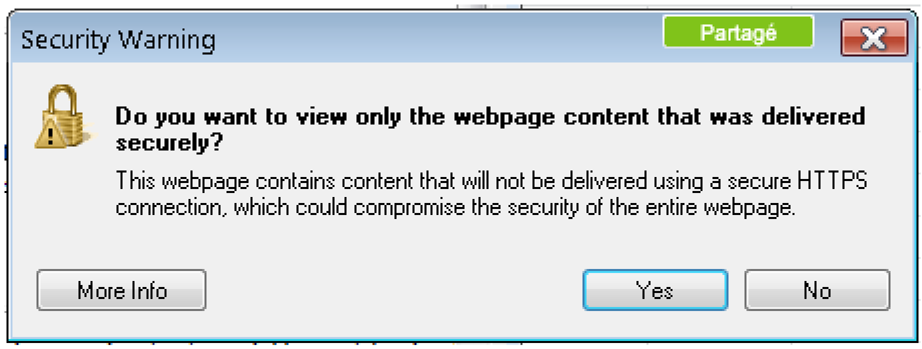


Enter the URL provided by the tbITSM v.1.0. administrator in the **Address** part and click on ***Go***.

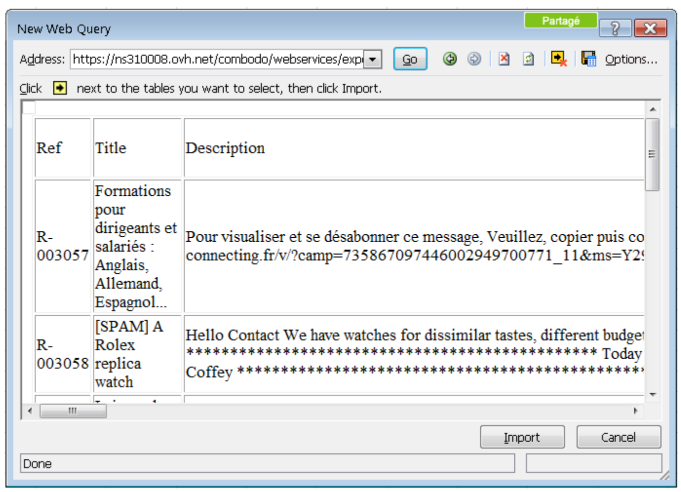
The tbITSM v.1.0. authentication window is displayed. Enter your user account and your password:



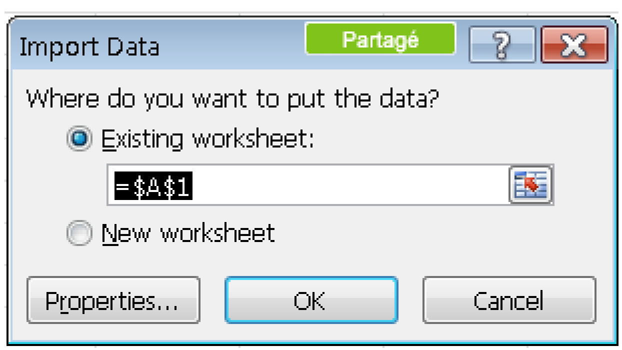
Click on ***Ok*:** if you are prompt by security popup windows.



The **New Web Query** is updated with the list of columns to be imported:

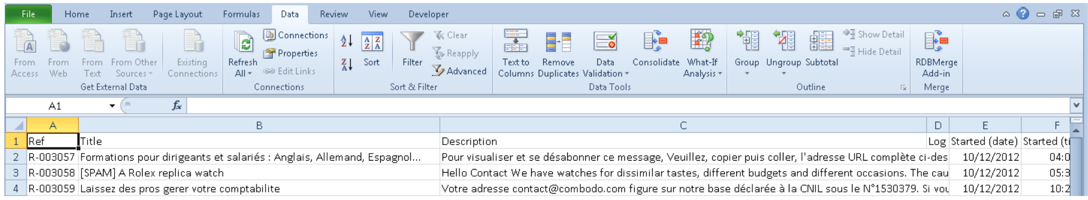


Click the ***Import*** button. Excel asks you where you want to insert the data:



If your query contains parameters, Excel will open several successive windows to enter them. Be careful, date parameters only support the format *yyyy-mm-dd*.

Click the ***OK*** button and Excel displays the result of the query. The language used for the display of columns corresponds to the default language of your login in tbITSM v.1.0.

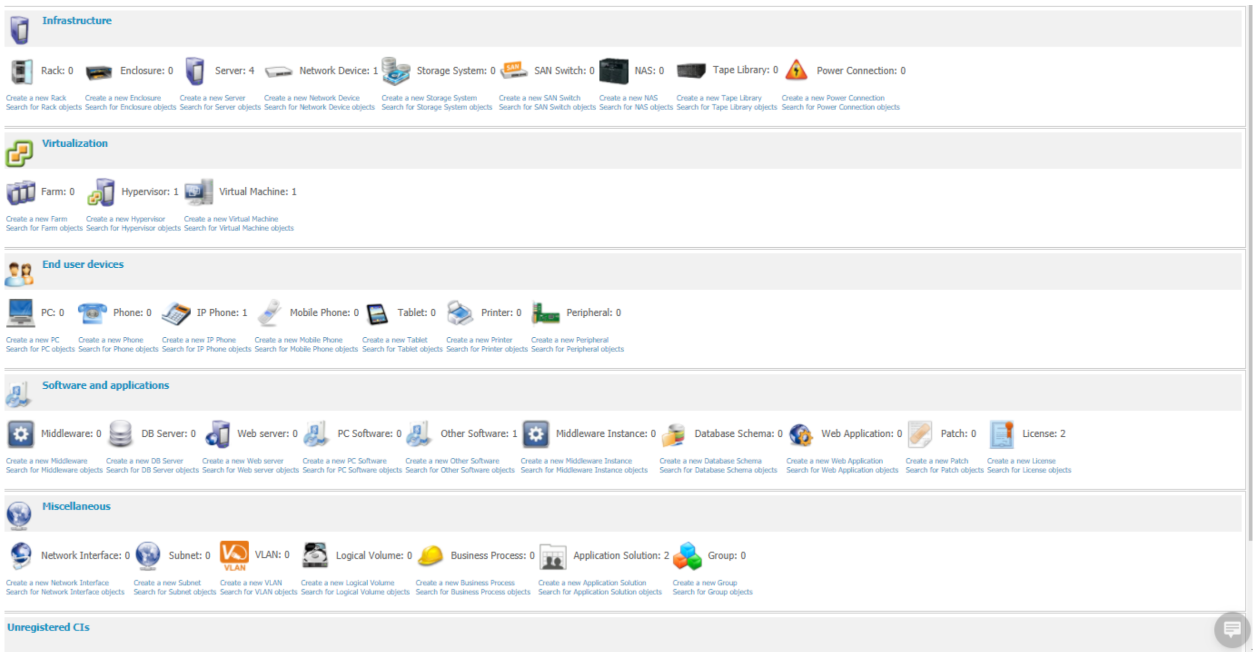


To refresh the data, click on the ***Refresh All*** button in the **Data** tab of Excel. You can use as many tbITSM v.1.0. queries as you want and use the results to create graphs or charts dynamically using standard Excel features.

3. Service Configuration Management

The Service Configuration Management module is mandatory module of tbITSM v.1.0. It contains the definition of all the basic building blocks of the CMDB: Organizations, Contacts (Persons and Teams) as Well as the usual physical devices (Servers, Network Devices…), software elements (DB Server, Database Schemas, Licenses, Patches…) and the relations between all these objects.

Dashboard:



3.1. Configuration Management Database

With the tbITSM Configuration Management Database (CMDB) application, build logical representations of assets, services, and the relationships between them that comprise the infrastructure of your organization. Details about these components are stored in the CMDB which you can use to monitor the infrastructure, helping ensure integrity, stability, and continuous service operation.

Use core features such as CMDB Health, CMDB Identification and Reconciliation, and CMDB CI Lifecycle Management to monitor and detect health issues, reconcile data integrity issues, and manage data life cycle.

3.1.1. Configuration Management and the CMDB

The Configuration Management data base (CMDB) creates and maintains the logical configurations your network infrastructure needs to support a tbITSM service.

These logical service configurations are mapped to the physical layout data of the supporting network and application infrastructure in each of your respective domains. They track the physical and logical state of IT service elements and associate incidents to the state of service elements, which helps in analyzing trends and reducing problems and incidents.

The configurations are stored in a configuration management database (tbITSM CMDB) which consists of entities, called Configuration Items (CI), that are part of your environment. A CI may be:

- A physical entity, such as a computer or router
- A logical entity, such as an instance of a database
- Conceptual, such as a Requisition Service

In each case, there are attributes about the CI that you want to maintain, and there is control you want to have over the CI. There are changes that may need to be made and tracked against the CI. Also, a CI does not exist on its own. CIs have dependencies and relationship with other CIs. For example, the loss of disk drives may take a database instance down, which affects the requisition service that the HR department uses to order equipment for new employees.

It is this relationship data that makes the CMDB a powerful decision support tool. Understanding the dependencies and other relationships among your CIs can tell you, for example, exactly who and what is affected by the loss of that bank of disk drives. When you find out that a router has failed, you will be able to assess the effect of that outage. When you decide to upgrade the processor in a server, you can tell who or what will be affected during the outage.

Configuration items differ from environment to environment because each customer has unique needs. Details about the exact physical attributes of a computer may be needed by one customer, but may represent meaningless data to another. The tbITSM Platform provides a mechanism to easily define new classes of configuration items and new relationships that may exist between CIs. New classes can be defined that extend other classes. For example, a laptop class exists that extends the computer class. The computer class itself extends the base CI class. Customer class extensions are automatically part of the tbITSM environment and blend seamlessly into the integration points for other ITIL processes.

You can for example, set the **Used for** attribute to a value such as ‘development’, ‘test’, or ‘production’. These values indicate the environment that the CI is supporting, and serve as a way of tracking a CI through its lifecycle in a changing environment.

3.1.2. Domain Separation and Configuration Management Database (CMDB)

Domain separation is supported in the Configuration Management Database (CMDB). Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can control several aspects of this separation, including which users can see and access data.

Support level: Standard

- Includes **Basic** level support.
- Business logic: The service provider (SP) creates or modifies processes per customer. The use cases reflect proper use of the application by multiple SP customers in a single instance.
- The instance owner must configure the minimum viable product (MVP) business logic and data parameters per tenant as expected for the specific application.

3.1.3. ITIL and Configuration Management

The CMDB has relationships with IT service management processes in the following areas: ITIL incident management, ITIL problem management, ITIL change management, ITIL service catalog management, and financial management.

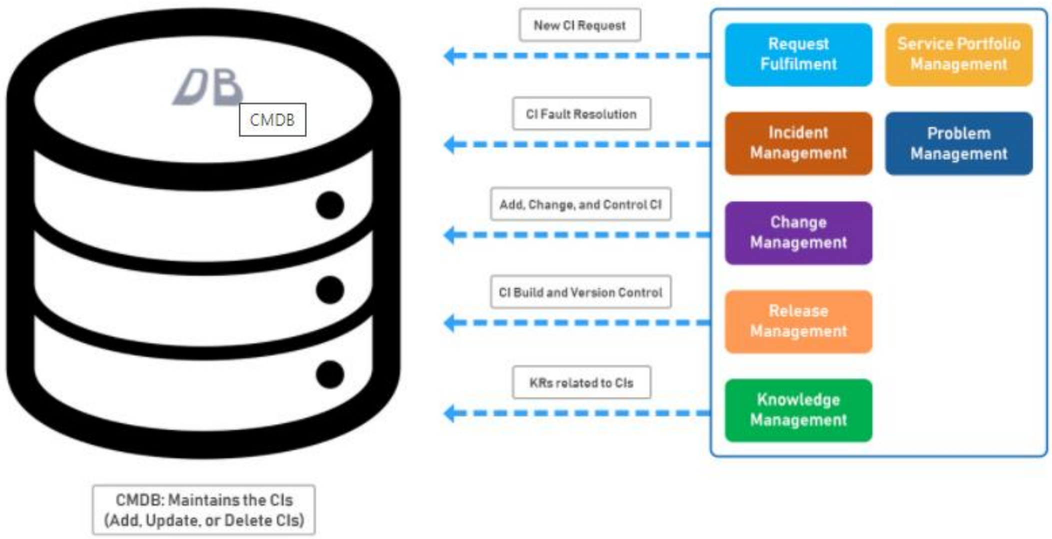
ITIL Incident Management: Configuration management assists Incident Management by providing the Service Desk with immediate information on the CIs affected, and more timely resolution of faults by understanding what CIs have been affected and changed.

ITIL Problem Management: Configuration Management assists Problem Management by linking the CIs affected by problems to the Incident / Problem / Change Management processes, and ensuring the CI status is properly maintained.

ITIL Change Management: Configuration Management assists Change Management by recording which CIs have been changed and controlling the status of CIs throughout the entire CI lifecycle. Configuration Management ensures that any changes made to CIs are recorded and kept accurate. Configuration Management is also used in the integration of Change Management with the Discovery feature.

ITIL Service Catalog Management: With Service Portfolio Management, business services in the CMDB can also be managed by the Service Catalog team, and exposed to end users who can then request items from them.

3.1.4. CMDB Process Diagram



3.1.5. Populate the CMDB

You can populate the CMDB by using Discovery, by using Import Sets to import and integrate data from a third-party source, by integrating with an external CMDB, or by manually creating CIs.

When you populate the CMDB with information, you create a record for each configuration item or on one of the tables that extend that table.

- **CMDB schema model:** A collection of class diagrams and class attributes for key CMDB classes.
- **CMDB tables descriptions:** Descriptions of key CMDB tables in the base system.
- **CMDB CI Class Models**: A tbITSM Store app that adds class models that extend the base CMDB class hierarchy. This includes class descriptions, identification rules, identifier entries, and dependent relationships if applicable. You can then use the added classes as any other CMDB base class.

Discovery patterns: A tbITSM Store app that provides a library of Discovery patterns for discovering specific devices and applications in the industry.

Service Graph connectors: tbITSM Store apps that provide pre-defined integrations for importing and integrating common third-party data into CMDB classes.

ITIL configuration management auto-discovery

The key to any configuration management business practice is the initial and on-going inventory or discovery of what you own. The tbITSM platform provides three options for auto-discovery:

- The separate and highly robust Discovery product.
- For organizations that want to leverage the discovery technologies. tbITSM v.1.0. integrations to those technologies via web services. Scanned data can be mapped directly into the CMDB.

Discovery

The Discovery product automatically populates the CMDB. Discovery uses probes, sensors, and patterns, to collect and process data about computers, servers, printers, a variety of IP-enabled devices, the applications that run on them, and the relationships between all the items found. This information is sent back to the instance and is used to populate the CMDB.

Import data from another source using Import Sets

You can import data to the CMDB using Import Sets. Import sets find files of information (in formats such as Excel, or CSV), import them, and transform them onto the required table. This process can be scheduled or performed on demand.

3.1.6. Baseline CMDB

CMDB baseline provides capabilities that help you understand and control the changes that have been made to your configuration items (CIs) in the CMDB.

You can create a baseline, which is a snapshot of your configuration items in the CMDB. You can review the changes that have been made to that configuration item since a previous baseline. Multiple baselines may be created and the system tracks the changes that have been made per baseline.

- Creating a baseline captures the attributes of the CI as well as all first-level relationships for the CI. Any changes to the base CI or to any related CIs are captured and displayed. Newly created CIs are not automatically added to a baseline.

- Associate a configuration item with a task, a change or change task, and to propose changes to the CI after the change is complete. You can record changes, and these changes are not applied to the CI immediately but are delayed until the change is complete.

- When the change is complete, you can choose to apply the proposed changes which makes all changes previously proposed and associates the changes with the task.

3.1.7. Managing proposed changes

The proposed changes feature allows you to pre-configure changes to configuration items and their associated relationships. These pre-configured changes are prepared to be implemented, but do not actually happen until they are applied at a later time. When you view a CI, the proposed changes can be displayed so that you can see what is planned.

This feature is useful when you want to make modifications while a change process is in the approval stage, and only implement the changes after the approvals are complete. If the change is never approved, no changes to records have to be reversed. If the change is approved, a quick command applies all the proposed changes.

You can make the following proposed changes to a CI:

- Modify any field on the CI form.
- Add or delete a relationship to that CI.

To modify a relationship, you must delete the current relationship and add a new relationship. You cannot delete a proposed change.

3.2. Contacts

3.2.1. Person

The class Person is used to describe physical persons as contacts in the CMDB. Persons can be grouped into Teams. Persons can be linked to other configuration items (for example to describe who to contact in case of problem with an application). In tbITSM v.1.0. the caller of a User request is a Person as Well as the agent assigned to resolve it.

Person Properties

Name	Type	Mandatory?
Last Name	Alphanumeric string	Yes
First Name	Alphanumeric string	Yes
Organization	Foreign key to a(n) Organization	Yes
Status	Possible values: Active, Inactive	Yes
Location	Foreign key to a(n) Location	No

Function	Alphanumeric string	No
Manager	Foreign key to a(n) Person	No
Employee number	Alphanumeric string	No
Email	Email address	No
Notification	Possible values: no, yes	No
Phone	Alphanumeric string	No
Mobile phone	Alphanumeric string	No

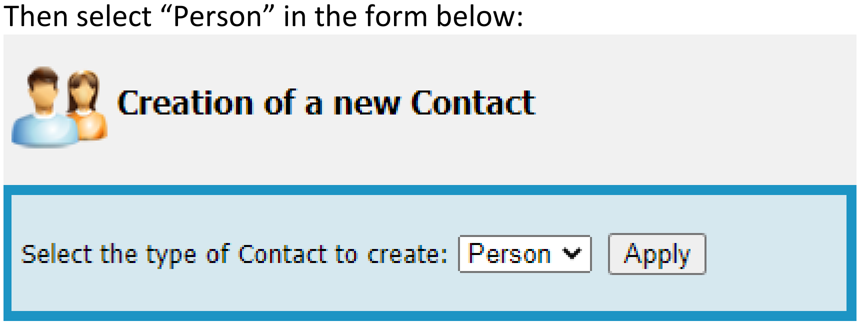
Tabs

Tab	Description
Teams	All the teams this person belongs to
Tickets	All the tickets this person is the caller
Cis	All the configuration items linked to this contact

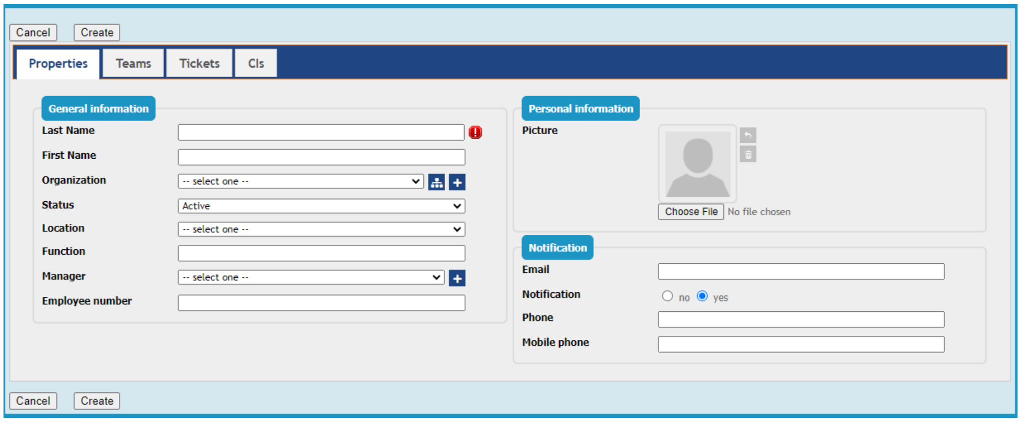
Creating a New Person

Click on the “New contact” menu:

Then select “Person” in the form below:



And click “Apply” to display the person creation form:



Team

A team represents a group of persons. In tbITSM v.1.0, Teams can be used to: associate a set of persons to a given configuration item, create a “workgroup” for assigning tickets or group persons for receiving notifications.

Team Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Status	Possible values: Active, Inactive	Yes
Organization	Foreign key to a(n) Organization	Yes
Email	Email address	No
Phone	Alphanumeric string	No
Notification	Possible values: no, yes	No
Function	Alphanumeric string	No

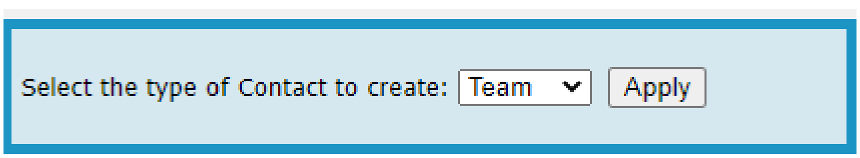
Tabs

Tab	Description
Members	All the people belonging to this team
Tickets	All the tickets assigned to this team
Cis	All the configuration items linked to this contact

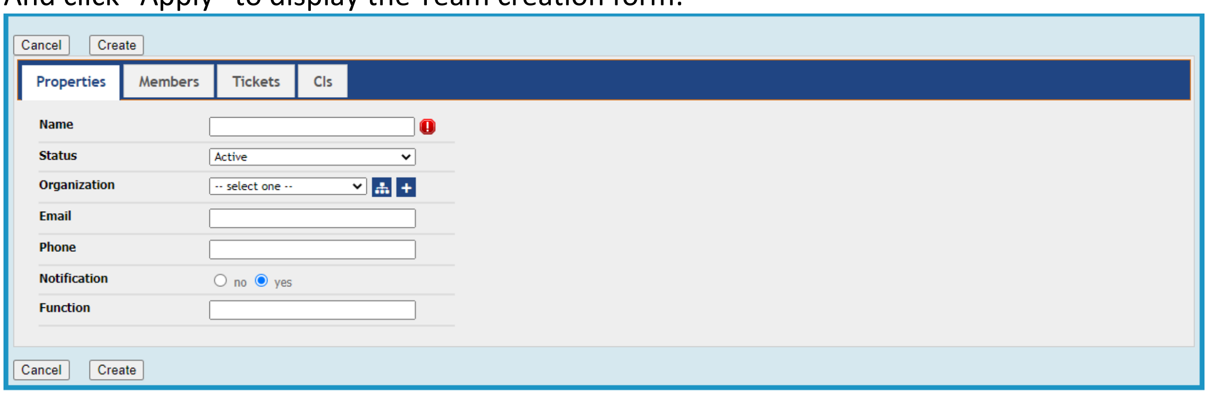
Creating a new Team

Click on the “New contact” menu:

Then select “Person” in the form below:



And click “Apply” to display the Team creation form:



To add members to the Team, click on the tab “Members” and use the buttons to Add/Remove Persons from the list.

You can also specify a “Role” for each member of the Team.

3.3. Documents

There are several types of documents in tbITSM v.1.0. All of them can be managed from the “Documents” menu in the “Configuration Management” section:

3.3.1. Document File

A File Document is used to upload documentation that can be shared across multiple objects in the CMDB. Once a document (in any format: Word, PDF, Spreadsheet, etc.) has been uploaded into tbITSM v.1.0, it can be "attached" to many different configuration items that share the same documentation. This is useful for processing instructions, technical documentation, etc. that need to be referenced from several places in the CMDB.

Document File Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Organization	Foreign key to a(n) Organization	Yes
Status	Possible values: Draft, Obsolete, Published	No
Version	Alphanumeric string	No
Document type	Foreign key to a(n) Document Type	No
Description	Multiline character string	No
File	Any binary content (document)	Yes

Tabs

Tab	Description
Cis	All the configuration items linked to this document
Contracts	All the contracts linked to this document
Services	All the services linked to this document

3.3.2. Document Note

A Note is used to store a short text document. HTML formatting of Notes is supported using the WYSIWYG editor.

Document Note Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Organization	Foreign key to a(n) Organization	Yes
Status	Possible values: Draft, Obsolete, Published	No
Version	Alphanumeric string	No
Document type	Foreign key to a(n) Document Type	No
Description	Multiline character string	No
Text	HTML string	Yes

Tabs

Tab	Description
Cis	All the configuration items linked to this document
Contracts	All the contracts linked to this document
Services	All the services linked to this document

3.4. Document Web

Web documents are used to store hyperlinks to external applications or documents. Think to them as "pointers" to external documents.

Document Web Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Organization	Foreign key to a(n) Organization	Yes
Status	Possible values: Draft, Obsolete, Published	No
Version	Alphanumeric string	No
Description	Multiline character string	No
URL	Absolute or relative URL as a text string	No

Tabs

Tab	Description
CIs	All the configuration items linked to this document
Contracts	All the contracts linked to this document
Services	All the services linked to this document

3.5. Configuration Items

3.5.1. Application Solution

Application Solutions describe complex applications that are made of (or depend on) several basic components. They are a specialized type of "Group" for documenting large applications. The main information conveyed by an Application Solution is its list of relationships.

Application Solution Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Organization	Foreign key to a(n) Organization	Yes
Status	Possible values: active, inactive	No
Business critic	Possible values: high, low, medium	No
Move to production date	Date (year-month-day)	No
Description	Multiline character string	No

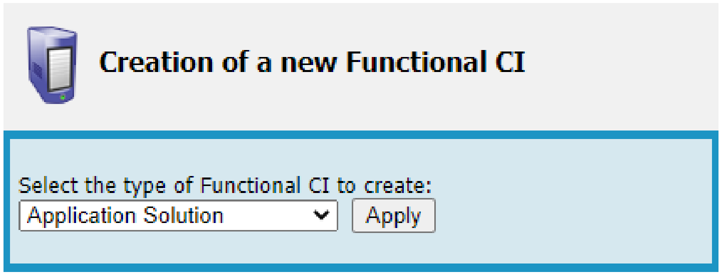
Tabs

Tab	Description
Contacts	All the contacts for this configuration item
Documents	All the documents linked to this configuration item
Tickets	All the tickets for this configuration item
Cis	All the configuration items that compose this application solution
Business processes	All the business processes depending on this application solution
Provider contracts	All the provider contracts for this configuration item
Services	All the services impacted by this configuration item

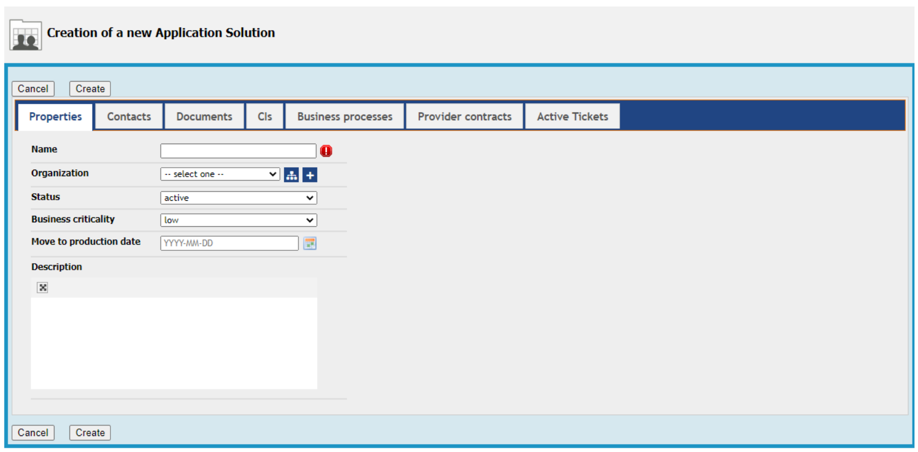
Creating a new Application Solution

Click on the "New CI" menu:

Then select “Application Solution” in the form below:



And click “Apply” to display the Application Solution creation form:



To link other configuration items to the new Application Solution, click on the tab “CIs” and use the buttons to Add/Remove CIs from the list. The tab “Contacts” is for managing the relationships with Contacts, etc.

Relations

- **Impacts:** an Application Solution is considered as impacting all its related Business Processes (listed in the tab “Business Processes”).

- **Depends on:** an Application Solution is considered as depending on all its related CIs (listed in the tab “CIs”).

3.5.2. Business Process

A Business Process is used to document a high-level process or an important application for the operations. It is quite similar to an Application Solution but for describing higher level applications or whole processes in the organization. The main information conveyed by a Business Process is its list of relationships with Application Solutions and Contacts.

Business Process Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Organization	Foreign key to a(n) Organization	Yes
Status	Possible values: active, inactive	No
Business critic	Possible values: high, low, medium	No
Move to production date	Date (year-month-day)	No
Description	Multiline character string	No

Tabs

Tab	Description
Contacts	All the contacts for this configuration item
Documents	All the documents linked to this configuration item
Tickets	All the tickets for this configuration item
Application solutions	All the application solutions that impact this business process

Creating a new Business Process

The creation of a Business Process object is similar to the creation of an Application Solution, except that its components are Application Solutions instead of any CIs.

Relations

- **Impacts:** no additional CIs are impacted by a Business Process.
- **Depends on:** a Business Process is considered as depending on all its related Application Solutions (listed in the tab "Application Solutions").

3.5.3. DB Server

An instance of a database server software (like MySQL, Oracle, SQL Server, DB2…) running on a particular system (PC, Server or Virtual Machine).

DB Server Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Organization	Foreign key to a(n) Organization	Yes
Status	Possible values: active, inactive	No

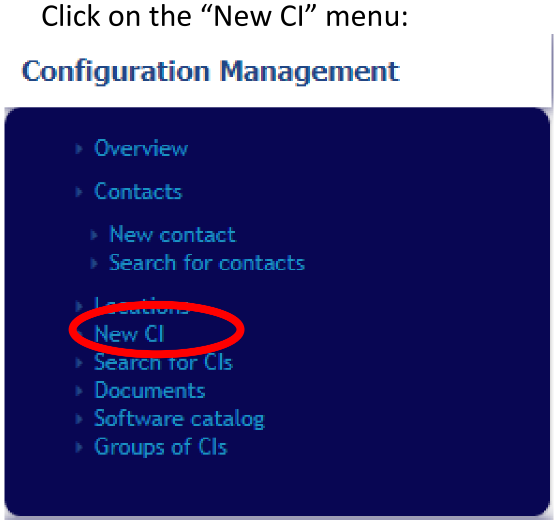
Business critic	Possible values: high, low, medium	No
System	Foreign key to a(n) Functional CI	Yes
Software	Foreign key to a(n) Software	No
Software license	Foreign key to a(n) Software License	No
Path	Alphanumeric string	No
Move to production date	Date (year-month-day)	No
Description	Multiline character string	No

Tabs

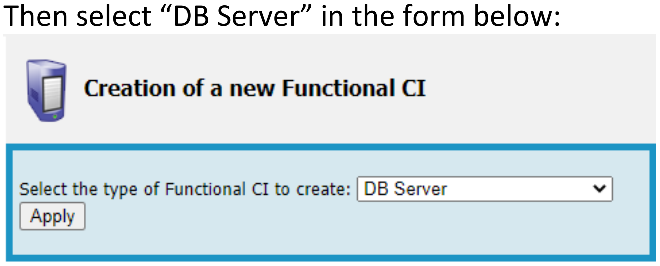
Tab	Description
Contacts	All the contacts for this configuration item
Documents	All the documents linked to this configuration item
Tickets	All the tickets for this configuration item
Application solutions	All the application solutions depending on this configuration item
DB schemas	All the database schemas for this DB server
Provider contracts	All the provider contracts for this configuration item
Services	All the services impacted by this configuration item

Creating a new DB Server

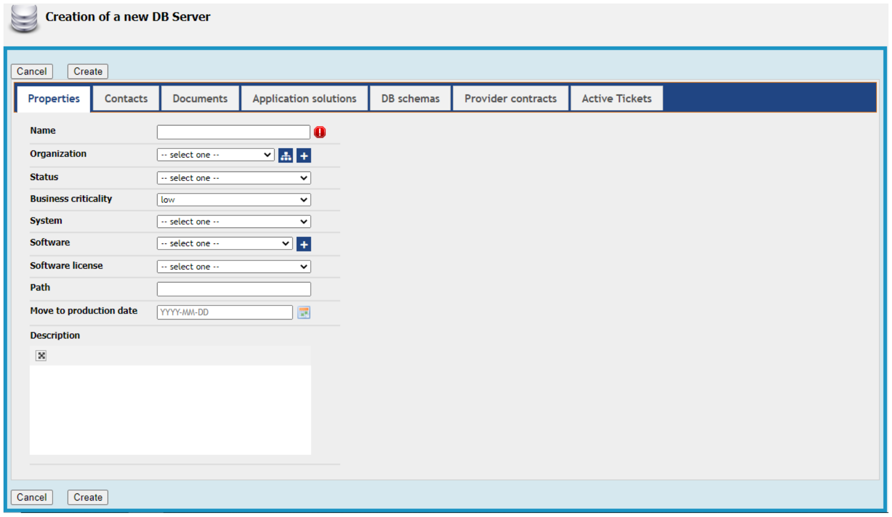
Click on the “New CI” menu:



Then select “DB Server” in the form below:



And click “Apply” to display the DB Server creation form:



Relations

- **Impacts:** a DB Server impacts all its Database Schemas, as Well as the Application Solutions it belongs to. A DB Server also impacts the contacts (Persons or Teams) linked to it.
- **Depends on:** a DB Server is considered as depending on the system (PC, Server or Virtual Machine) on which it runs.

3.5.4. Database Schema

An instance of database running of a specific DB server.

Database Schema Properties

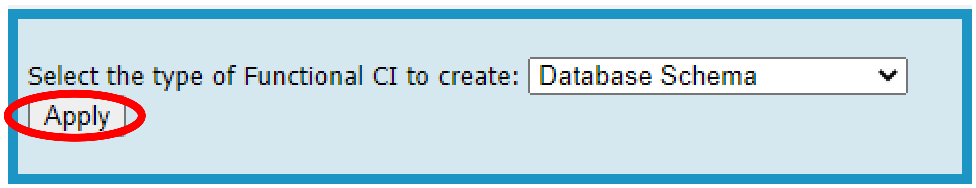
Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Organization	Foreign key to a(n) Organization	Yes
DB server	Foreign key to a(n) DB Server	Yes
Business critic	Possible values: high, low, medium	No
Move to production date	Date (year-month-day)	No
Description	Multiline character string	No

Tabs

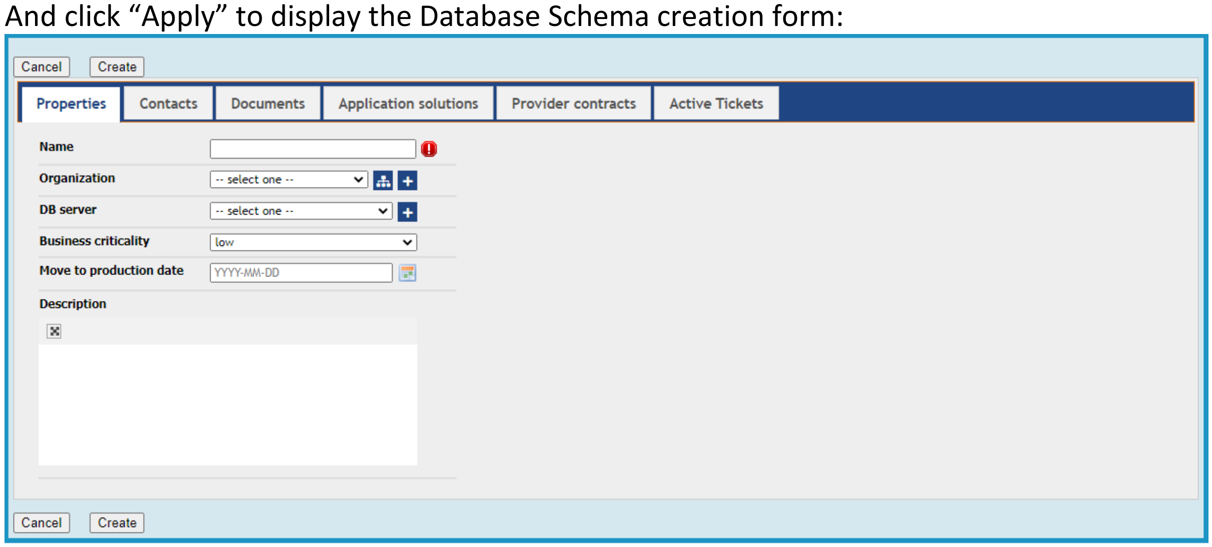
Tab	Description
Contacts	All the contacts for this configuration item
Documents	All the documents linked to this configuration item
Tickets	All the tickets for this configuration item
Application solutions	All the application solutions depending on this configuration item
Provider contracts	All the provider contracts for this configuration item
Services	All the services impacted by this configuration item

Creating a new Database Schema
Click on the “New CI” menu:

Then select “Database Schema” in the form below:



And click “Apply” to display the Database Schema creation form:



Relations

- **Impacts:** a Database Schema impacts all the Application Solutions it belongs to.
- **Depends on:** a Database Schema is considered as depending on its DB Server.

3.5.5. Middleware

An instance of a middleware software (software offering services to other software, or enterprise integration software) installed on a particular system (PC, Server or Virtual Machine). For example: Tomcat, JBoss, Talend, Microsoft BizTalk, IBM WebSphere or Lotus Domino can be put under this category.

Middleware Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Organization	Foreign key to a(n) Organization	Yes
Status	Possible values: active, inactive	No
Business critic	Possible values: high, low, medium	No
System	Foreign key to a(n) Functional CI	Yes
Software	Foreign key to a(n) Software	No
Software license	Foreign key to a(n) Software License	No
Path	Alphanumeric string	No
Move to production date	Date (year-month-day)	No
Description	Multiline character string	No

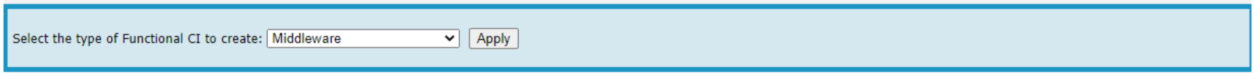
Tabs

Tab	Description
Contacts	All the contacts for this configuration item
Documents	All the documents linked to this configuration item
Tickets	All the tickets for this configuration item
Application solutions	All the application solutions depending on this configuration item
Middleware instances	All the middleware instances provided by this middleware
Provider contracts	All the provider contracts for this configuration item
Services	All the services impacted by this configuration item

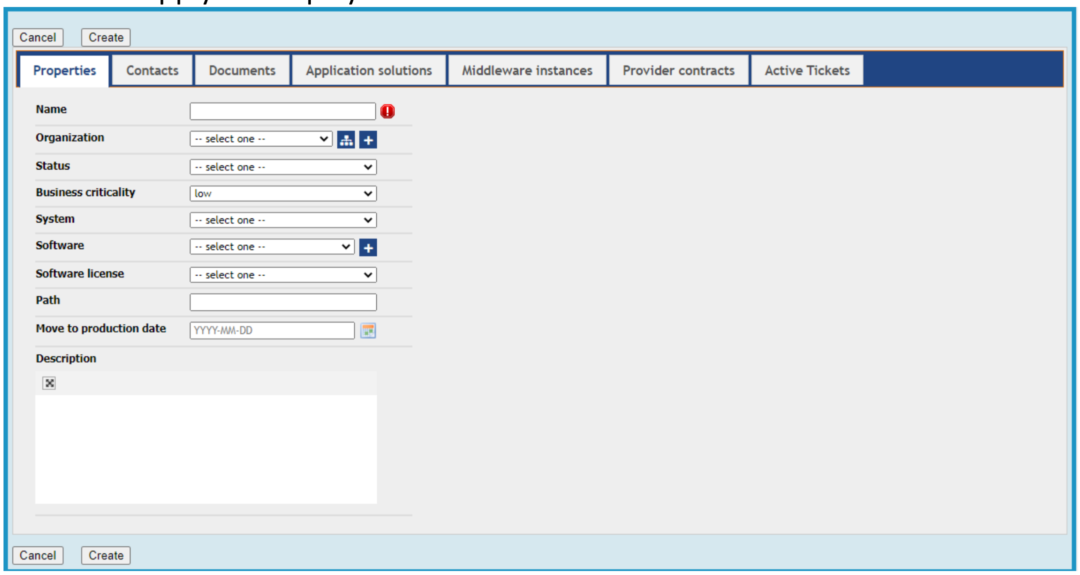
Creating a new Middleware

Click on the “New CI” menu:

Then select “Middleware” in the form below:



And click “Apply” to display the Middleware creation form:



Relations

- **Impacts:** a Middleware impacts all its Middleware Instances, as Well as the Application Solutions it belongs to. A Middleware also impacts the contacts (Persons or Teams) linked to it.

- **Depends on:** a Middleware is considered as depending on the system (PC, Server or Virtual Machine) on which it runs.

3.5.6. Middleware Instance

A particular application instance (or service) provided by an installed middleware software. For example, a web application deployed with Tomcat can be documented as a Middleware Instance.

Middleware Instance Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Organization	Foreign key to a(n) Organization	Yes
Middleware	Foreign key to a(n) Middleware	Yes
Business criticity	Possible values: high, low, medium	No
Move to production date	Date (year-month-day)	No

Description	Multiline character string	No

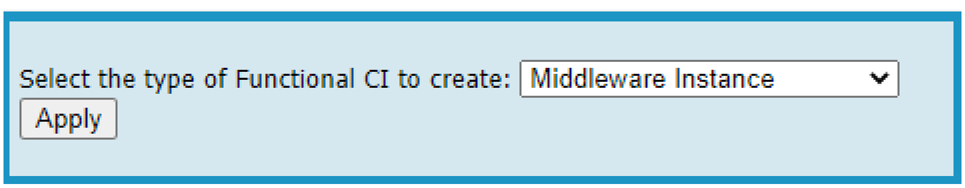
Tabs

Tab	Description
Contacts	All the contacts for this configuration item
Documents	All the documents linked to this configuration item
Tickets	All the tickets for this configuration item
Application solutions	All the application solutions depending on this configuration item
Provider contracts	All the provider contracts for this configuration item
Services	All the services impacted by this configuration item

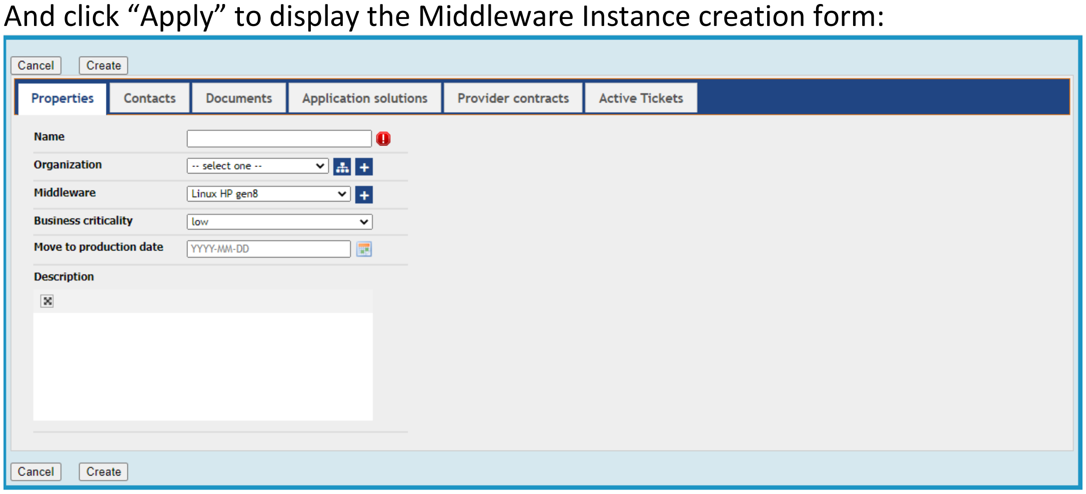
Creating a new Middleware Instance

Click on the “New CI” menu:

Then select “Middleware Instance” in the form below:



And click “Apply” to display the Middleware Instance creation form:



Relations

- **Impacts:** a Middleware Instance impacts all the Application Solutions it belongs to. A Middleware Instance also impacts the contacts (Persons or Teams) linked to it.

- **Depends on:** a Middleware Instance is considered as depending on its Middleware.

3.5.7. Network Device

Any type of network device: router, switch, hub, load balancer, firewall…

Network Device Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Organization	Foreign key to a(n) Organization	Yes
Status	Possible values: implementation, obsolete, production, stock	No
Business criticity	Possible values: high, low, medium	No
Location	Foreign key to a(n) Location	No
Rack	Foreign key to a(n) Rack	No
Enclosure	Foreign key to a(n) Enclosure	No
Network type	Foreign key to a(n) Network Device Type	Yes
Brand	Foreign key to a(n) Brand	No
Model	Foreign key to a(n) Model	No
IOS version	Foreign key to a(n) IOS Version	No
Management IP	IP Address	No
RAM	Alphanumeric string	No
Rack units	Numeric value (could be negative)	No
Serial number	Alphanumeric string	No
Asset number	Alphanumeric string	No
Move to production date	Date (year-month-day)	No
Purchase date	Date (year-month-day)	No
End of warranty	Date (year-month-day)	No
Power A source	Foreign key to a(n) Power Connection	No
Power source	Foreign key to a(n) Power Connection	No
Description	Multiline character string	No

Tabs

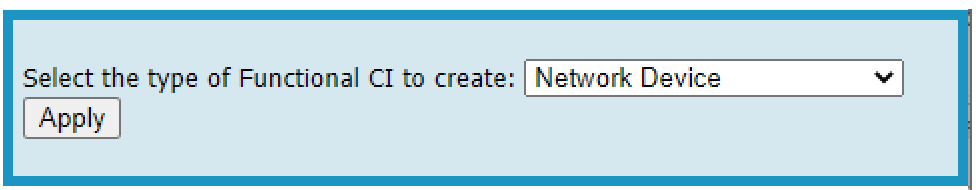
Tab	Description
Contacts	All the contacts for this configuration item
Documents	All the documents linked to this configuration item

Tickets	All the tickets for this configuration item
Network interfaces	All the physical network interfaces
Devices	All the devices connected to this network device
Provider contracts	All the provider contracts for this configuration item
Services	All the services impacted by this configuration item

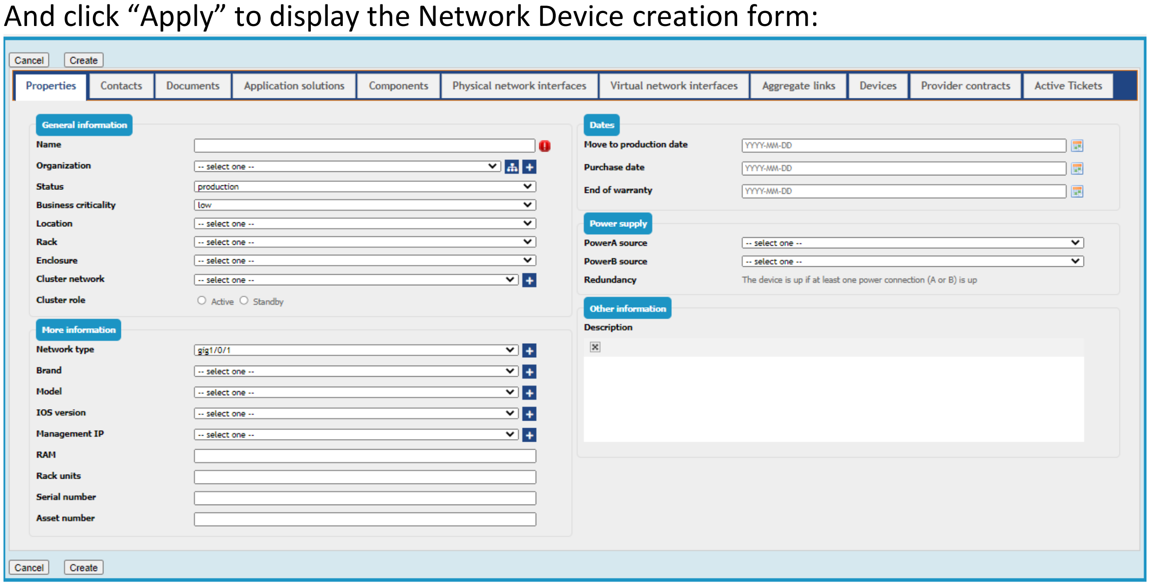
Creating a new Network Device

Click on the “New CI” menu:

Then select “Network Device” in the form below:



And click “Apply” to display the Network Device creation form:



Relations

- **Impacts:** a Network Device impacts all the devices connected to it (Servers, PCs, SAN Switches, NAS, Storage Systems, other Network Devices, etc.), the Application Solutions it belongs to and all the contacts (Persons or Teams) directly linked to it.

- **Depends on:** a Network Device is considered as depending on the other Network Devices and SAN Switches it is connected to. If the Network Devices is located in an Enclosure, it depends on it.

3.5.8. Other Software

Any type of software that do not fit in the other categories: PC Software, Middleware, DB server or Web Server.

Other Software Properties

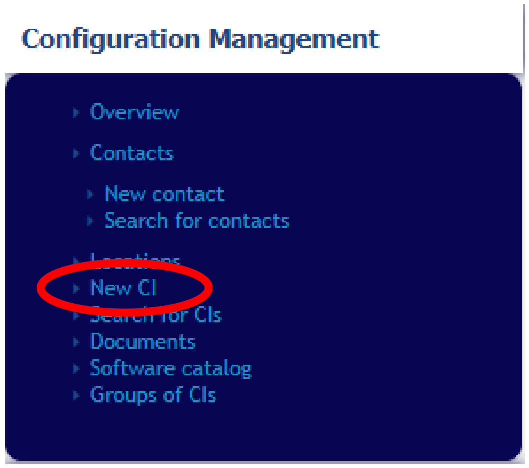
Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Organization	Foreign key to a(n) Organization	Yes
Status	Possible values: active, inactive	No
Business criticity	Possible values: high, low, medium	No
System	Foreign key to a(n) Functional CI	Yes
Software	Foreign key to a(n) Software	No
Software license	Foreign key to a(n) Software License	No
Path	Alphanumeric string	No
Move to production date	Date (year-month-day)	No
Description	Multiline character string	No

Tabs

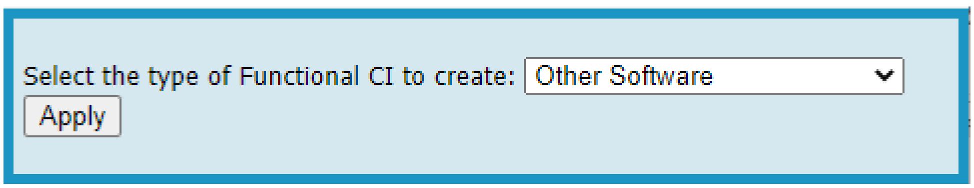
Tab	Description
Contacts	All the contacts for this configuration item
Documents	All the documents linked to this configuration item
Tickets	All the tickets for this configuration item
Application solutions	All the application solutions depending on this configuration item
Provider contracts	All the provider contracts for this configuration item
Services	All the services impacted by this configuration item

Creating a new Other Software

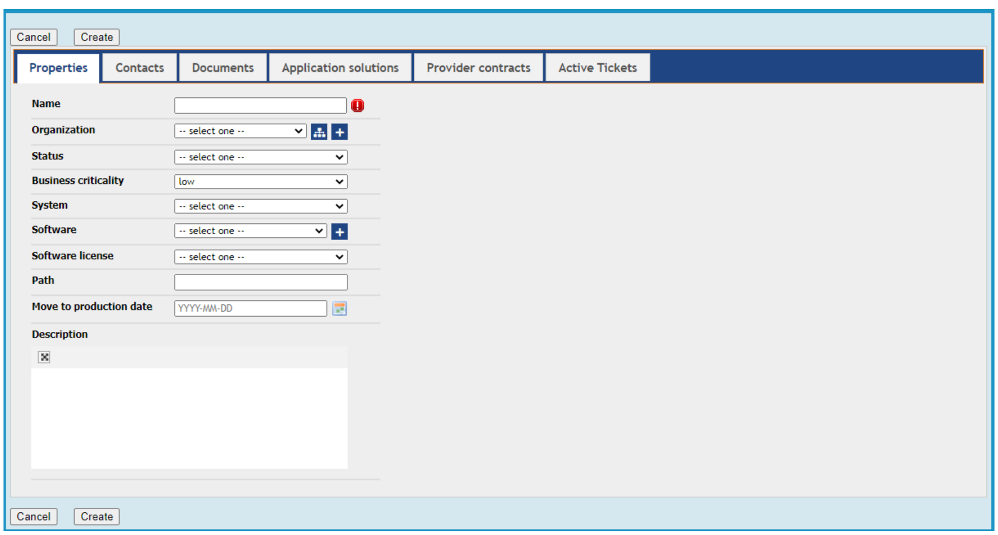
Click on the “New CI” menu:



Then select “Other Software” in the form below:



And click “Apply” to display the Other Software creation form:



Relations

- **Impacts:** an Other Software impacts all its instances, the Application Solutions it belongs to and all the contacts (Persons or Teams) directly linked to it.

- **Depends on:** an Other Software is considered as depending on the system (PC, Server or Virtual Machine) on which it runs.

3.5.9. PC Software

An instance of a software application installed on a PC. (For example: an instance of MS Office, Corel Draw, Adobe Photoshop or FileZilla).

PC Software Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Organization	Foreign key to a(n) Organization	Yes
Status	Possible values: active, inactive	No
Business criticity	Possible values: high, low, medium	No
System	Foreign key to a(n) Functional CI	Yes
Software	Foreign key to a(n) Software	No
Software license	Foreign key to a(n) Software License	No
Path	Alphanumeric string	No
Move to production date	Date (year-month-day)	No
Description	Multiline character string	No

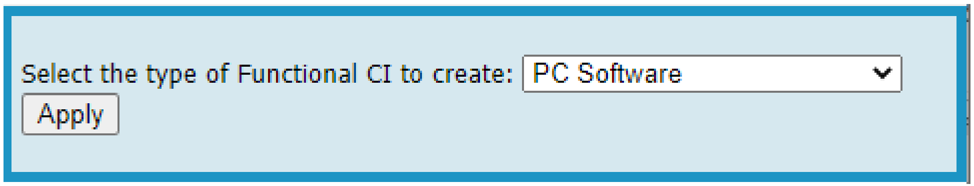
Tabs

Tab	Description
Contacts	All the contacts for this configuration item
Documents	All the documents linked to this configuration item
Tickets	All the tickets for this configuration item
Application solutions	All the application solutions depending on this configuration item
Provider contracts	All the provider contracts for this configuration item
Services	All the services impacted by this configuration item

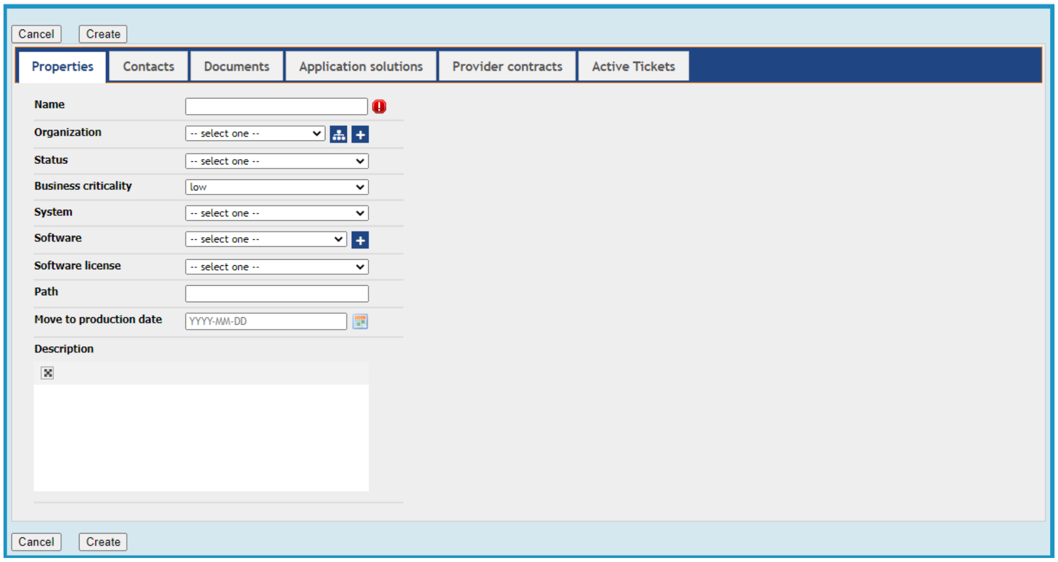
Creating a new PC Software

Click on the “New CI” menu:

Then select “PC Software” in the form below:



And click “Apply” to display the PC Software creation form:



Relations

- **Impacts:** a PC Software impacts all its instances, the Application Solutions it belongs to and all the contacts (Persons or Teams) directly linked to it.
- **Depends on:** a PC Software is considered as depending on the PC on which it runs.

3.5.10. Server

A Server is typically used in tbITSM v.1.0. to document a physical server machine. The Server object documents asset management information, physical characteristics and operational information (contracts and services). A Server can be related to software components, contacts and documents. More relationships can be added to Servers by other modules like the module “Advanced Storage”.

Server Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Organization	Foreign key to a(n) Organization	Yes
Status	Possible values: implementation, obsolete, production, stock	No
Business criticity	Possible values: high, low, medium	No

Location	Foreign key to a(n) Location	No
Rack	Foreign key to a(n) Rack	No
Enclosure	Foreign key to a(n) Enclosure	No
Brand	Foreign key to a(n) Brand	No
Model	Foreign key to a(n) Model	No
OS family	Foreign key to a(n) OS Family	No
OS version	Foreign key to a(n) OS Version	No
Management IP	IP Address	No
OS license	Foreign key to a(n) OS License	No
CPU	Alphanumeric string	No
RAM	Alphanumeric string	No
Rack units	Numeric value (could be negative)	No
Serial number	Alphanumeric string	No
Asset number	Alphanumeric string	No
Move to production date	Date (year-month-day)	No
Purchase date	Date (year-month-day)	No
End of warranty	Date (year-month-day)	No
Power A source	Foreign key to a(n) Power Connection	No
Power B source	Foreign key to a(n) Power Connection	No
Description	Multiline character string	No

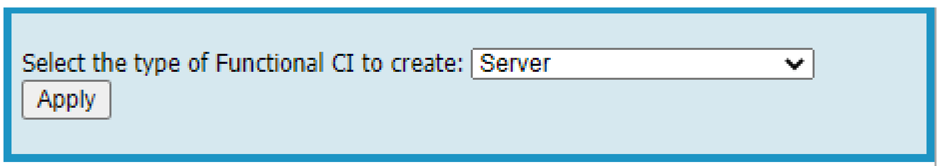
Tabs

Tab	Description
Software	All the software installed on this configuration item
Contacts	All the contacts for this configuration item
Documents	All the documents linked to this configuration item
Tickets	All the tickets for this configuration item
Network interfaces	All the physical network interfaces
FC ports	All the fiber channel interfaces for this device
Network devices	All network devices connected to this device
SANs	All the SAN switches connected to this device
Logical volumes	All the logical volumes connected to this server
Provider contracts	All the provider contracts for this configuration item
Services	All the services impacted by this configuration item

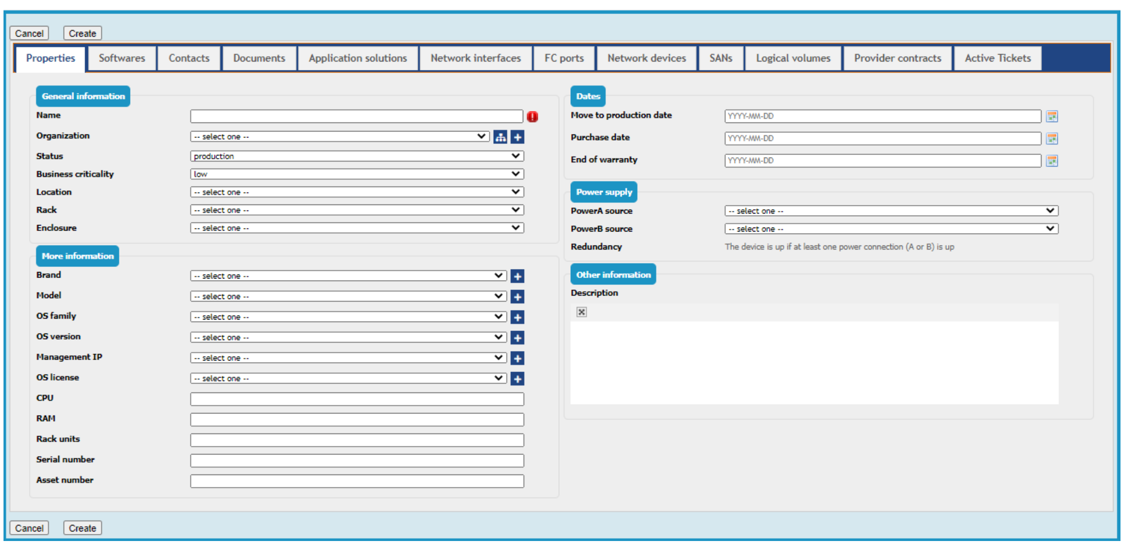
Creating a New Server

Click on the “New CI” menu:

Then select “PC Software” in the form below:



And click “Apply” to display the Server creation form:



Relations

- **Impacts:** a Server impacts all the software instances running on it, as Well as hypervisors. A Server is also considered as impacting any Application Solution it belongs to.

- **Depends on:** a Server is considered as depending on the Logical Volumes, SAN Switches and the Network Devices it is connected to. If the Server is located in an Enclosure, it is considered as depending on it.

3.5.11. Web Application

An instance of an application accessible using a web browser and that runs on a given Web Server instance.

Web Application Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Organization	Foreign key to a(n) Organization	Yes
Web server	Foreign key to a(n) Web server	Yes
URL	Absolute or relative URL as a text string	No
Business criticity	Possible values: high, low, medium	No
Move to production date	Date (year-month-day)	No
Description	Multiline character string	No

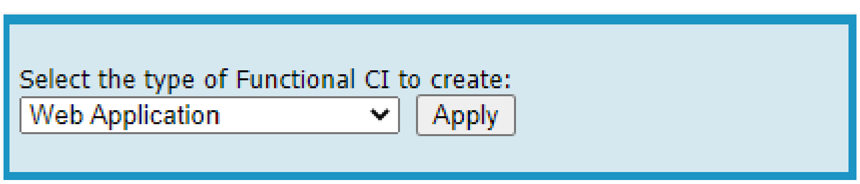
Tabs

Tab	Description
Contacts	All the contacts for this configuration item
Documents	All the documents linked to this configuration item
Tickets	All the tickets for this configuration item
Application solutions	All the application solutions depending on this configuration item
Provider contracts	All the provider contracts for this configuration item
Services	All the services impacted by this configuration item

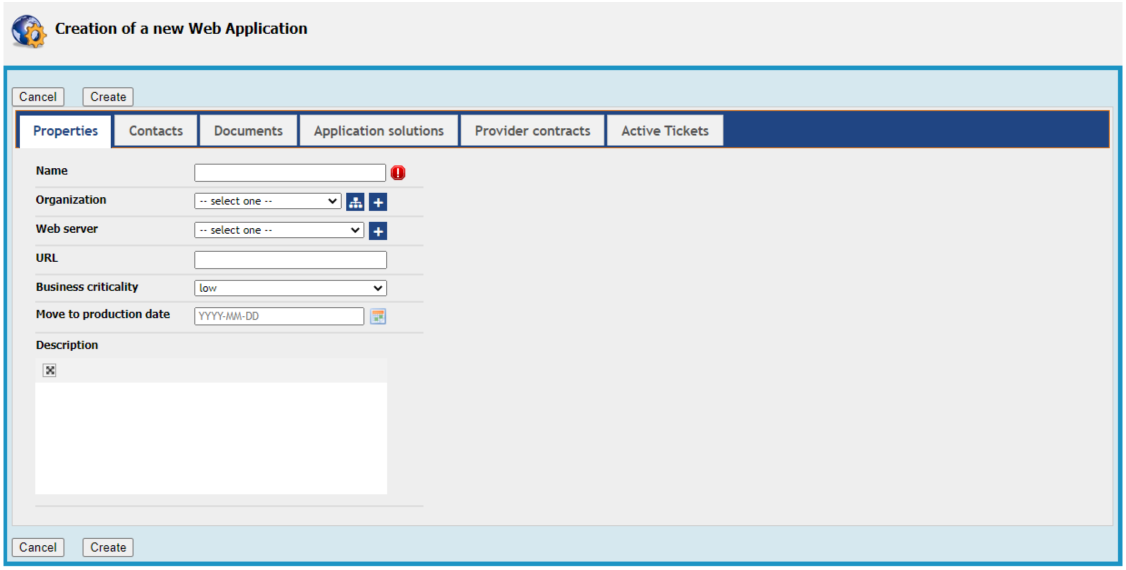
Creating a new Web Application

Click on the "New CI" menu:

Then select "Web Application" in the form below:



And click “Apply” to display the Web Application creation form:



Relations

- **Impacts:** a Web Application impacts the Application Solutions it belongs to and all the contacts (Persons or Teams) directly linked to the Web Application.

- **Depends on:** a Web Application is considered as depending on the Web Server on which it runs.

3.5.12. Web Server

An instance of a web server software running on a specific system (PC, Server or Virtual Machine).

Web Server Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Organization	Foreign key to a(n) Organization	Yes
Status	Possible values: active, inactive	No
Business criticity	Possible values: high, low, medium	No
System	Foreign key to a(n) Functional CI	Yes
Software	Foreign key to a(n) Software	No
Software license	Foreign key to a(n) Software License	No
Path	Alphanumeric string	No
Move to production date	Date (year-month-day)	No
Description	Multiline character string	No

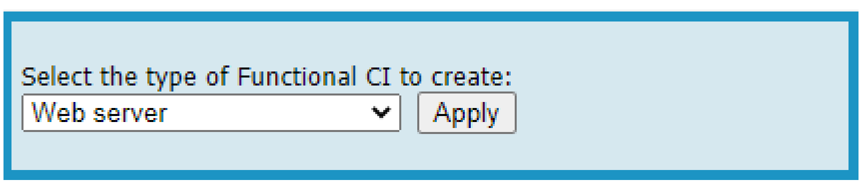
Tabs

Tab	Description
Contacts	All the contacts for this configuration item
Documents	All the documents linked to this configuration item
Tickets	All the tickets for this configuration item
Application solutions	All the application solutions depending on this configuration item
Web applications	All the web applications available on this web server
Provider contracts	All the provider contracts for this configuration item
Services	All the services impacted by this configuration item

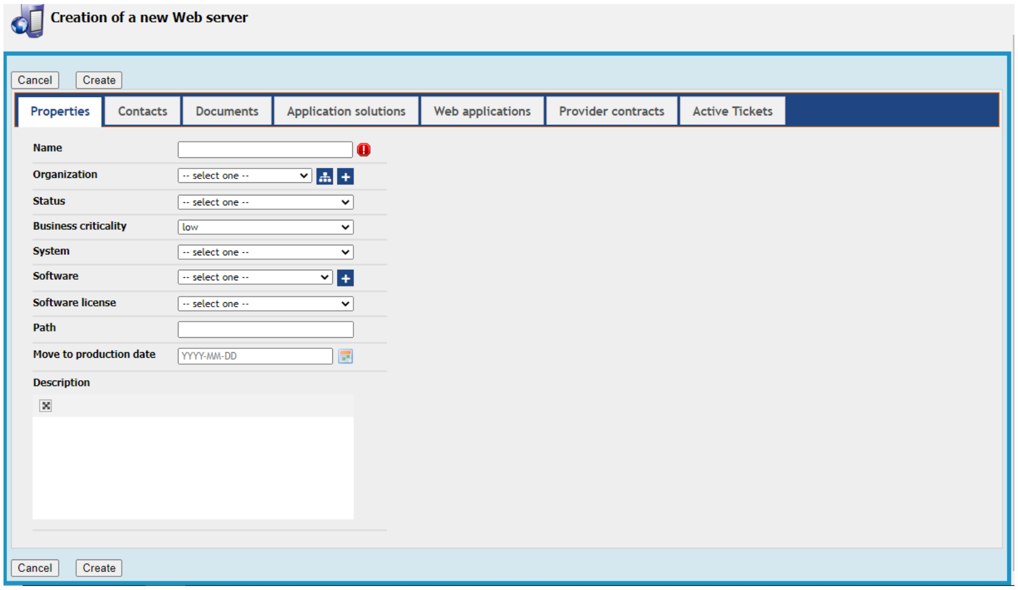
Creating a new Web Server

Click on the “New CI” menu:

Then select “Web Server” in the form below:



And click “Apply” to display the Web Server creation form:



Relations

- **Impacts:** a Web Server impacts all the Web Applications running on it, the Application Solutions it belongs to and all the contacts (Persons or Teams) directly linked to the Web Server.

- **Depends on:** a Web Server is considered as depending on the system (Server, PC or Virtual machine) on which it runs.

3.5.13. Miscellaneous

Group

Any arbitrary group of CIs. Groups can be used to define explicit sets of configurations items for any documentation purpose. For example, when doing an OS migration, it can be convenient to create a group “Servers to be migrated” and to remove the servers from the group as the migration progresses.

Group Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Status	Possible values: Implementation, Obsolete, Production	Yes
Organization	Foreign key to a(n) Organization	Yes
Type	Alphanumeric string	No
Description	Multiline character string	No
Parent Group	Foreign key to a parent Group	No

Tabs

Tab	Description
Linked CIs	All the configuration items linked to this group

Location

A Location object in tbITSM v.1.0. is used to document the physical location of other configuration items. A Location basically has an address and therefore is most often used to describe a physical site. It is a good practice to ensure that for a given Organization, Location names are unique.

Location Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Status	Possible values: Active, Inactive	No
Owner organization	Foreign key to a(n) Organization	Yes
Address	Multiline character string	No
Postal code	Alphanumeric string	No
City	Alphanumeric string	No
Country	Alphanumeric string	No

Tabs

Tab	Description
Contacts	All the contacts located on this location
Devices	All the devices in this location

OS License

A license contract for a particular operating system. The license is related to one operating system (for example Windows 2008 R2) and can be associated with several servers or virtual machines.

OS License Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
OS version	Foreign key to a(n) OS Version	Yes
Organization	Foreign key to a(n) Organization	Yes
Usage limit	Alphanumeric string	No
Description	Multiline character string	No
Perpetual	Possible values: no, yes	Yes

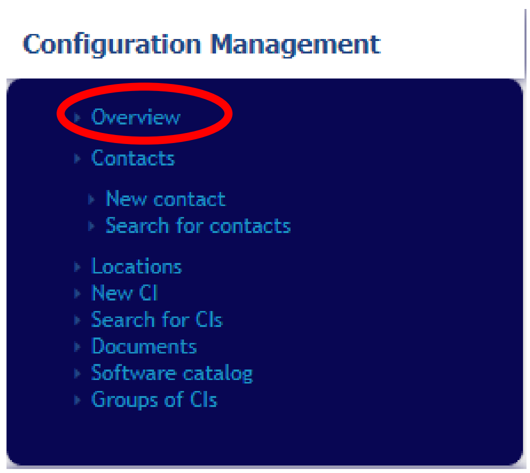
Start date	Date (year-month-day)	No
End date	Date (year-month-day)	No
Key	Alphanumeric string	No

Tabs

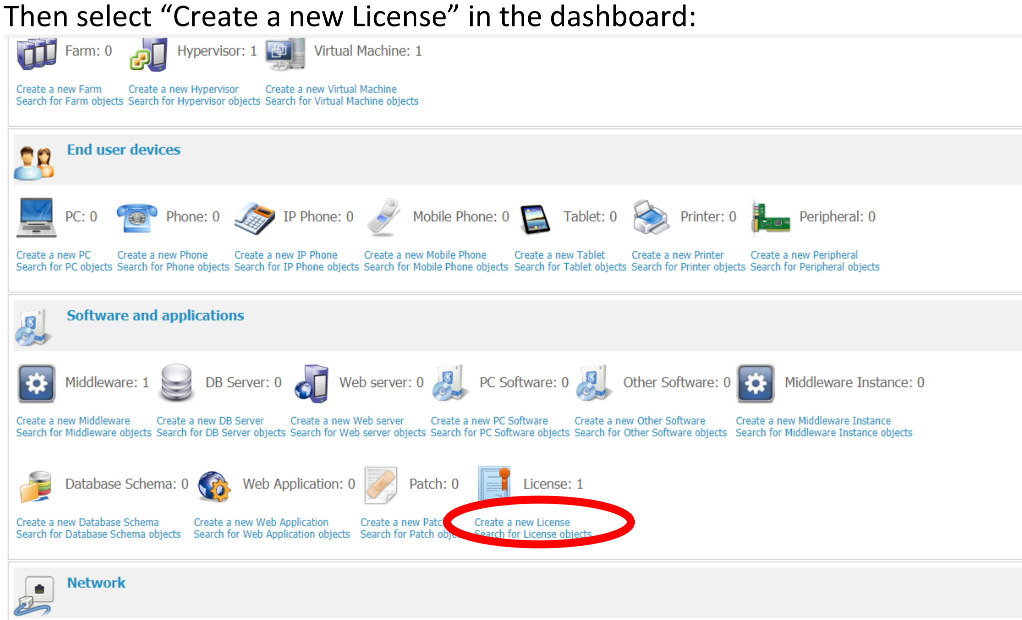
Tab	Description
Documents	All the documents linked to this license
servers	All the servers where this license is used
Virtual machines	All the virtual machines where this license is used

Creating a new OS License

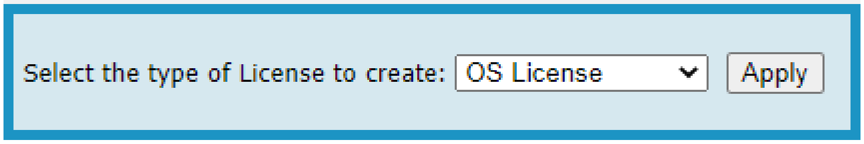
Click on the “Overview” menu:



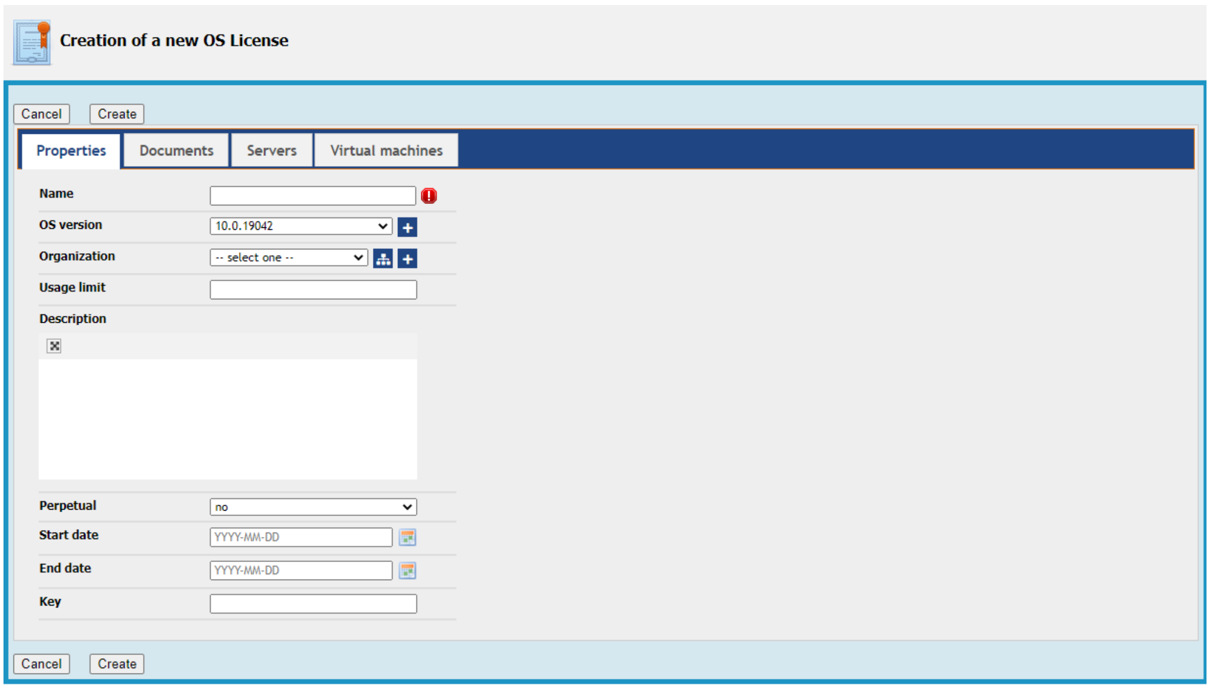
Then select “Create a new License” in the dashboard:



Then select “OS License” in the form below:



And click “Apply” to display the OS License creation form:



OS Patch
A patch, hotfix, security fix or service pack for a given operating system.

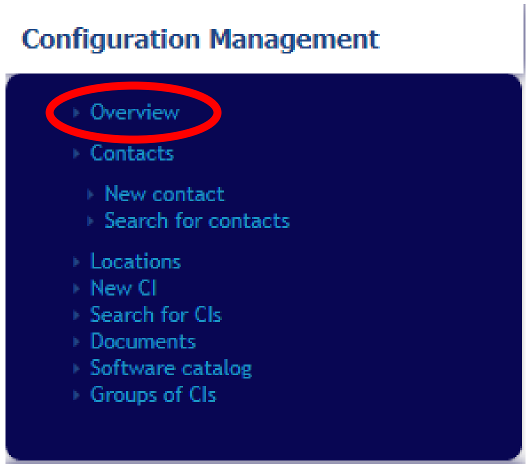
OS Patch Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Description	Multiline character string	No
OS version	Foreign key to a(n) OS Version	Yes

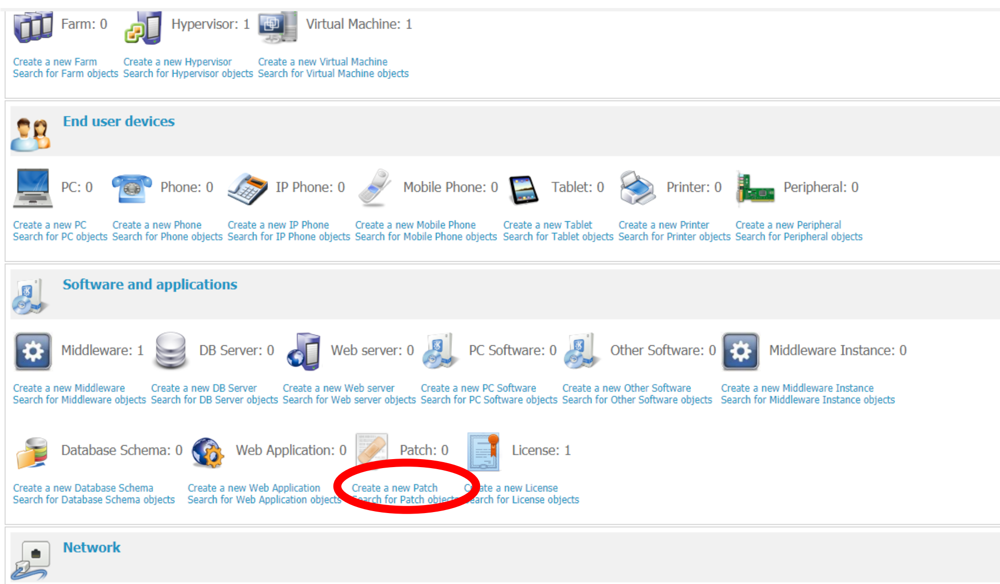
Tabs

Tab	Description
Documents	All the documents linked to this patch
Devices	All the systems where this patch is installed

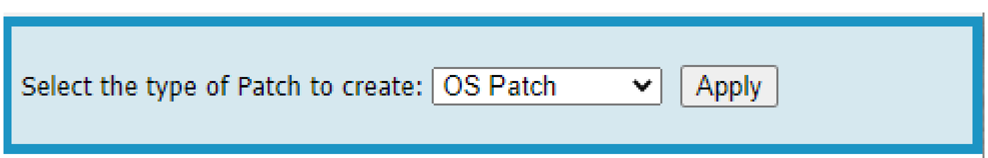
Creating a new OS Patch
Click on the “Overview” menu:



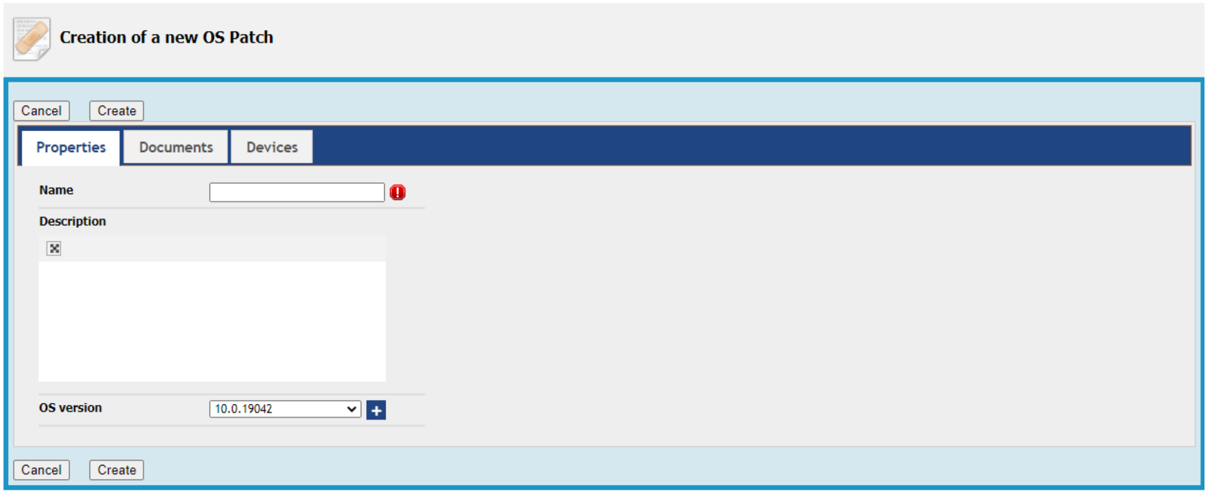
Then select “Create a new Patch” in the dashboard:



Then select “OS Patch” in the form below:



And click “Apply” to display the OS Patch creation form:



Organizations

The Organization is the base entity for the ownership of configuration items in tbITSM v.1.0. Organizations are also used in tbITSM v.1.0. to document customers, providers, and organizational units within the same company. To describe complex structures, Organizations can be arranged into a hierarchy (by use of the optional “Parent” field).

Organization Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Code	Alphanumeric string	No
Status	Possible values: Active, Inactive	No
Parent	Foreign key to a parent Organization	No
Delivery model	Foreign key to a(n) Delivery Model	No

Physical Interface

A physical network interface on a physical server, as opposed to virtual interfaces that are defined at the software level either in operating systems or virtual machines.

Physical Interface Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Device	Foreign key to a(n) Connectable CI	Yes
IP Address	IP Address	No
MAC address	Core: AttributeMacAddress+	No
Comment	Multiline character string	No
Speed	Decimal value (could be negative)	No

Tab	Description
VLANs	VLANs to which this interface is connected

Software
A generic item of a software in the software catalog. Software instances installed on a particular system always refer to a given Software. tbITSM v.1.0. splits the software in 5 categories: DB Servers, Middleware, PC Software, Web Servers and Other Software.

Software Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
vendor	Alphanumeric string	Yes
Version	Alphanumeric string	Yes
Type	Possible values: DB Server, Middleware, Other Software, PC Software, Web Server	No

Tabs

Tab	Description
Documents	All the documents linked to this software
Software Instances	All the software instances for this software
Software Patches	All the patches for this software
Software Licenses	All the licenses for this software

Software License
A license contract for a particular software. The license is related to one software (for example MS Office 2010) and can be associated with several instances of this software.

Software License Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Software	Foreign key to a(n) Software	Yes
Organization	Foreign key to a(n) Organization	Yes
Usage limit	Alphanumeric string	No
Description	Multiline character string	No
Perpetual	Possible values: no, yes	Yes
Start date	Date (year-month-day)	No
End date	Date (year-month-day)	No
Key	Alphanumeric string	No

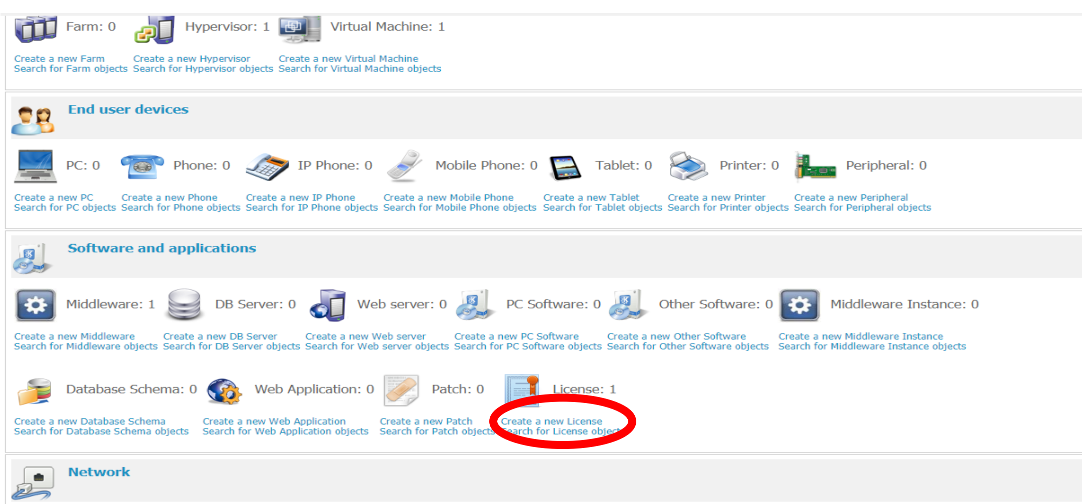
Tabs

Tab	Description
Documents	All the documents linked to this license
Software instances	All the systems where this license is used

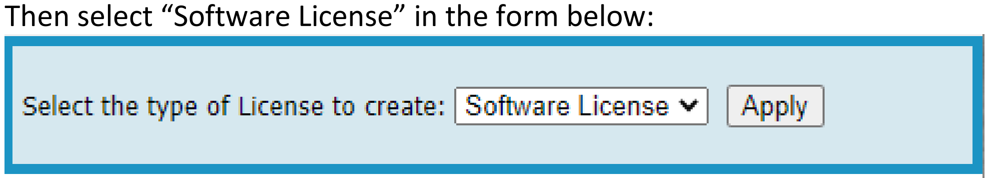
Creating a new Software License

Click on the “Overview” menu:

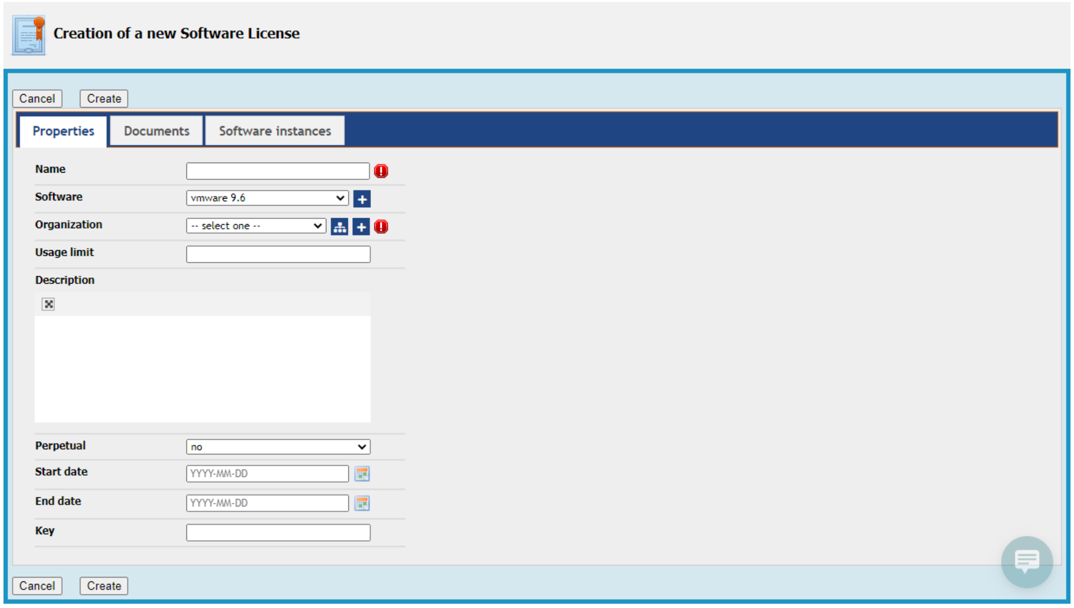
Then select “Create a new License” in the dashboard:



Then select “Software License” in the form below:



And click "Apply" to display the Software License creation form:



Software Patch
A patch, hotfix, security fix or service pack for a given software.

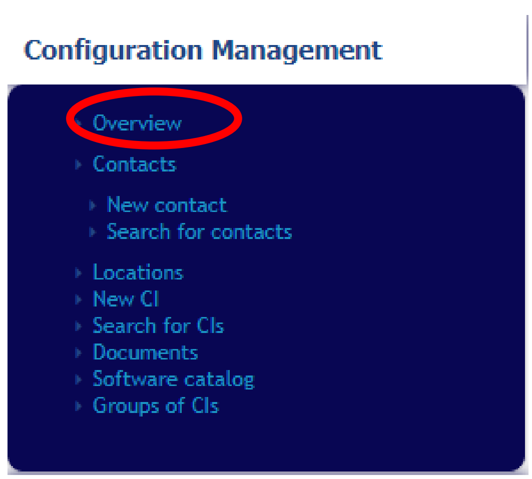
Software Patch Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Software	Foreign key to a(n) Software	Yes
Description	Multiline character string	No

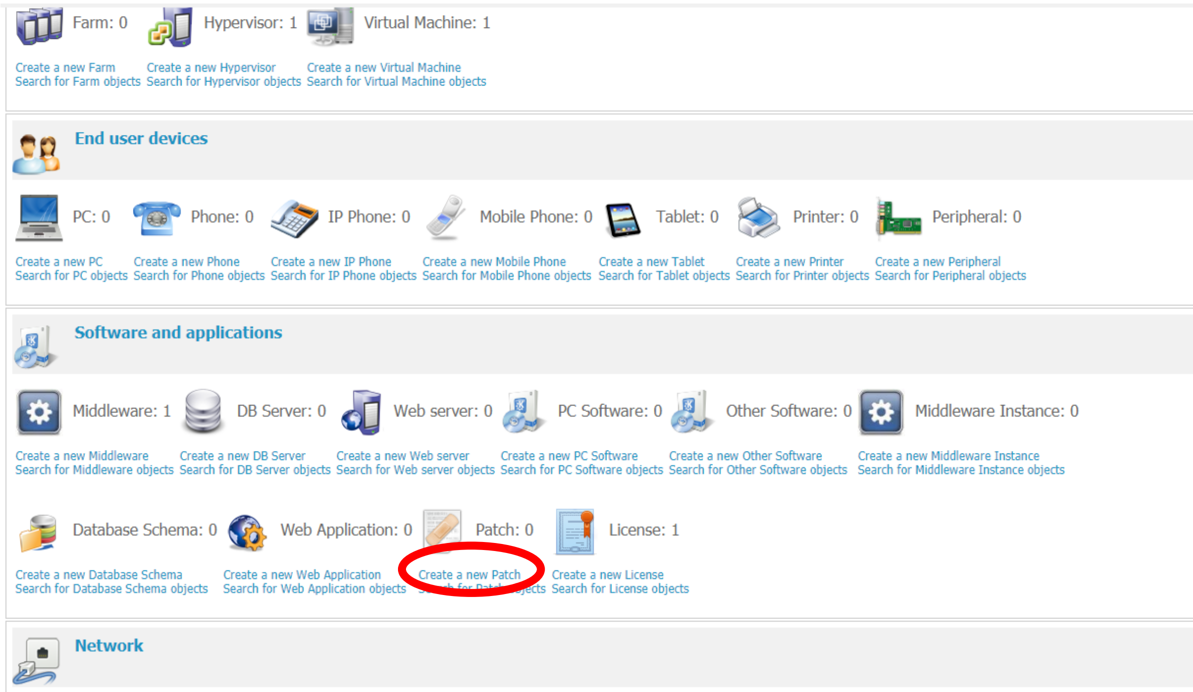
Tabs

Tab	Description
Documents	All the documents linked to this patch
Software instances	All the systems where this software patch is installed

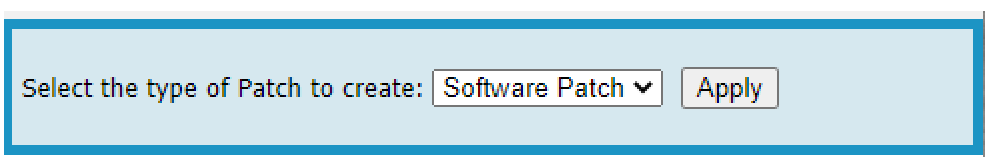
Creating a new Software Patch
Click on the "Overview" menu:



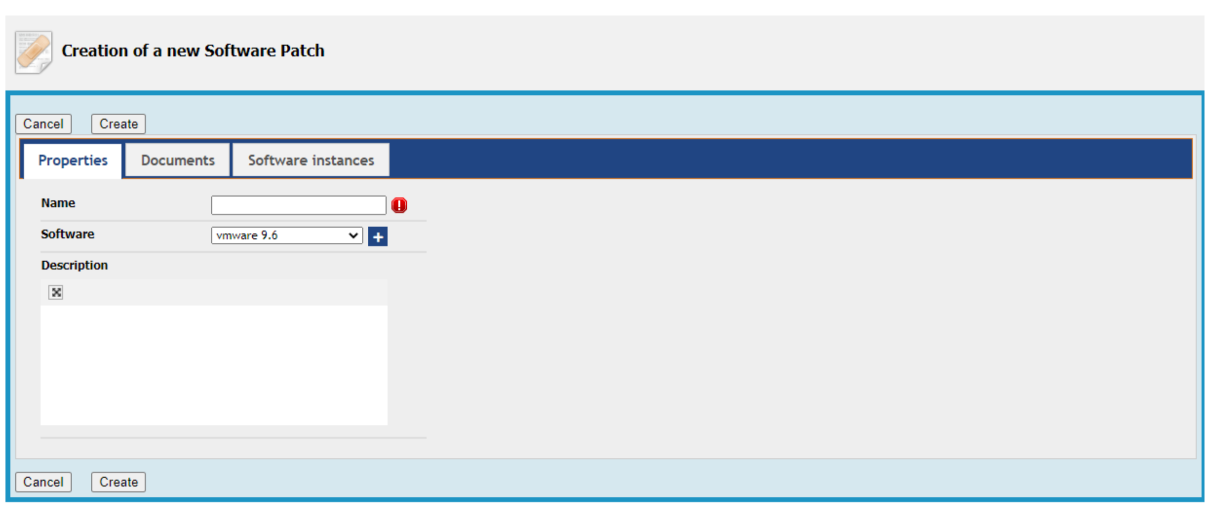
Then select “Create a new Patch” in the dashboard:



Then select “Software Patch” in the form below:



And click “Apply” to display the Software Patch creation form:



VLAN

A Virtual LAN is used to group together in a logical way networks Subnets and Physical Interfaces that participate in the same VLAN.

VLAN Properties

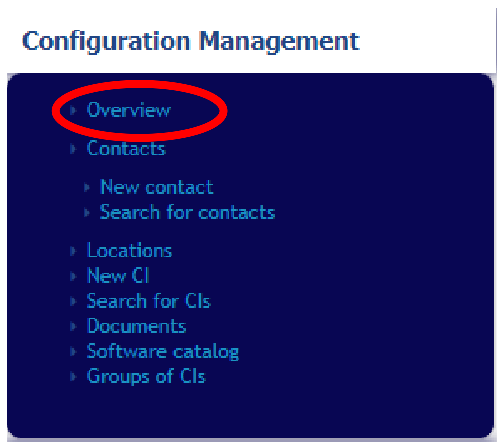
Name	Type	Mandatory?
VLAN Tag	Alphanumeric string	Yes
Organization	Foreign key to a(n) Organization	Yes
Description	Multiline character string	No

Tabs

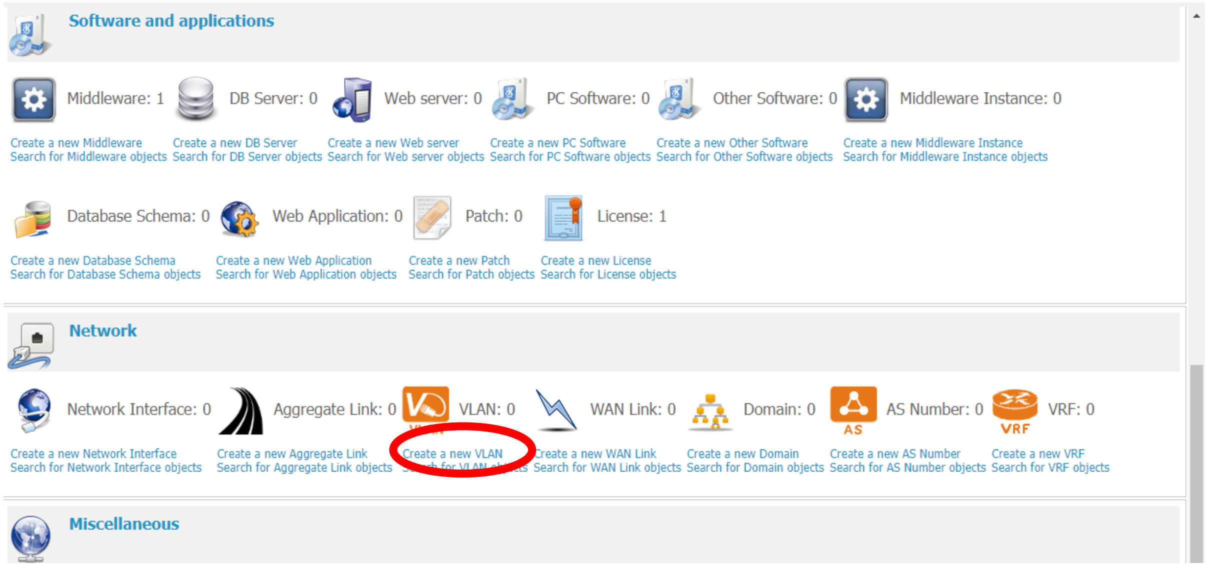
Tab	Description
Subnets	Subnets which are part of this VLAN
Physical network interfaces	Interfaces which are part of this VLAN

Creating a new VLAN

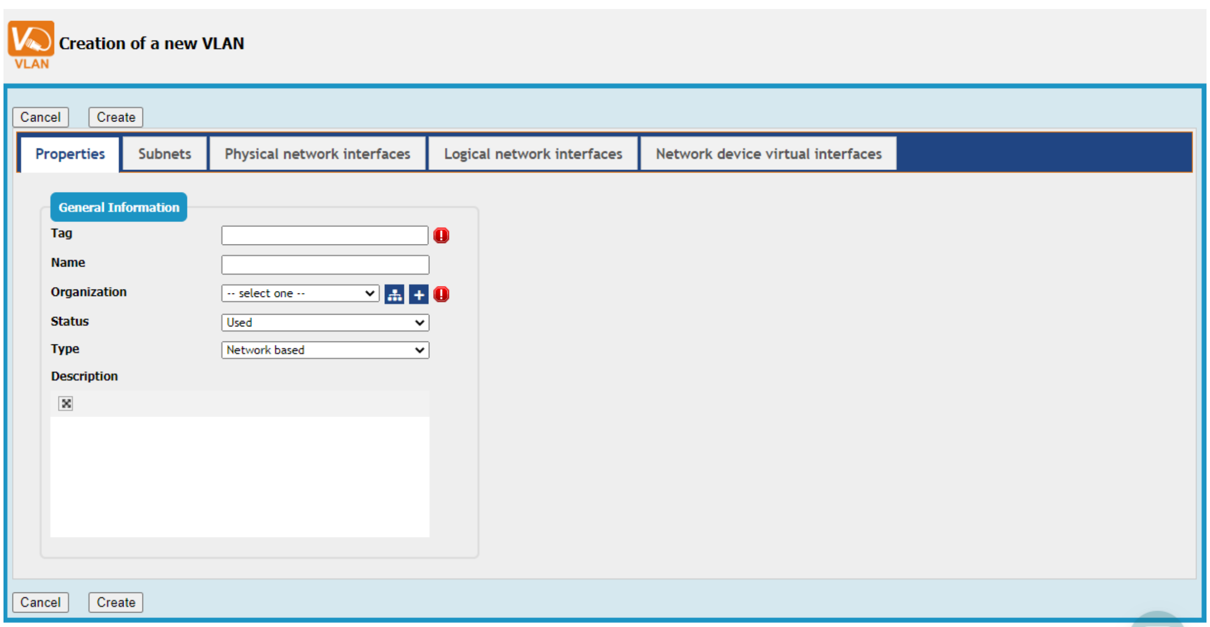
Click on the “Overview” menu:



Then select “Create a new VLAN” in the dashboard:



To display the following form:



Typology

Type	Description
Brand	The possible values for the “Brand” of a device.
Contact Type	Possible values for the “role” of a Person inside a Team or a Delivery Model.
Document Type	
IOS Version	Possible values for the types of operating system for network devices (from Cisco's Internetwork Operating System).
Model	The possible values for the “Model” of a device. The list of models is organized per brand in tbITSM v.1.0. .

Network Device Type	The possible values for the “Type” of a Network Device (e.g., Router, Switch, Firewall, etc.)
OS Family	The possible values for the “OS Family” attribute of Servers, Virtual Machines and PCs. A sample list can be: * CentOS * Debian * Fedora * HP-UX * MacOS X * RedHat * Solaris * SuSE * Ubuntu * Windows
OS Version	The possible values for the “OS Version” of a computer. The OS Versions are organized per OS Family in tbITSM v.1.0. . For example, if you define the OS Type “MacOS X”, the versions for this OS Type can be: * 10.0 - Cheetah * 10.1 - Puma * 10.2 - Jaguar * 10.3 - Panther * 10.4 - Tiger, etc. Whereas for the OS Type “Windows”, the versions can be something like: * XP * XP SP1 * XP SP2 * XP SP3 * Vista * Vista SP1 * Vista SP2 * 7 * 7 SP 1 * Server 2008 * Server 2008 R2 * Server 2012, etc.

3.6. Fetching of Adhoc Report

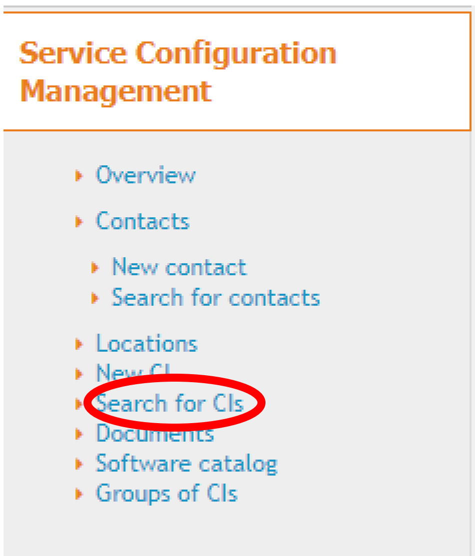
Adhoc report may have tabular data arrangement, a matrix as well as a chart. On Adhoc Wizard, you can also:

- Sort the data to re-arrange it in order of your choice
- Get totals / summaries
- Highlights selective records (when a condition is met)

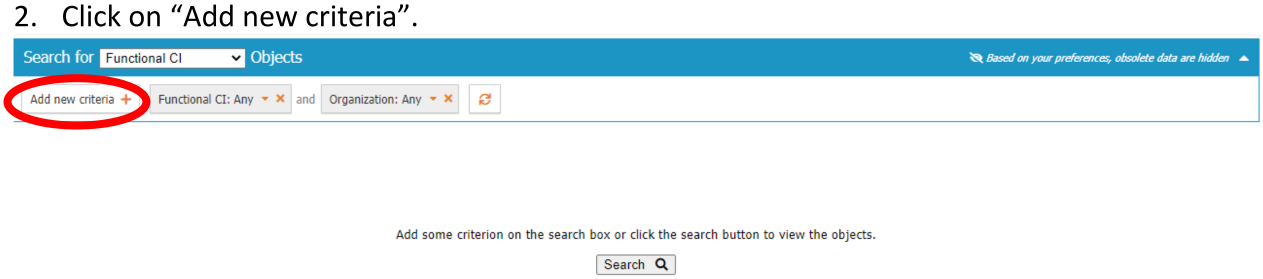
When you run an ad hoc report in output format, the viewer provides Power Viewer that helps altering the report on the viewer itself.

Steps for fetching Adhoc Report: -

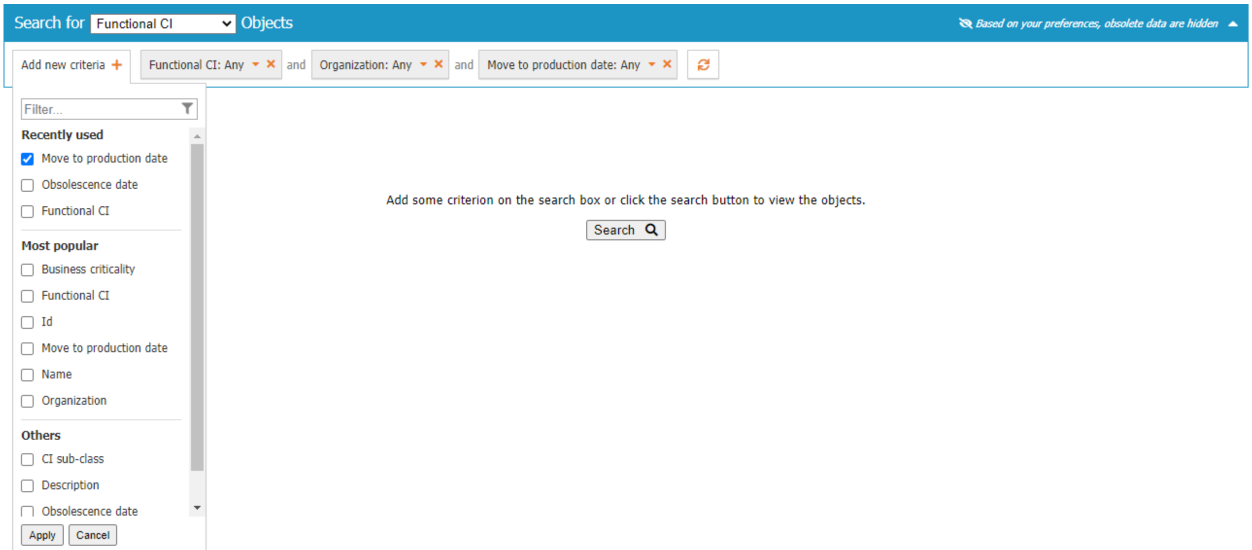
1. Click on Search for CIs.



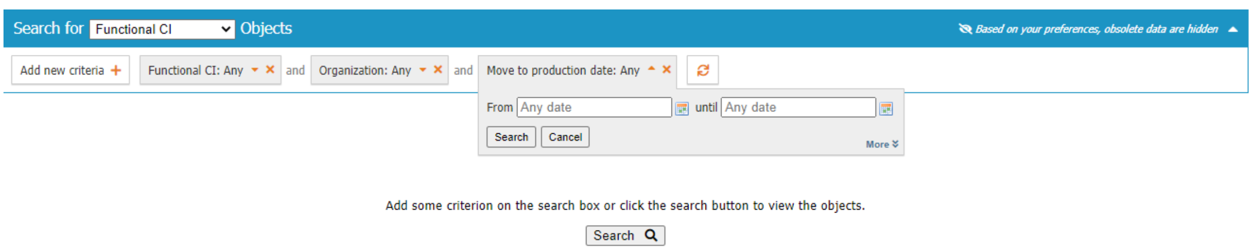
2. Click on "Add new criteria".



3. Select the Criteria.



For example, if you select Move to production date, then you will get the display as:



4. Then write the values in the given fields and then click on Search.

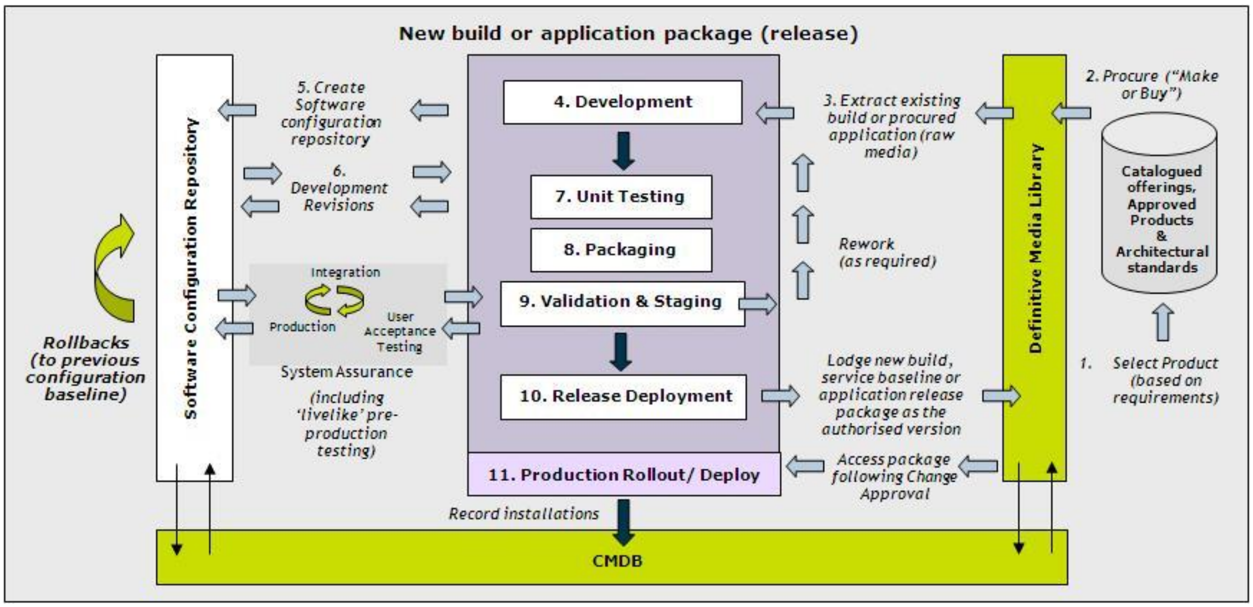
5. The display will be shown as below:

3.7. Definitive Media Library(DML)

A **Definitive Media Library** is a secure Information Technology repository in which an organisation's definitive, authorised versions of software media are stored and protected. Before an organisation releases any new or changed application software into its operational environment, any such software should be fully tested and quality assured. The Definitive Media Library provides the storage area for software objects ready for deployment and should only contain master copies of controlled software media configuration items (CIs) that have passed appropriate quality assurance checks, typically including both procured and bespoke application and gold build source code and executables. In the context of the ITIL best practice framework, the term Definitive Media Library supersedes the term definitive software library referred to prior to version ITIL v3.

In conjunction with the configuration management database (CMDB), it effectively provides the DNA of the data center i.e. all application and build software media connected to the CMDB record of installation and configuration.

The Definitive Media Library (DML) is a primary component of an organisation's release and provisioning framework and service continuity plan.



3.7.1. Scope

The DML plays a critical role in supporting the transition from development to production phases and DML solutions should be distinguished from other software and source code repositories e.g. Software Configuration Management or SCM (sometimes referred to as Software Change and Configuration Management) that supports the development or software evolution phase. This is an important distinction and often causes some confusion. In essence, whereas SCM tools or repositories store and manage all development versions and revisions of code (or work products) up to but not including the final authorised product, the DML stores only the final authorised versions of the code or product. This is analogous to a high-street product lifecycle where the product moves from design house to factory, through to warehouse and then shop, i.e.

- records (metadata) are kept about how a product is designed developed and built. This enables the tracking down of which process is to blame where faulty products are discovered either during quality control or even in later service.

- records (metadata) are kept in a CMDB about where the software is installed and deployed from the DML and into the production environment. Each installation or deployment should be authorised by a corresponding production change request and the resulting change recorded in the CMDB as a relationship between the DML artefact and the platform where it has been deployed.

3.7.2. Media Release Lifecycle

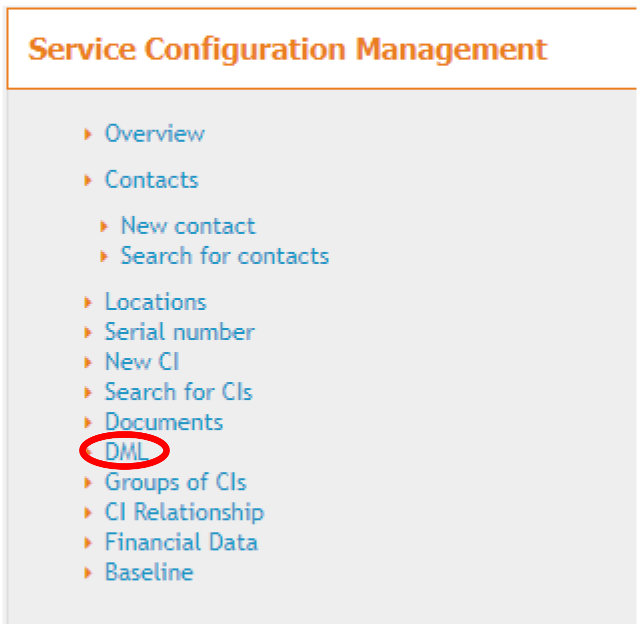
The media release lifecycle steps are:

- Demand for new service or product arises.
- Decision is made to make or buy the product (service, build or application) based on functional requirements extracted from the requirements traceability tool. Product is created or selected from the service/ product catalogue in accordance with architectural design policies (Service Design). COTS product is procured and stored in the DML with asset status 'procured'. If new, the product is added to the Approved Products Catalogue. In-house created application source code is managed directly in the software configuration management repository.
- If COTS product or gold build is being packaged, media is extracted from the DML.
- Product is packaged or developed and packaged (in which case add-on functionality is treated in the same way as in-house applications and builds).
- Stub records or original baselines are created in the software configuration management tool.
- Development code revisions and package revisions are recorded in the software configuration management tool throughout development.
- Unit testing is carried out.
- Packaging is completed to create the release package.
- Product package is quality assured (inc testing, staging and any rework).
- Completed media package (build, service or application) is lodged back in the DML as authorised media ready for deployment.
- Following Change Management approval, product is released to the estate via the appropriate distribution system with logical installations being recorded via due process in the CMS (CMDB).

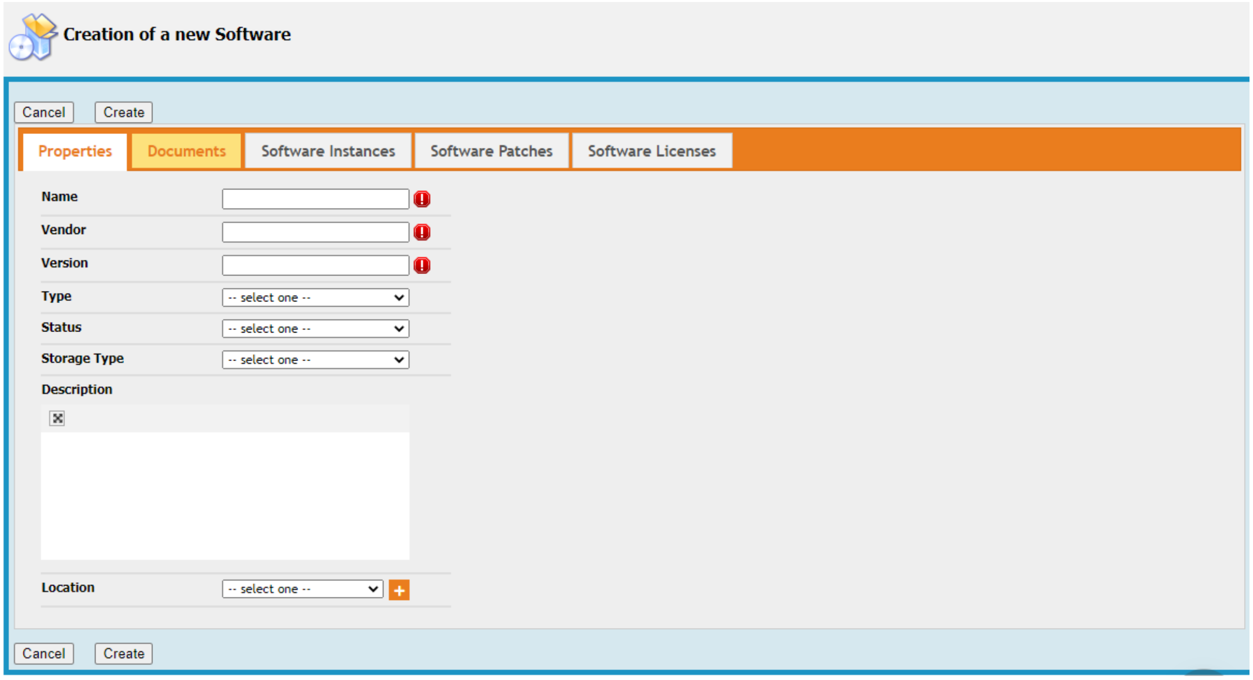
- DML entities are archived as soon as:
    - CMS or CMDB indicates that packaged release is no longer in use at any location (a period of grace is required following the last decommission or upgrade to allow for any necessary regressions) and
    - The DML entity has been removed from the technical or user (service) catalogue as a selectable item

3.8. Creation of DML

- Click on the Menu Option "Service Configuration Management:



- Then, click on DML as shown in the above figure.
- Click on "Create a New Software".
- The below dialog box will appear:



Then click on Create.

4. Service Request Management

There are two alternatives for managing user requests in tbITSM v.1.0. You can choose to install one of the two following modules:

- **Simple Ticket Management**
- **User Request Management ITIL v4**

The **Simple Ticket Management** module provides a simplified ticketing system. It is used to keep track of end-users' requests. There are two types of requests:

- **Incidents** are used to track unexpected issues that have an impact on the delivered services.
- **Service requests** are used to request new services or features like installing a new PC, creating a new email address.

This module manages both types of requests in a single type of ticket. Incidents and service requests will follow the same workflow. This allows agent to easily manage any kind of ticket and reclassify a request without having to create a new one.

The **User Request ITIL v4** module focuses on service requests.

Whatever module you choose, a *user request* can be created via the customer portal or directly in tbITSM v.1.0. The support agent can then modify and communicate with the customer via a journal called "Public log." He can also communicate with internal teams within his company through a journal called "Private log".

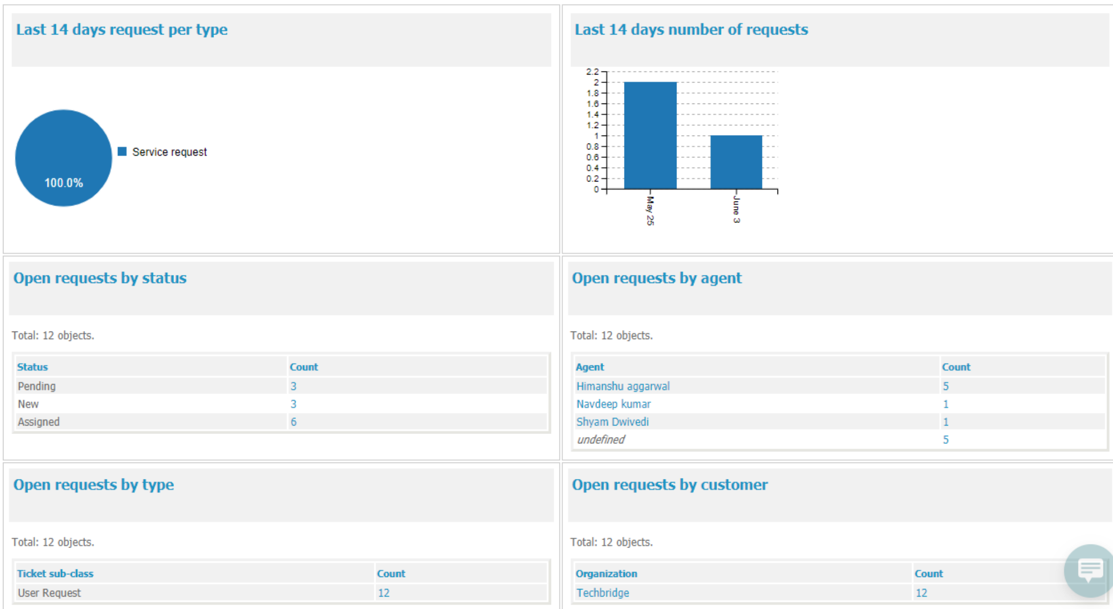
A customer user will see only the public log. The private log cannot be viewed from the portal.

A user request is controlled by a workflow in order to make sure it is managed according to a defined process. Only authorized users can manage a user request and change its status.

A user request can be linked to a parent problem, or a parent change. In case you have installed the User Request ITIL V4 V3 module, your request can be linked to a parent incident.

It is also possible to regroup user requests under a single user request.

The overview dashboard allows agents and managers to monitor the helpdesk activity.



4.1. User Request

User request are used to document all the request submitted by users.

User Request Properties

Name	Type	Mandatory?
Organization	Foreign key to a(n) Organization	Yes
Caller	Foreign key to a(n) Person	Yes
Status	Possible values: Approved, Assigned, Closed, Escalated TTO, Escalated TTR, New, Pending, Rejected, Resolved, Waiting for approval	Yes
Origin	Possible values: mail, monitoring, phone, portal	No
Title	Alphanumeric string	Yes
Description	Multiline character string	Yes
Service	Foreign key to a(n) Service	No
Service subcategory	Foreign key to a(n) Service Subcategory	No
Hot Flag	Possible values: No, Yes	No
Hot reason	Alphanumeric string	No
Pending reason	Multiline character string	No
Request Type	Possible values: Incident, Service request	No
Impact	Possible values: A department, A service, A person	Yes
Urgency	Possible values: critical, high, medium, low	Yes
Priority	Possible values: critical, high, medium, low	Yes
Team	Foreign key to a(n) Team	No
Agent	Foreign key to a(n) Person	No

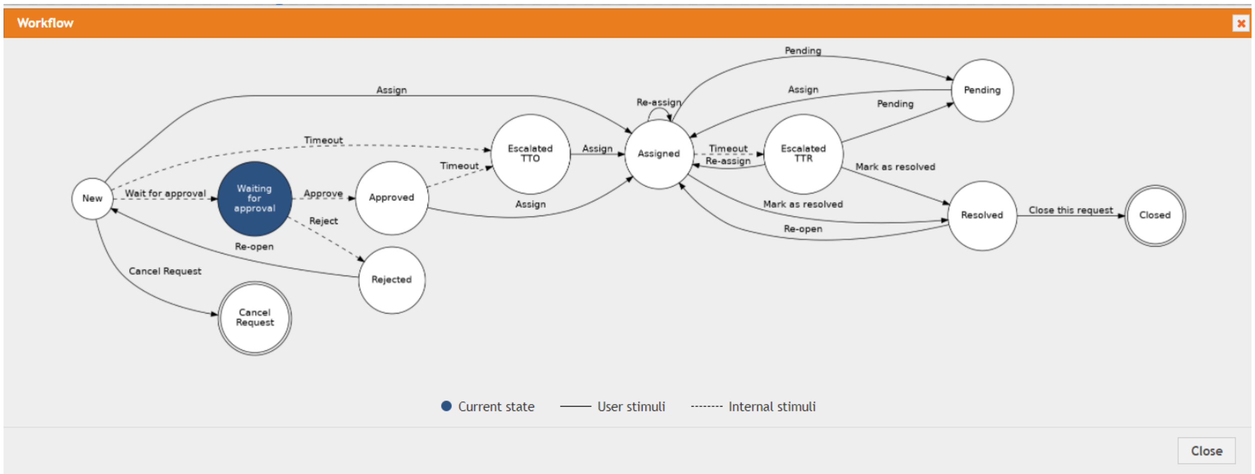
Approver	Foreign key to a(n) Person	No
Start date	Date and time (year-month-day hh: mm: ss)	No
Last update	Date and time (year-month-day hh: mm: ss)	No
Assignment date	Date and time (year-month-day hh: mm: ss)	No
TTO Deadline	Core: AttributeStopWatch+ (100_deadline)	No
TTR Deadline	Core: AttributeStopWatch+ (100_deadline)	No
Last pending date	Date and time (year-month-day hh: mm: ss)	No
Resolution date	Date and time (year-month-day hh: mm: ss)	No
Close date	Date and time (year-month-day hh: mm: ss)	No
Parent request	Foreign key to a(n) User Request	No
Parent problem	Foreign key to a(n) Problem	No
Parent change	Foreign key to a(n) Change	No
Resolution code	Possible values: assistance, bug fixed, hardware repair, other, software patch, system update, training	No
Solution	Multiline character string	No
Resolution delay	Core: AttributeDuration+	No
User satisfaction	Possible values: Very satisfied, Fairly satisfied, Rather Dissatisfied, Very Dissatisfied	No
User comment	Multiline character string	No
SLA tto passed	Core: AttributeStopWatch+ (100_passed)	No
SLA tto over	Core: AttributeStopWatch+ (100_overrun)	No
SLA ttr passed	Core: AttributeStopWatch+ (100_passed)	No
SLA ttr over	Core: Attribute Stopwatch+ (100_overrun)	No

Tags

Tab	Description
CIs	All the configuration items impacted for this ticket
Contacts	All the contacts linked to this ticket
Child Requests	All the requests that are linked to this parent request
Work orders	All the work orders for this ticket

4.2. Workflow of Service Request

The following diagram explains the Service Request Management Life cycle:

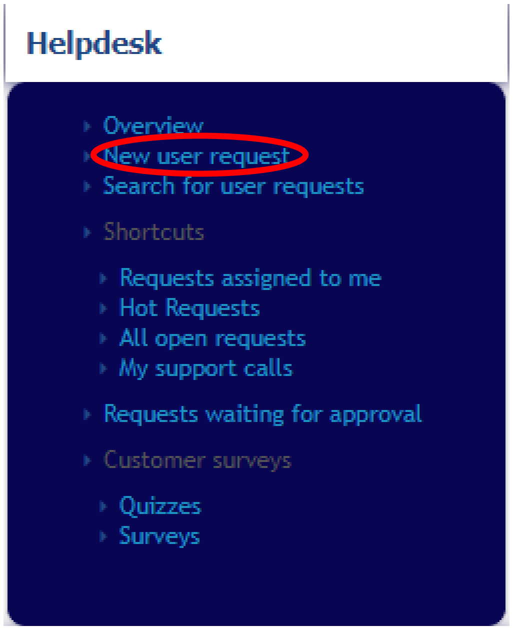


The following section explains the status of a Service Request:

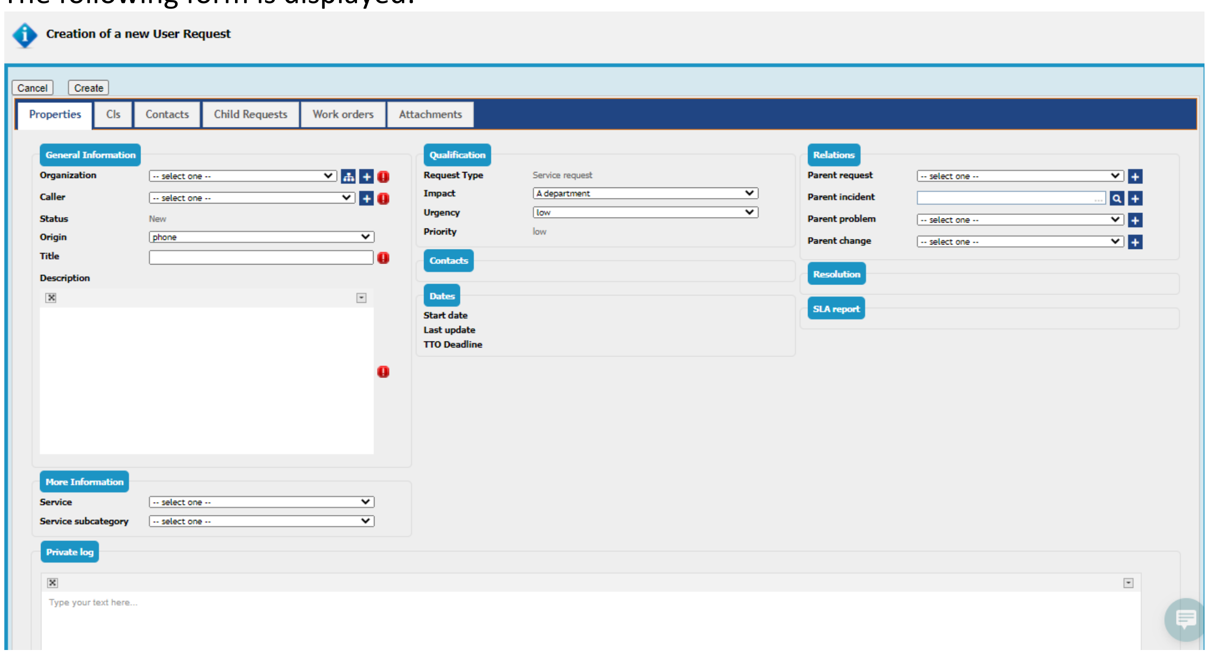
- **New**: The status *New* cannot be selected by the users, it is set by the application after a Service Request is created.
- **Assigned**: After the SR is created, it is assigned to a Workgroup, based on the selected Catalog. The SR is then assigned to the appropriate Analyst. The status of the SR changes to *Assigned*.
- **In Progress**: When the SR is assigned to an Analyst, the status of the Service Request is changed to *In Progress*. The Analyst can refer to various Knowledge Articles related to the Service catalog to work on the SR.
- **Pending**: If the Analyst cannot continue working on the SR due to dependency on any information from the End User or any other activity to complete, the status of the SR is moved to *Pending*.
- **Resolved**: After the SR is *In Progress*, the Analyst should resolve the SR within the provided Service Window. After an SR is resolved, the status of the Service Request is changed to *Resolved*. The End User can reopen the SR if the resolution is not satisfactory.
- **Closed**: After an SR is resolved, the status of the SR can be changed to *Closed* based on the configuration (manual closure or auto closure).

4.3. Creating a User Request

Click on the “New user request” menu:



The following form is displayed:



4.4. Managing Public & Private Log

The public and the private log are used to keep track of all communications and activities related to a user request.

The **public log** is aimed at exchanging information with the requestor.

The **private log** is the preferred way for keeping track of the investigations or operations: copy/paste of command line results, summary of communications with a provider, etc.

Each entry in the public or private log is tracked with the name of the user who updated it and when it was done. It cannot be modified nor deleted.

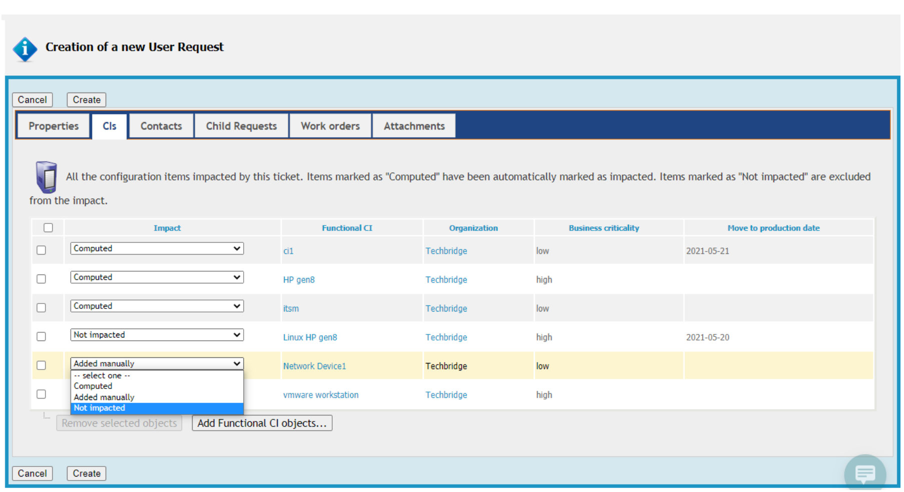
The public log is visible from the customer portal.

4.5. Managing impacted CIs and Contacts

When a user request is edited, the agent can specify which configuration items (CIs) or contacts are related to this request via the tabs "CIs" and "Contacts". The objects manually added to the ticket are to be flagged as "Added manually" (which is the default).

When saving the ticket, the impact analysis engine automatically adds to this list all the other CIs (and Contacts) impacted by the original CIs. The "impact" computation is based on rules defined in the tbITSM v.1.0. data model. The additional objects resulting from this computation are also linked to the ticket and flagged as "Computed" (to differentiate them from the original objects marked as "Added manually").

The impact rules of each type of CI are described in the Configuration management module.



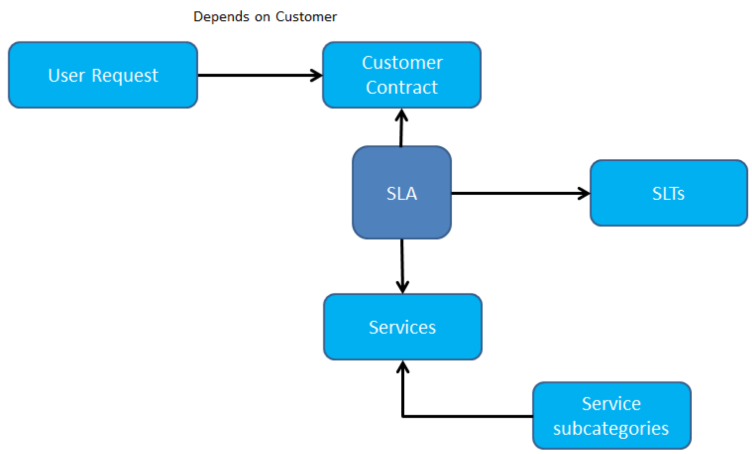
4.6. Dependencies with the Service Catalog

The Helpdesk module is linked to the service catalog in order to:

- define which service and service subcategories can be selected for a given customer
- define to which team a user request can be assigned
- compute time to own (TTO) and time to resolve (TTR) deadlines

The list of services displays only the service that have been purchased by the selected customer via a *customer contract*. The list of *service subcategories* displays only the sub categories corresponding to the selected service and the selected request type.

The following picture describes the relationships between the service catalog elements and user requests.

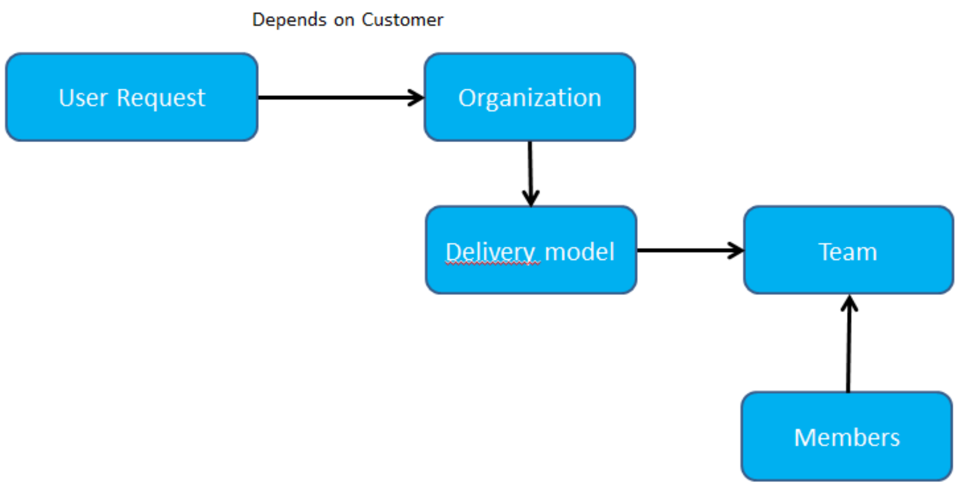


Only Services with a defined Service Family are proposed in the Enhanced Portal.

4.7. Assigning a User request to a Team and Agent

The list of teams to which you can assign a user request is defined by the *delivery model* of the corresponding customer. When creating a user request, the agent has to select the customer organization, then the list of teams is strictly limited to the teams defined for this customer. If a team is missing, the delivery model of the customer must be updated to reflect this need.

The following picture describes the relationships between the delivery model and user requests.



4.8. Automated Priority Computation

The priority is computed automatically. This computation depends on the impact and the urgency of the user request. The following matrix describes how the priority is computed:

Urgency / Impact	Department	Service	Person
critical	Critical	critical	high
high	Critical	high	medium
medium	High	medium	medium
low	Low	low	low

4.9. Deadline Computation

To meet service agreements with customers, tbITSM v.1.0. automatically computes time to own (TTO) and time to resolve (TTR) deadlines. These deadlines depend on the *service level agreements* defined in the customer contracts. In the basic version of tbITSM v.1.0. there is no coverage window. The calculations assume a 24*7 service coverage.

The measured TTO is the time cumulated while the user request is not assigned. When the TTO deadline is passed, the ticket status is automatically changed to "Escalated TTO".

The measured TTR is the time cumulated while the user request is neither pending nor resolved. When the TTR deadline is passed, the ticket status is automatically changed to "Escalated TTR".

The computation of the deadlines depends on:

- The *service level agreement* defined in the customer contract for the selected service
- The *priority* of the user request
- The *type* of request

These are defined in the service level targets (SLT) corresponding to the service level agreement (SLA).

The deadlines computation is performed each time a modification is made on the user request.

When the cumulated TTO/TTR reaches 75% of the TTO/TTR deadline, then the user request is displayed in yellow. Once the deadline is passed, it becomes red.

Once the user request is resolved, deadlines and measures are kept within the user request. This can be used both for analyzing process issues and for reporting purposes.

The following information are recorded:

- TTO deadline (date and time)
- TTO cumulated (seconds)
- TTO passed (yes / no)
- TTO overrun (seconds)
- TTR deadline (date and time)
- TTR cumulated (seconds)
- TTR passed (yes / no)
- TTR overrun (seconds)

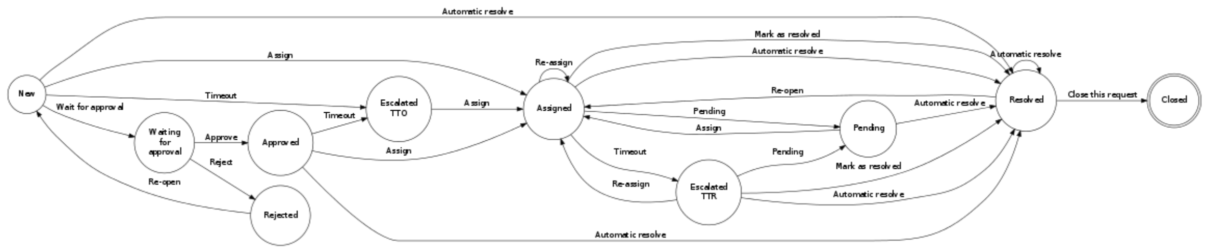
4.10. Regrouping User Request

It is sometimes useful to regroup user requests under an incident which is the root cause of the issue. For instance, when a mail server is down, you may have several end users complaining about mailbox being unavailable.

To group user requests, use the field **parent request**.

If an incident ticket is parent of a user request, then each time its private and public logs are modified, tbITSM v.1.0. will automatically update the logs of the child requests. When the parent incident gets resolved, tbITSM v.1.0. will automatically mark the child requests as “resolved”.

4.11. User Request Life Cycle



	New	Assigned	Escalated TTO	Waiting for approval	Resolved	Pending	Escalated TTR	Closed	Approved	Rejected
Organization	M	M	M	M	R/O	M	M	R/O	M	M
Caller	M		M	M	R/O			R/O	M	M
Status	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
Origin					R/O			R/O		
Title					R/O			R/O		
Description					R/O			R/O		
Service					M			R/O		
Service subcategory								R/O		
Hot Flag	H		H	H	R/O			R/O	H	H
Hot reason	H		H	H	R/O			R/O	H	H
Pending reason	H	H	H	H	R/O	M	H	R/O	H	H
Request Type		M			R/O	M	M	R/O		
Impact					R/O			R/O		
Urgency					R/O			R/O		
Priority	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
Team	H	M			R/O	M	M	R/O		
Agent	H	M	H	H	R/O	M	M	R/O	H	H
Approver	H	R/O	H		R/O	R/O	R/O	R/O	R/O	H
Start date	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
Last update	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
Assignment date	H	R/O	H	H	R/O	R/O	R/O	R/O	H	H
TTO Deadline	R/O	H	R/O	R/O	H	H	H	H	R/O	H
TTR Deadline	H	R/O	H	H	H	H	R/O	H	H	H
Last pending date	H	H	H	H	H	R/O	H	H	H	H
Resolution date	H	H	H	H	R/O	H	H	R/O	H	H
Close date	H	H	H	H	H	H	H	R/O	H	H
Parent					R/O			R/O		

request										
Parent problem					R/O			R/O		
Parent change					R/O			R/O		
Resolution code	H	H	H	H	M	H	H	R/O	H	H
Solution	H	H	H	H	M	H	H	R/O	H	H
Resolution delay	H	H	H	H	R/O	H	H	R/O	H	H
User satisfaction	H	H	H	H	H	H	H		H	H
User comment	H	H	H	H	H	H	H		H	H
SLA tto passed	H	R/O	H	H	R/O	R/O	R/O	R/O	H	H
SLA tto over	H	R/O	H	H	R/O	R/O	R/O	R/O	H	H
SLA ttr passed	H	H	H	H	R/O	H	H	R/O	H	H
SLA ttr over	H	H	H	H	R/O	H	H	R/O	H	H

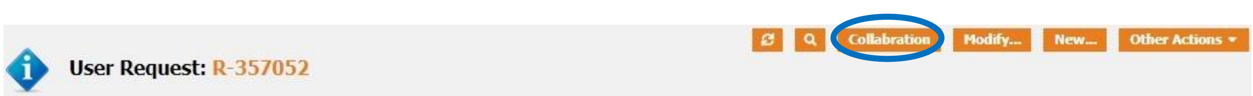
Table key:

- H: hidden
- R/O: read-only
- M: mandatory

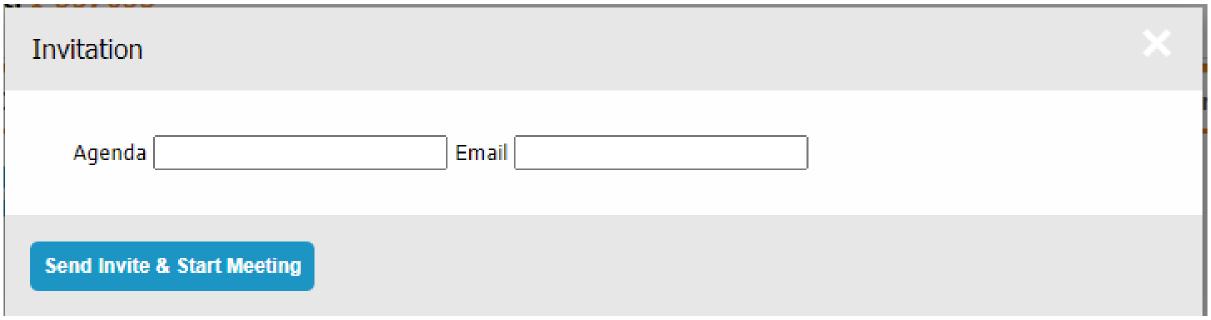
4.12. Collaboration and Communication

The tool is integrated with communication and collaboration tool to facilitate coordination and cooperation with onsite/remote users. It also helps in diagnosis and swarming.

- Click on Collaboration option.



Then, the screen displays are:



- Then, fill the Agenda and mark all the email ids with whom communication is needed.

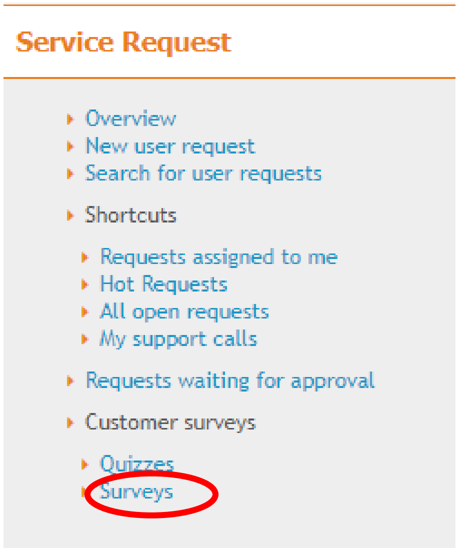
- Then click on “Send Invite & Start Meeting”.

The screen displayed is

This is the interface where users can communicate and collaborate easily and efficiently.

4.13. Automatic Survey

Click on the Survey



Then the display will be shown as below, click on New.

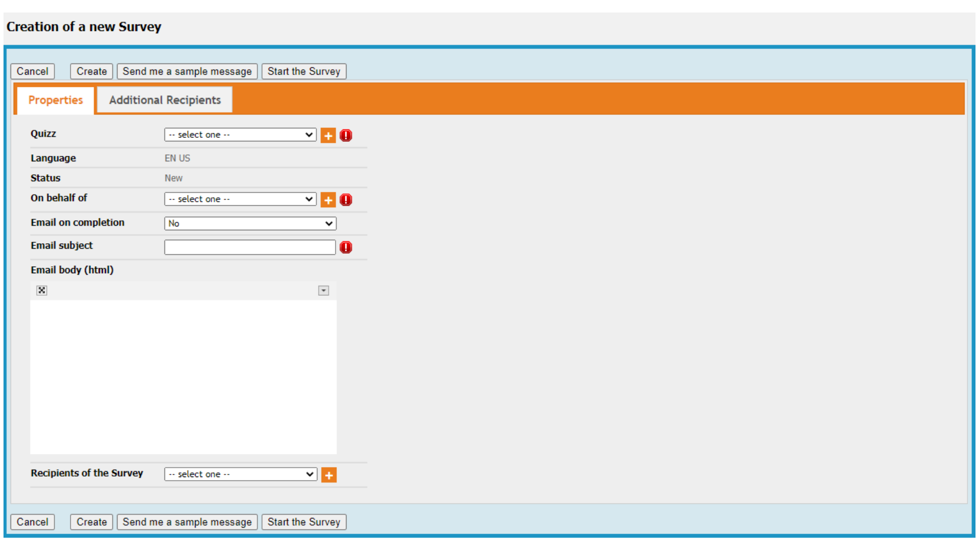

Surveys

Total: 9 objects.

New... Other Actions

Survey	Status	Sent	Language
Incident Closed Review	New		EN US
Incident Closed Review	New		EN US
Incident Closed Review	New		EN US
Incident Closed Review	New		EN US
Incident Closed Review	New		EN US
Incident Closed Review 2021-06-29 15:09:44	Ongoing	2021-06-29 15:09:44	EN US
Incident Closed Review 2021-06-29 15:11:49	Closed	2021-06-29 15:11:49	EN US
ServiceRequest Closed Feedback	New		EN US
ServiceRequest Closed Feedback 2021-06-29 15:30:12	Closed	2021-06-29 15:30:12	EN US

The display is shown as:



Fill out all the fields and then click on Start the survey.

4.14. Fetching of Adhoc Report

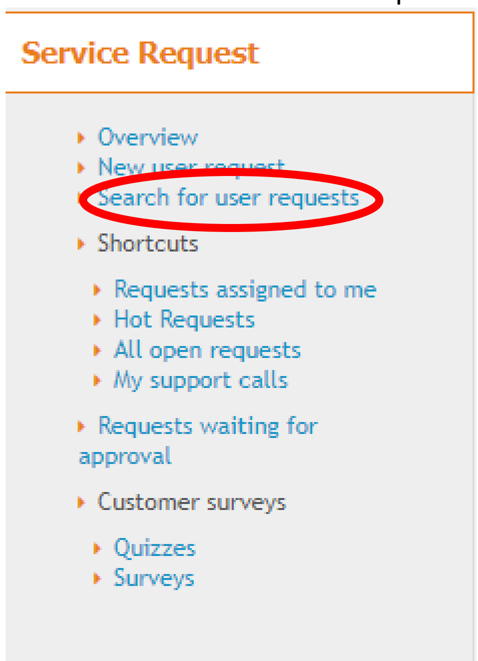
Adhoc report may have tabular data arrangement, a matrix as well as a chart. On Adhoc Wizard, you can also:

- Sort the data to re-arrange it in order of your choice
- Get totals / summaries
- Highlights selective records (when a condition is met)

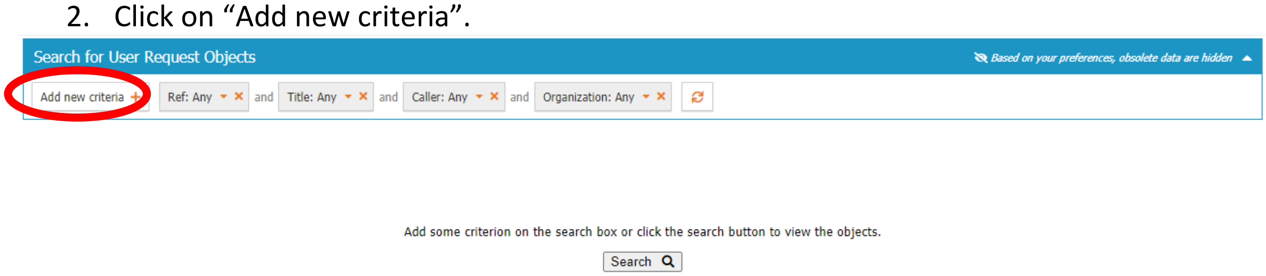
When you run an ad hoc report in output format, the viewer provides Power Viewer that helps altering the report on the viewer itself.

Steps for fetching Adhoc Report: -

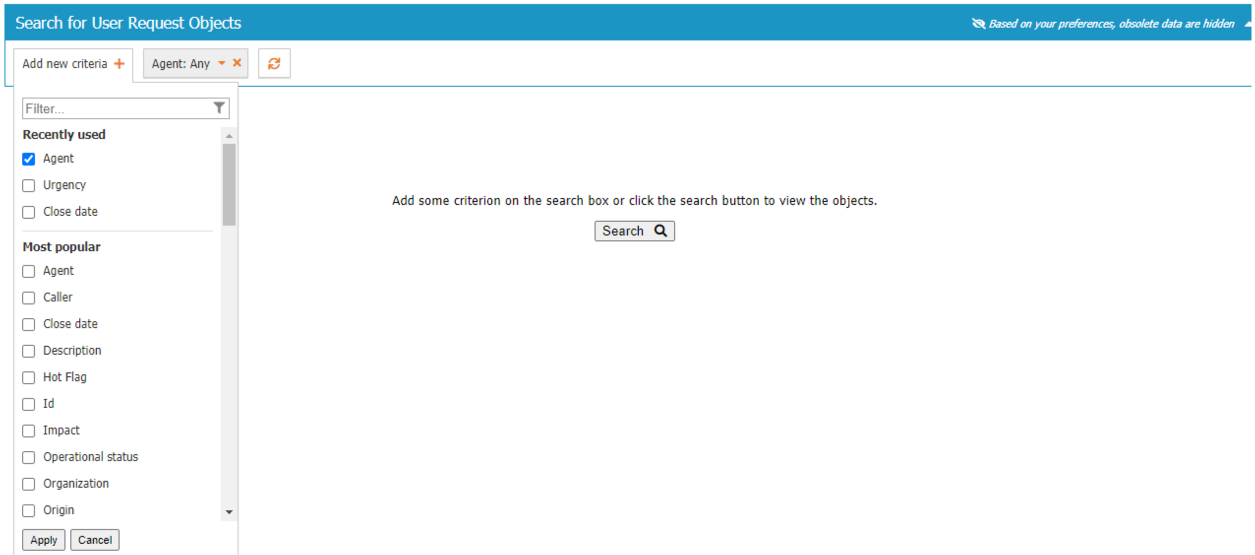
1. Click on Search for user requests.



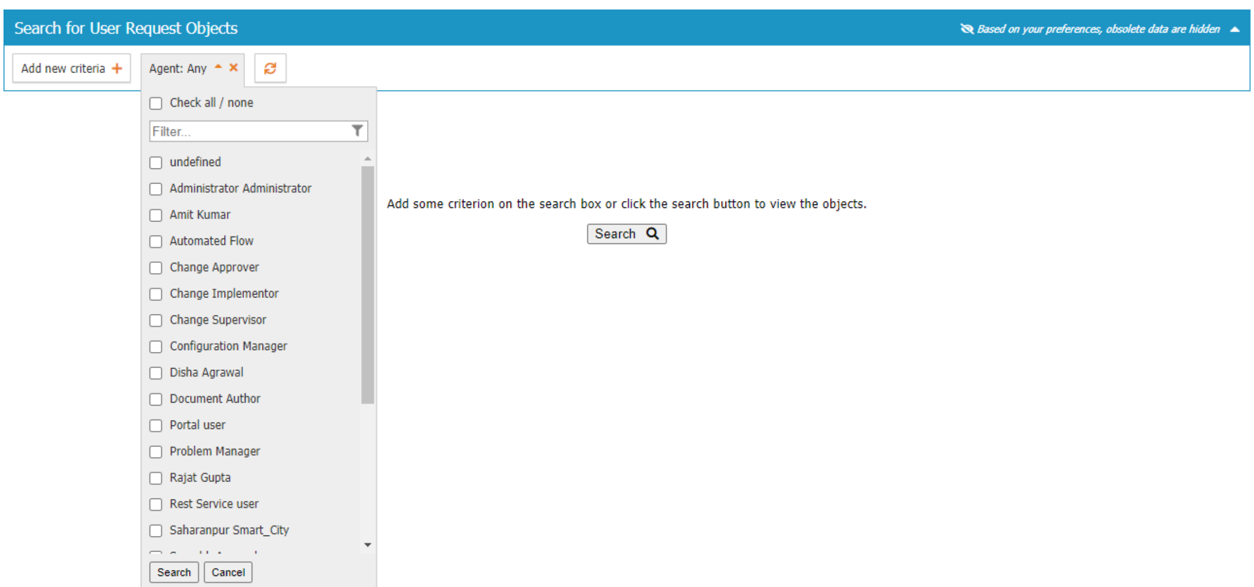
2. Click on "Add new criteria".



3. Select the Criteria.

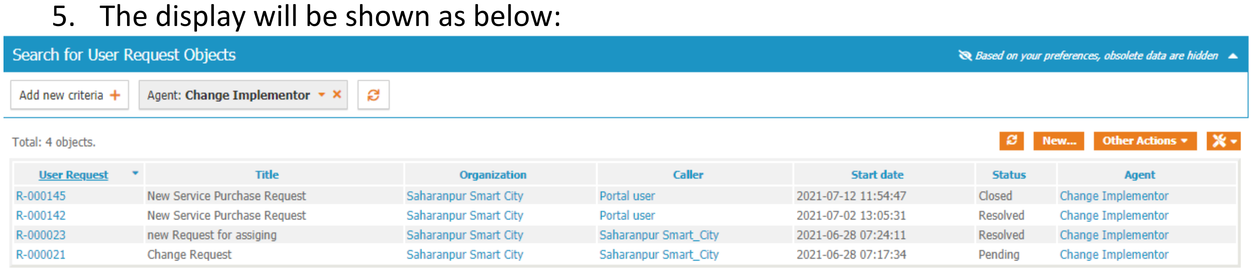


For example, if you select Agent, then you will get the display as:



4. Then select the values and then click on Search.

5. The display will be shown as below:



5. Incident Management

5.1. Incident

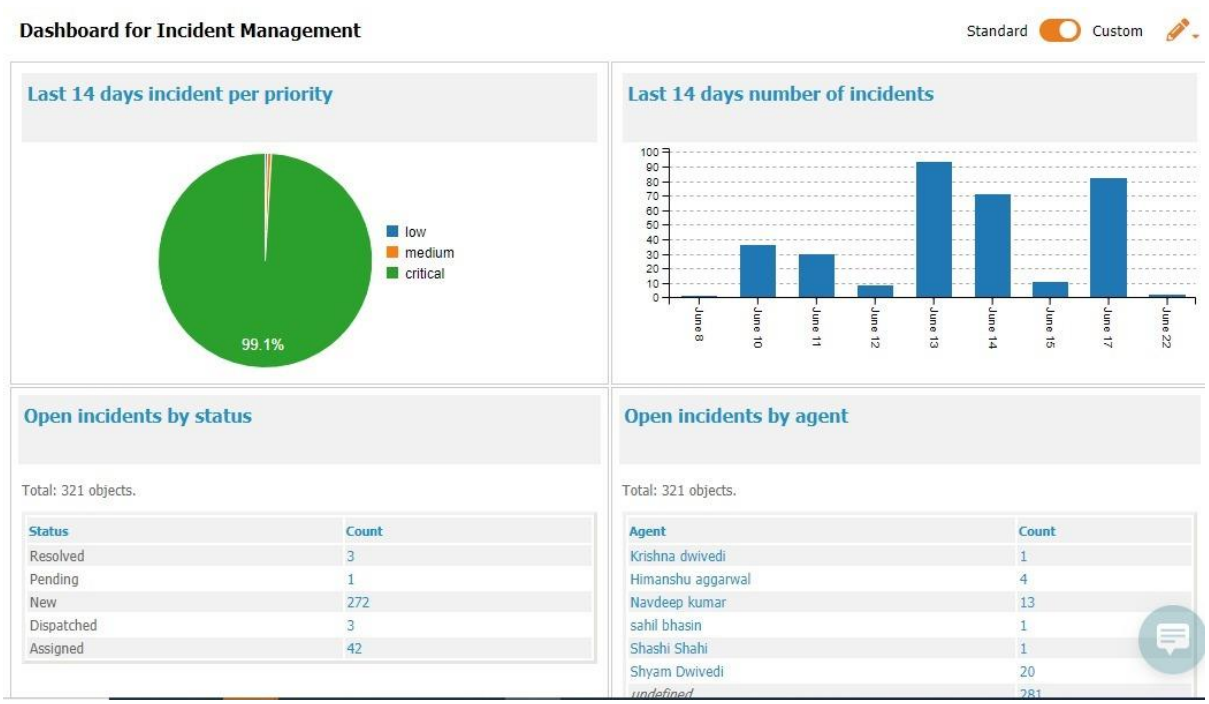
An *incident ticket* or simply an "*incident*" keeps tracks of a technical issue within the IT:

- System down
- Network issue
- Application failure

An *incident* can be linked to a *problem* (ticket). For instance, when the same *incident* is occurring often and you would like to investigate the root cause of the *problem*. An *incident* can be linked to a *change* (ticket), if it requires the implementation of a modification. For instance, when installing a patch.

Incident tickets are managed by people having the profile **Support agent**.

Dashboard:



5.1.1. Incident Properties

Name	Type	Mandatory?
Organization	Foreign key to a(n) Organization	Yes
Caller	Foreign key to a(n) Person	Yes
Status	Possible values: Assigned, Closed, Escalated TTO, Escalated TTR, New, Pending, Resolved	Yes
Origin	Possible values: mail, monitoring, phone, portal	No
Title	Alphanumeric string	Yes
Description	Multiline character string	Yes
Service	Foreign key to a(n) Service	No
Service subcategory	Foreign key to a(n) Service Subcategory	No
Hot Flag	Possible values: No, Yes	No
Hot reason	Alphanumeric string	No
Pending reason	Multiline character string	No
Impact	Possible values: A department, A service, A person	Yes
Urgency	Possible values: critical, high, medium, low	Yes
Priority	Possible values: critical, high, medium, low	Yes
Team	Foreign key to a(n) Team	No
Agent	Foreign key to a(n) Person	No
Start date	Date and time (year-month-day hh: mm: ss)	No
Last update	Date and time (year-month-day hh: mm: ss)	No
Assignment date	Date and time (year-month-day hh: mm: ss)	No
TTO Deadline	Core: Attribute Stopwatch+ (100_deadline)	No
TTR Deadline	Core: Attribute Stopwatch+ (100_deadline)	No
Last pending date	Date and time (year-month-day hh: mm: ss)	No
Resolution date	Date and time (year-month-day hh: mm: ss)	No
Close date	Date and time (year-month-day hh: mm: ss)	No
Parent incident	Foreign key to a(n) Incident	No
Parent Problem ID	Foreign key to a(n) Problem	No
Parent change	Foreign key to a(n) Change	No
Resolution code	Possible values: assistance, bug fixed, hardware repair, other, software patch, system update, training	No
Solution	Multiline character string	No
Resolution delay	Core: Attribute Duration+	No
User satisfaction	Possible values: Very satisfied, Fairly satisfied, Rather Dissatisfied, Very Dissatisfied	No
User comment	Multiline character string	No
SLA TTO passed	Core: Attribute Stopwatch+ (100_passed)	No
SLA TTO over	Core: Attribute Stopwatch+ (100_overrun)	No
SLA TTR passed	Core: Attribute Stopwatch+ (100_passed)	No
SLA TTR over	Core: Attribute Stopwatch+ (100_overrun)	No

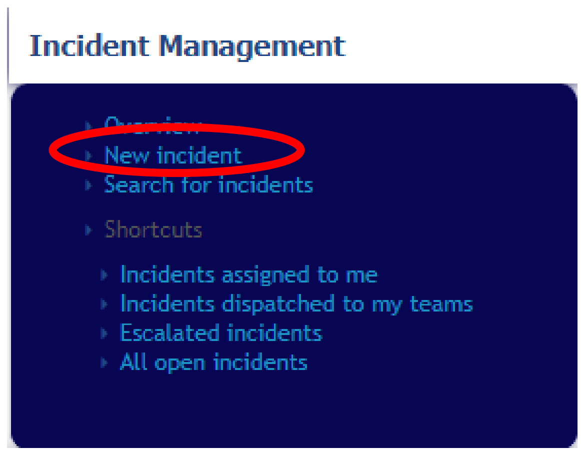
Tabs

Tab	Description
Cis	All the configuration items impacted for this ticket
Contacts	All the contacts linked to this ticket
Child incidents	All the child incidents related to this incident
Related Request List	
Work orders	All the work orders for this ticket

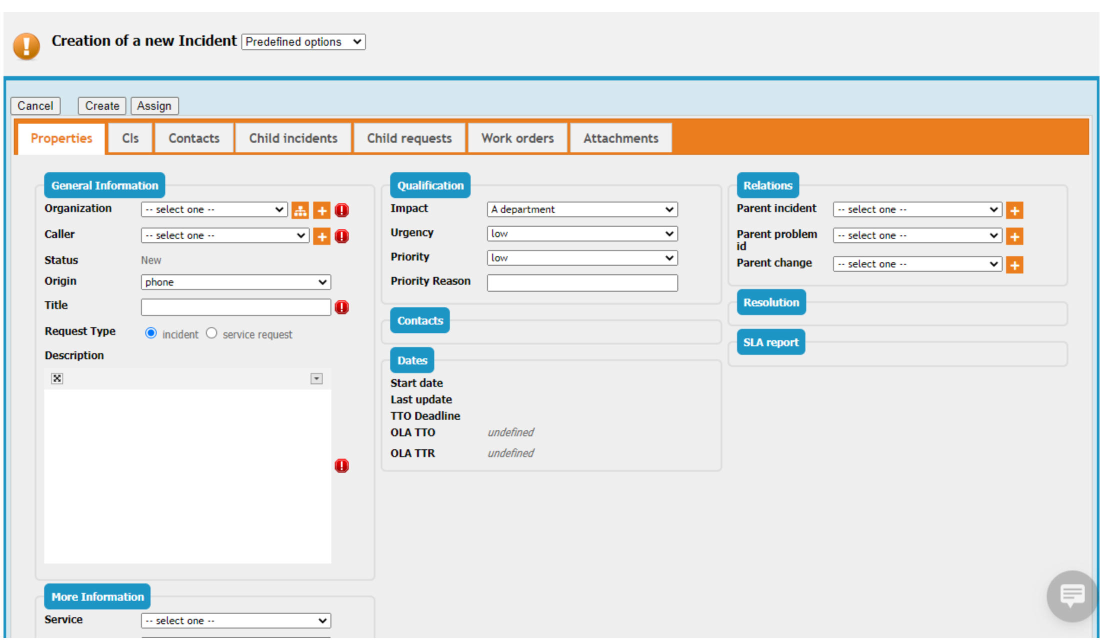
5.2. Creating an Incident

5.2.1. Through Web Portal (Manual)

Click on the “New incident” menu:

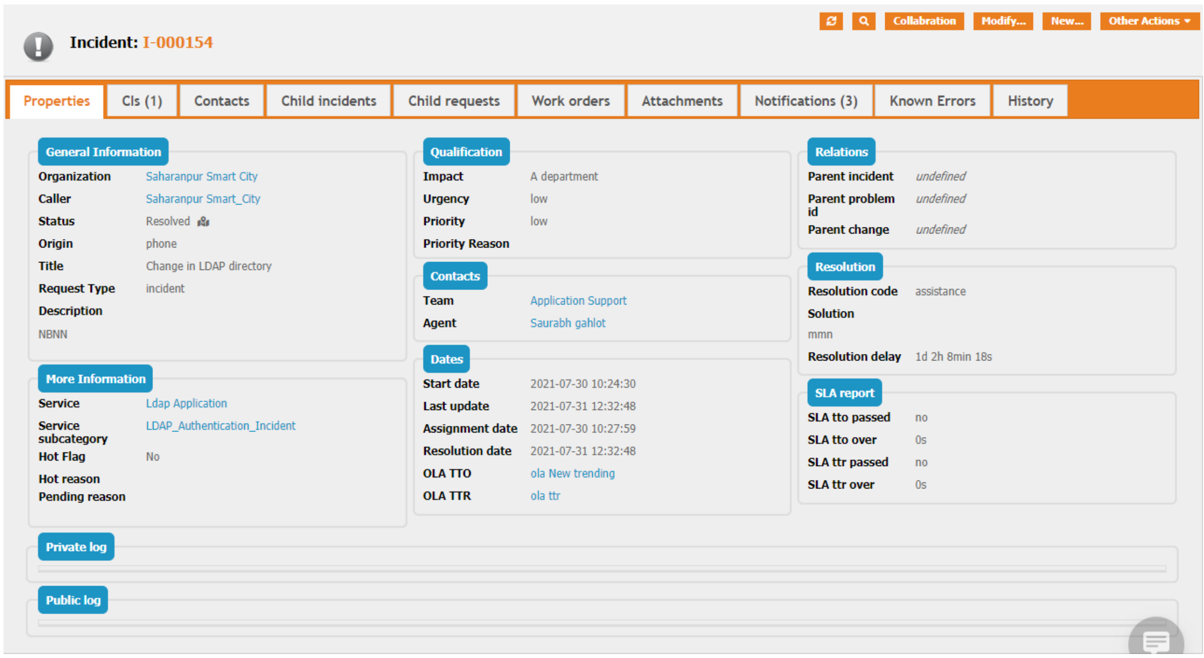


The following form is displayed:



5.2.2. Through Monitoring and Event Correlation Engine

- Once the service goes down, tbXMS will generate the event.
- All the events will be correlated and will convert into Alarm.
- All the critical alarms in tbXMS will convert into the Incidents
- Incident will be visible in the tbITSM v.1.0. portal.



5.2.3. Email to Incident

Please refer Admin Guide v.1.0., Section 21 (Page No. 277)

5.2.4. Through Customer Self Service Portal

Please refer User Guide of tbITSM v.1.0.

5.3. Managing Public & Private Log

The public and the private log are used to keep track of all communications and activities related to a user request.

The **public log** is aimed at exchanging information with the requestor.

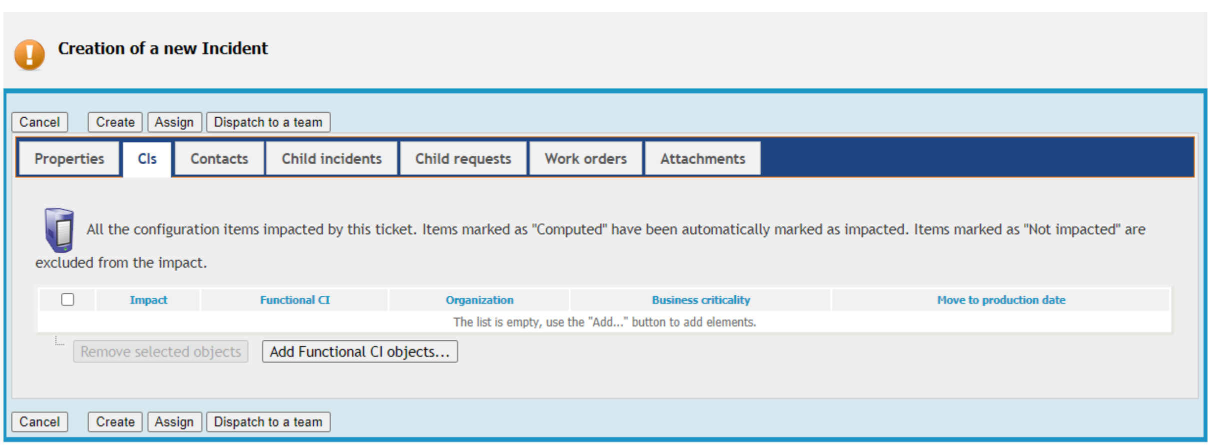
The **private log** is the preferred way for keeping track of the investigations or operations: copy/paste of command line results, summary of communications with a provider, etc.

Each entry in the public or private log is tracked with the name of the user who updated it and when it was done. It cannot be modified nor deleted.
The public log is visible from the customer portal.

5.4. Managing Impacted CIs and Contacts

When a user request is edited, the agent can specify which configuration items (CIs) or contacts are related to this request via the tabs “CIs” and “Contacts”. The objects manually added to the ticket are to be flagged as “Added manually” (which is the default).

When saving the ticket, the impact analysis engine automatically adds to this list all the other CIs (and Contacts) impacted by the original CIs. The “impact” computation is based on rules defined in the tbITSM v.1.0. data model. The additional objects resulting from this computation are also linked to the ticket and flagged as “Computed” (to differentiate them from the original objects marked as “Added manually”).



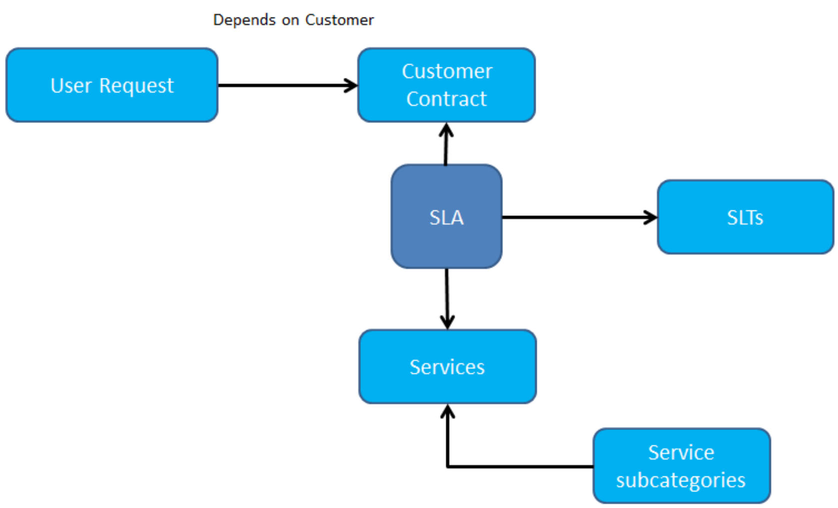
5.5. Dependencies with the Service Catalog

The Helpdesk module is linked to the service catalog in order to:

- define which service and service subcategories can be selected for a given customer
- define to which team a user request can be assigned
- compute time to own (TTO) and time to resolve (TTR) deadlines

The list of services displays only the service that have been purchased by the selected customer via a *customer contract*. The list of *service subcategories* displays only the sub categories corresponding to the selected service and the selected request type.

The following picture describes the relationships between the service catalog elements and user requests.

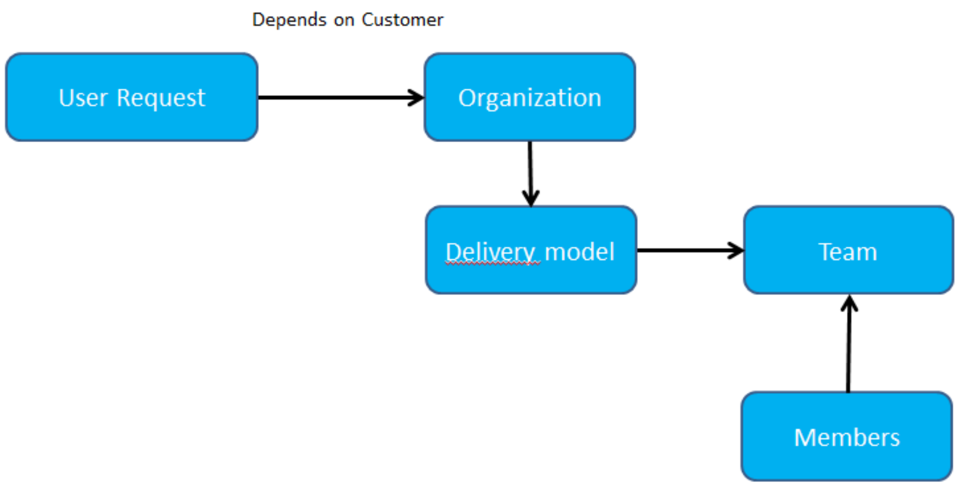


Only Services with a defined Service Family are proposed in the Enhanced Portal.

5.6. Assigning a User Request to a Team and Agent

The list of teams to which you can assign a user request is defined by the *delivery model* of the corresponding customer. When creating a user request, the agent has to select the customer organization, then the list of teams is strictly limited to the teams defined for this customer. If a team is missing, the delivery model of the customer must be updated to reflect this need.

The following picture describes the relationships between the delivery model and user requests.



5.7. Automated Priority Computation

The priority is computed automatically. This computation depends on the impact and the urgency of the user request. The following matrix describes how the priority is computed:

Urgency / Impact	Department	Service	Person
Critical	critical	critical	high
High	critical	high	medium
Medium	high	medium	medium
Low	low	low	low

5.8. Deadline Computation

To meet service agreements with customers, tbITSM v.1.0. automatically computes time to own (TTO) and time to resolve (TTR) deadlines. These deadlines depend on the *service level agreements* defined in the customer contracts. In the basic version of tbITSM v.1.0. there is no coverage window. The calculations assume a 24*7 service coverage.

The measured TTO is the time cumulated while the user request is not assigned. When the TTO deadline is passed, the ticket status is automatically changed to “Escalated TTO”.

The measured TTR is the time cumulated while the user request is neither pending nor resolved. When the TTR deadline is passed, the ticket status is automatically changed to “Escalated TTR”.

The computation of the deadlines depends on:

- The *service level agreement* defined in the customer contract for the selected service
- The *priority* of the user request
- The *type* of request

These are defined in the service level targets (SLT) corresponding to the service level agreement (SLA).

The deadlines computation is performed each time a modification is made on the user request.

When the cumulated TTO/TTR reaches 75% of the TTO/TTR deadline, then the user request is displayed in yellow. Once the deadline is passed, it becomes red.

Once the user request is resolved, deadlines and measures are kept within the user request. This can be used both for analyzing process issues and for reporting purposes.

The following information are recorded:

- TTO deadline (date and time)
- TTO cumulated (seconds)
- TTO passed (yes / no)
- TTO overrun (seconds)
- TTR deadline (date and time)
- TTR cumulated (seconds)
- TTR passed (yes / no)
- TTR overrun (seconds)

5.9. Grouping Related Incidents

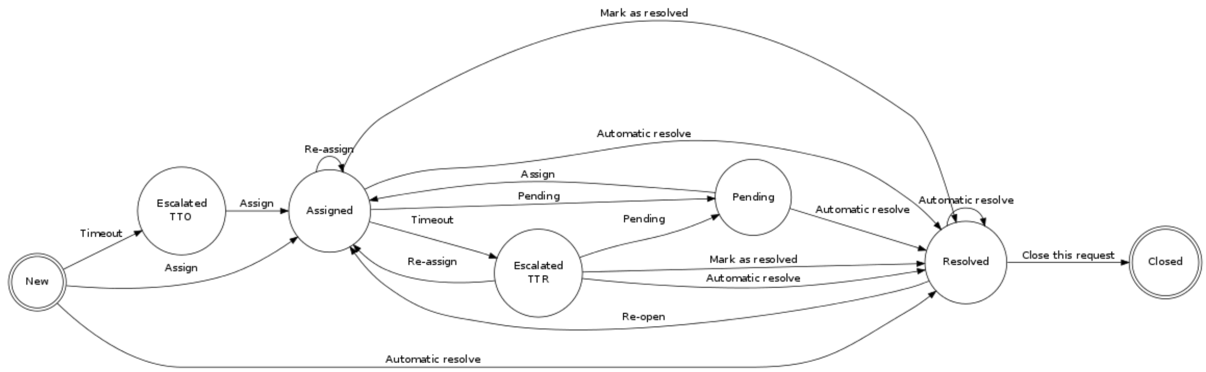
It is sometimes useful to regroup incident tickets under an incident which is the root cause of the issue. For instance, when a network device is down, you may have several servers reported as “not responding”.

To group tickets, use the field **parent incident**.

When an incident is parent of another ticket, each time its private and public logs are modified, tbITSM v.1.0. will automatically update the logs of the child tickets. When the parent incident gets resolved, tbITSM v.1.0. will automatically resolve the child incidents.

5.10. Incident Life Cycle

Incident objects have the following life cycle:



Depending on the status of the object, the constraints on the properties vary as shown on the table below:

	New	Assigned	Escalated TTO	Resolved	Pending	Escalated TTR	Closed
Organization	M	M	M	R/O	M	M	R/O
Caller	M		M	R/O			R/O
Status	R/O	R/O	R/O	R/O	R/O	R/O	R/O
Origin				R/O			R/O
Title				R/O			R/O
Description				R/O			R/O
Service				M			R/O
Service subcategory							R/O
Hot Flag	H		H	R/O			R/O
Hot reason	H		H	R/O			R/O
Pending reason	H	H	H	R/O	M	H	R/O
Impact				R/O			R/O
Urgency				R/O			R/O
Priority	R/O	R/O	R/O	R/O	R/O	R/O	R/O
Team	H	M		R/O	M	M	R/O
Agent	H	M	H	R/O	M	M	R/O
Start date	R/O	R/O	R/O	R/O	R/O	R/O	R/O
Last update	R/O	R/O	R/O	R/O	R/O	R/O	R/O
Assignment date	H	R/O	H	R/O	R/O	R/O	R/O
TTO Deadline	R/O	H	R/O	H	H	H	H
TTR Deadline	H	R/O	H	H	H	R/O	H
Last pending date	H	H	H	H	R/O	H	H
Resolution date	H	H	H	R/O	H	H	R/O

Close date	H	H	H	H	H	H	R/O
Parent incident				R/O			R/O
parent problem id				R/O			R/O
Parent change				R/O			R/O
Resolution code	H	H	H	M	H	H	R/O
Solution	H	H	H	M	H	H	R/O
Resolution delay	H	H	H	R/O	H	H	R/O
User satisfaction	H	H	H	H	H	H	
User comment	H	H	H	H	H	H	
SLA tto passed	H	R/O	H	R/O	H	R/O	R/O
SLA tto over	H	R/O	H	R/O	H	R/O	R/O
SLA ttr passed	H	H	H	R/O	H	H	R/O
SLA ttr over	H	H	H	R/O	H	H	R/O

Table key:

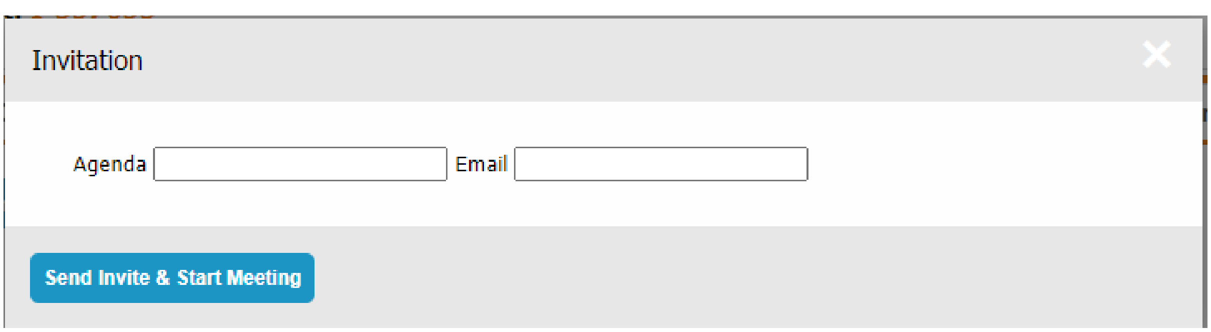
- H: hidden
- R/O: read-only
- M: mandatory

5.11. Collaboration and Communication

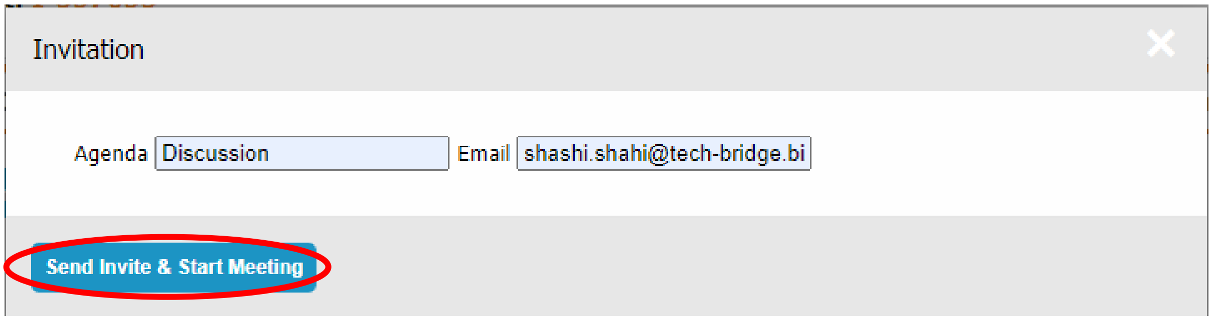
The tool is integrated with communication and collaboration tool to facilitate coordination and cooperation with onsite/remote users. It also helps in diagnosis and swarming.

- Click on Collaboration option.

Then, the screen displays are:



- Then, fill the Agenda and mark all the email ids with whom communication is needed.



- Then click on “Send Invite & Start Meeting”.

The screen displayed is

This is the interface where users can communicate and collaborate easily and efficiently.

5.12. Fetching of Adhoc Report

Adhoc report may have tabular data arrangement, a matrix as well as a chart. On Adhoc Wizard, you can also:

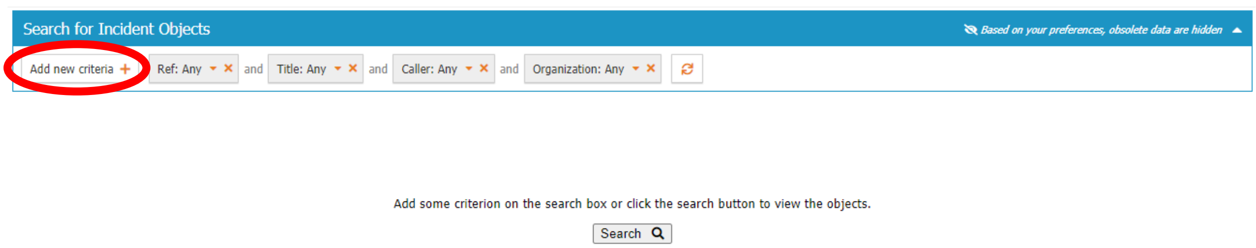
- Sort the data to re-arrange it in order of your choice
- Get totals / summaries
- Highlights selective records (when a condition is met)

When you run an ad hoc report in output format, the viewer provides Power Viewer that helps altering the report on the viewer itself.

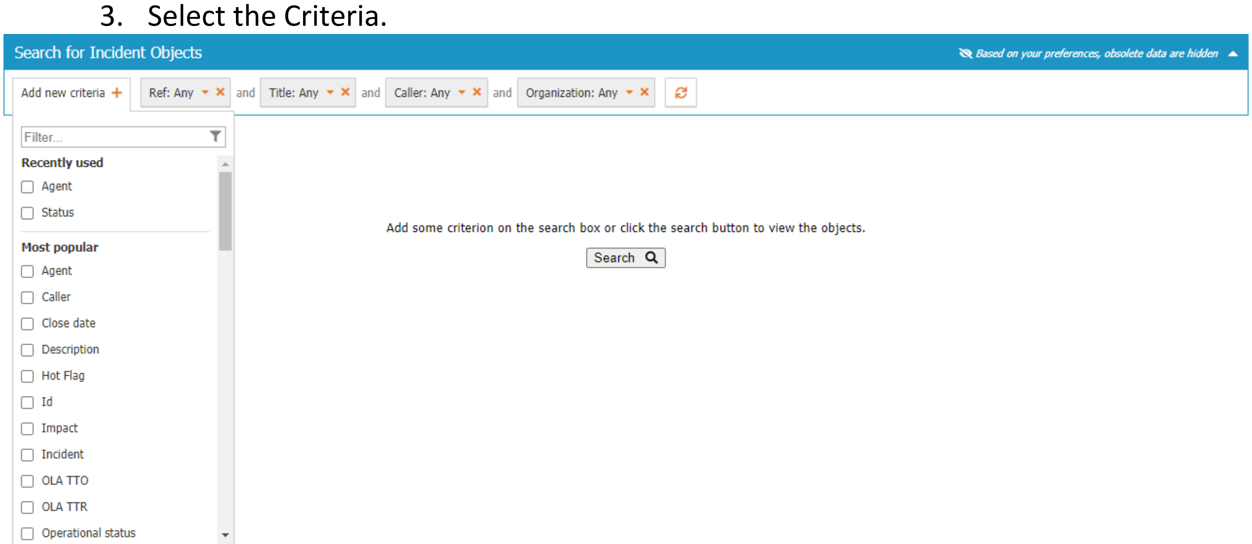
Steps for fetching Adhoc Report: -

1. Click on "Search for incidents".

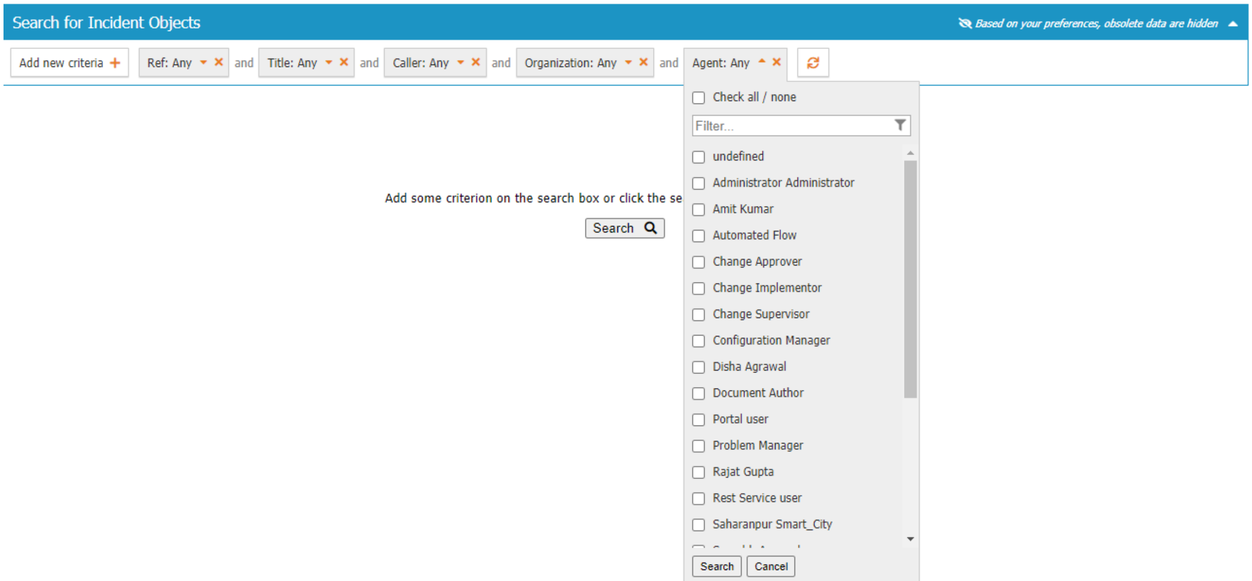
2. Click on "Add new criteria".



3. Select the Criteria.

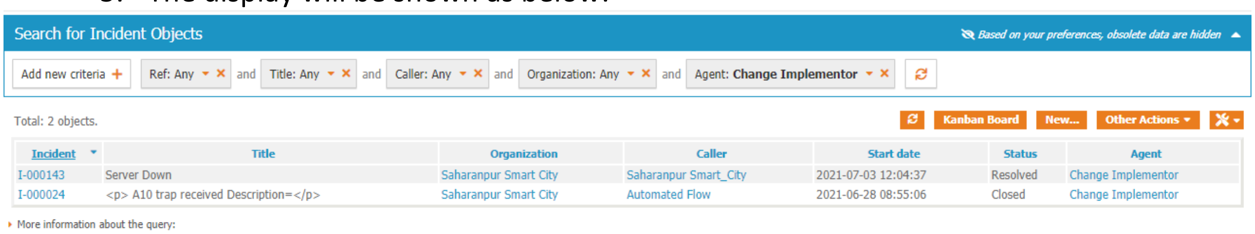


For example, if you select Agent, then you will get the display as:



4. Then select the values and then click on Search.

5. The display will be shown as below:



6. Problem Management Module

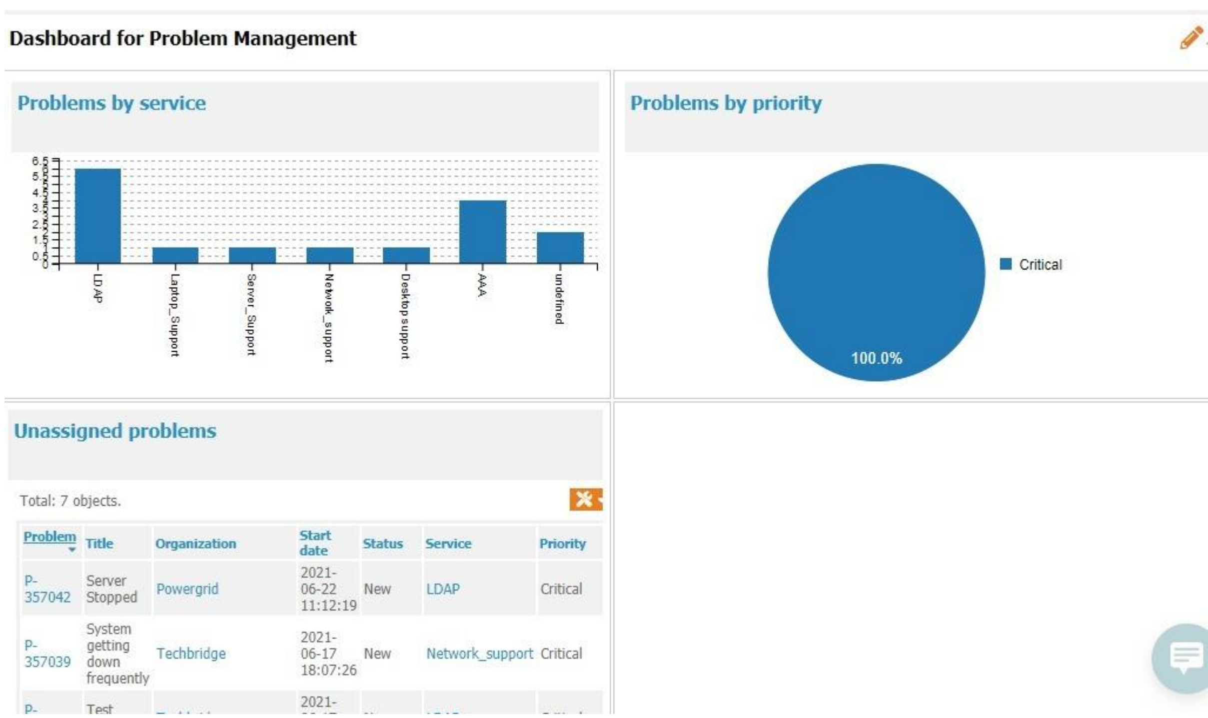
6.1. Problem

A *problem* is the cause of one or more *incidents* occurring in your IT. At the time the *problem* is recorded, the cause may be still unknown. Such tickets will allow IT engineers to document all the actions made to find the root cause and fix the issue.

The main difference between a *problem* and an *incident* is that an *incident* must be fixed as quickly as possible to reduce the unavailability period of the service, whereas the problem will focus on identifying the root cause. While the root cause remains undefined, a workaround is provided to help in fixing corresponding recurring *incidents*.

The *problems* are managed by users having the profile **Problem manager**.

Dashboard:



6.1.1. Problem Properties

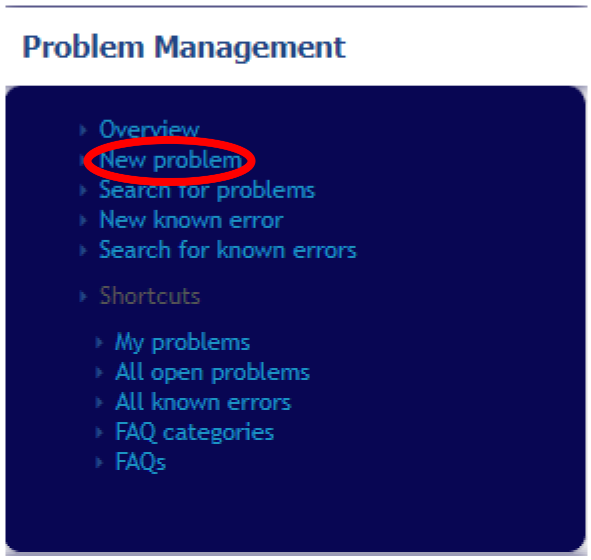
Name	Type	Mandatory?
Ref	Alphanumeric string	Yes
Title	Alphanumeric string	Yes
Organization	Foreign key to a(n) Organization	Yes
Status	Possible values: Assigned, Closed, New, Resolved	Yes
Priority	Possible values: Critical, High, Medium, Low	Yes
Service	Foreign key to a(n) Service	No
Service subcategory	Foreign key to a(n) Service Subcategory	No
Product	Alphanumeric string	No
Impact	Possible values: A Department, A Service, A person	Yes
Urgency	Possible values: critical, high, medium, low	Yes
Description	Multiline character string	Yes
Start date	Date and time (year-month-day hh: mm: ss)	No
Last update	Date and time (year-month-day hh: mm: ss)	No
Assignment Date	Date and time (year-month-day hh: mm: ss)	No
Resolution Date	Date and time (year-month-day hh: mm: ss)	No
Close date	Date and time (year-month-day hh: mm: ss)	No
Caller	Foreign key to a(n) Person	No
Team	Foreign key to a(n) Team	No
Agent	Foreign key to a(n) Person	No
Related Change	Foreign key to a(n) Change	No

Tabs

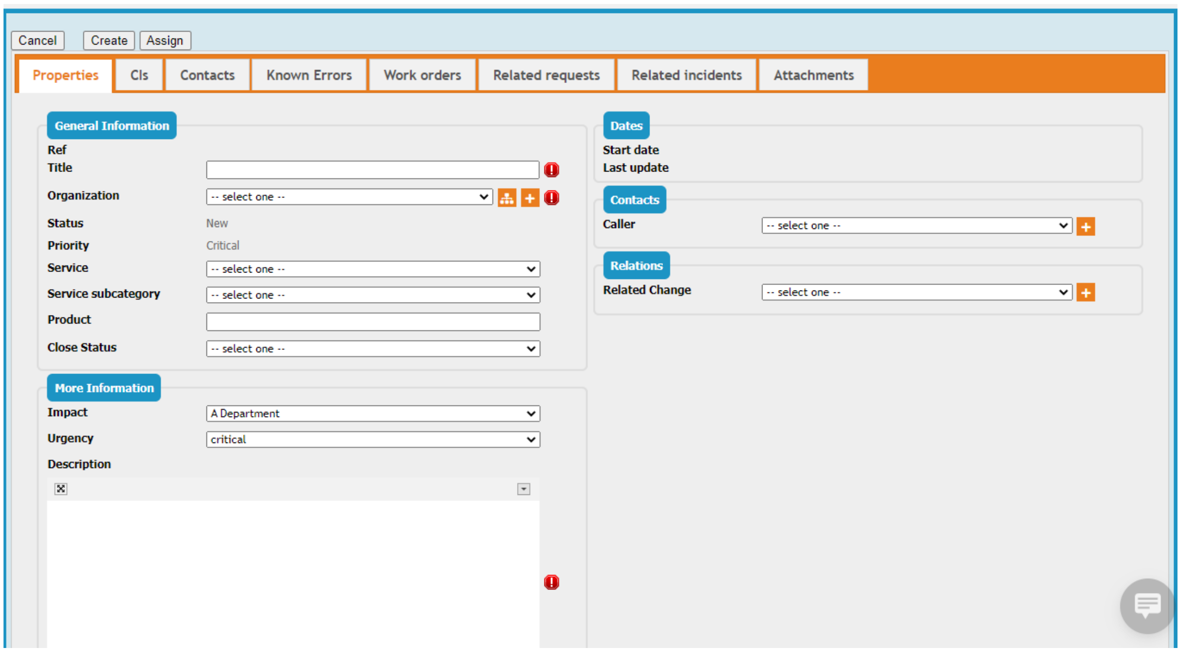
Tab	Description
CIs	All the configuration items impacted for this ticket
Contacts	All the contacts linked to this ticket
Known Errors	All the known errors that are linked to this problem
Work orders	All the work orders for this ticket
Related requests	All the requests that are related to this problem

6.1.2. Creating a Problem

Click on the “New Problem” menu:



The following form is displayed:



6.2. Managing Private Log

A problem ticket only has a private log to keep track of all the activities and communications related to it.

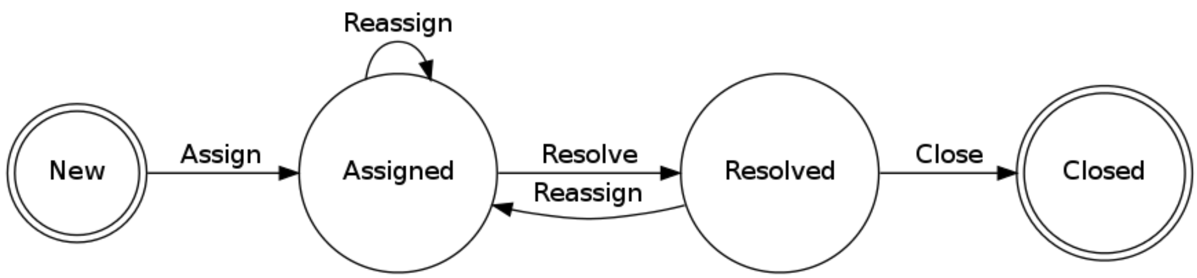
This log is not visible on the end user portal.

6.3. Managing CIs and Contacts

Unlike the incident and change management modules, the lists of CIs and contacts that are linked to a problem are managed manually.

6.4. Problem Life Cycle

Problem objects have the following life cycle:



Depending on the status of the object, the constraints on the properties vary as shown on the table below:

	New	Assigned	Resolved	Closed
Ref	R/O	R/O	R/O	R/O
Title			R/O	R/O
Organization			R/O	R/O
Status	R/O	R/O	R/O	R/O
Priority	R/O	R/O	R/O	R/O
Service			M	R/O
Service subcategory				R/O
Product				R/O
Impact			R/O	R/O
Urgency			R/O	R/O
Description			R/O	R/O
Start date	R/O	R/O	R/O	R/O
Last update	R/O	R/O	R/O	R/O
Assignment Date	H	R/O	R/O	R/O
Resolution Date	H	H	H	H
Close date	H	H	H	R/O
Caller			R/O	R/O
Team	H	M	R/O	R/O
Agent	H	M	R/O	R/O
Related Change				R/O

Table key:

- H: hidden
- R/O: read-only
- M: mandatory

6.5. Known Errors Database Module

6.5.1. FAQ

The frequently asked questions (FAQs) answer the most common questions asked by users about a service. FAQs can easily solve common problems.

FAQ Properties

Name	Type	Mandatory?
Category	Foreign key to a(n) FAQ Category	Yes
Title	Alphanumeric string	Yes
Error code	Alphanumeric string	No
Key words	Alphanumeric string	No
Summary	Multiline character string	No
Description	HTML string	No

6.5.2. Creating a FAQ

The menu “FAQs” provides a list of FAQS. Just click on the button “New” to create a new one.

6.5.3. FAQ Category

Categorizing the FAQs is a helper for the agent who is searching for a FAQ.

FAQ Category Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes

Tabs

Tab	Description
FAQs	All the frequently asked questions related to this category

6.5.4. Known Error

The *known errors* are part of the *problem management* process. They are used to document known issues even if the problem is not completely fixed, for instance by providing a workaround. This improves the efficiency of the *incident management* process.

A *known error* can be linked to *configuration items*. When a ticket is linked a CI then the known errors related to that CI are displayed automatically in the ticket. You can link documents to a known error.

Known Error Properties

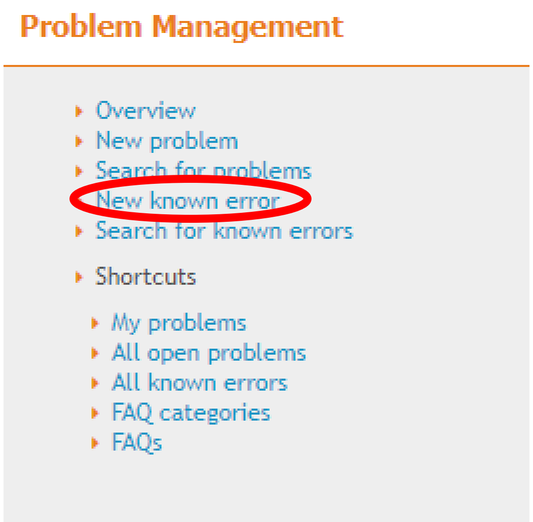
Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Customer	Foreign key to a(n) Organization	Yes
Related Problem	Foreign key to a(n) Problem	No
Error Code	Alphanumeric string	No
Domain	Possible values: Application, Desktop, Network, Server	Yes
Vendor	Alphanumeric string	No
Model	Alphanumeric string	No
Version	Alphanumeric string	No
Symptom	Multiline character string	Yes
Root Cause	Multiline character string	No
Work around	Multiline character string	No
Solution	Multiline character string	No

Tags

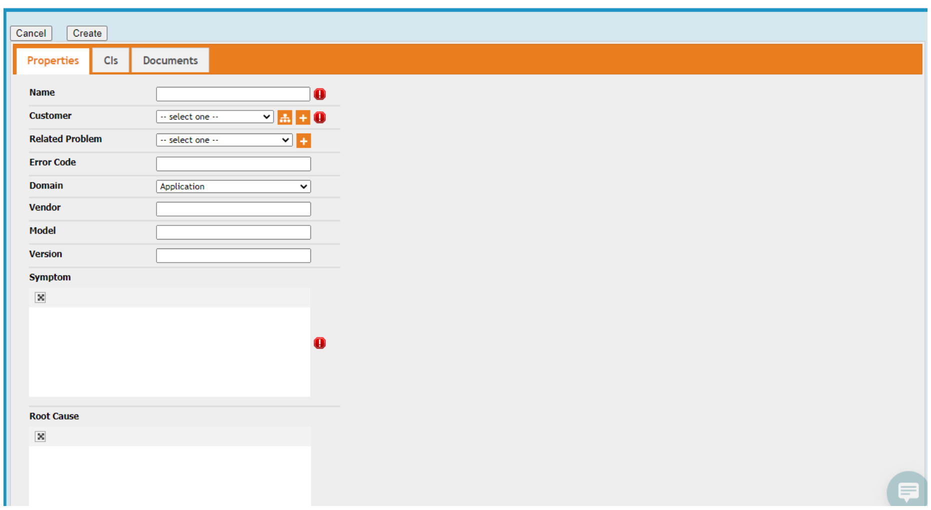
Tab	Description
CIs	All the configuration items that are related to this known error
Documents	All the documents link to this known error

6.5.5. Creating a Known Error

Click on the “New known error” menu:



The following form is displayed:



In addition to its name, and the corresponding organization, a known error can be characterized by the following attributes:

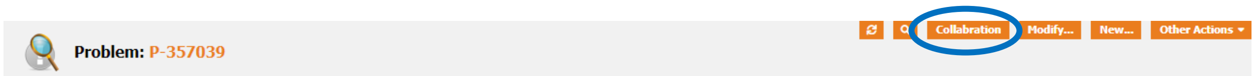
- Error code
- Domain
- Vendor
- Model
- Version

Most of the time, a known error should be linked to a problem. This way, support agents can quickly get the relevant information to troubleshoot and fix.

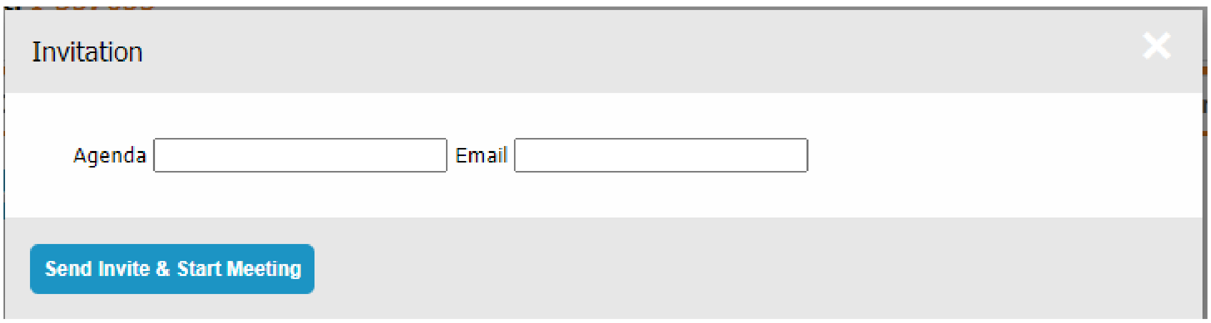
6.6. Collaboration and Communication

The tool is integrated with communication and collaboration tool to facilitate coordination and cooperation with onsite/remote users. It also helps in diagnosis and swarming.

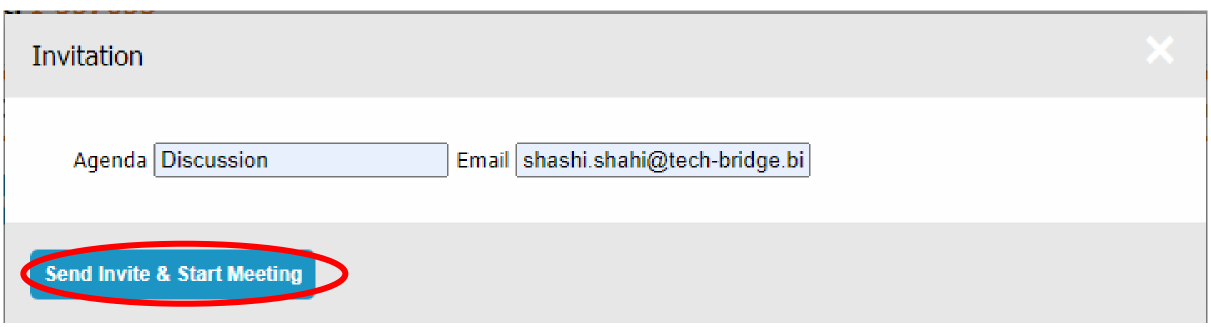
- Click on Collaboration option.



Then, the screen displays are:



- Then, fill the Agenda and mark all the email ids with whom communication is needed.



- Then click on “Send Invite & Start Meeting”.

The screen displayed is

This is the interface where users can communicate and collaborate easily and efficiently.

6.7. Fetching of Adhoc Report

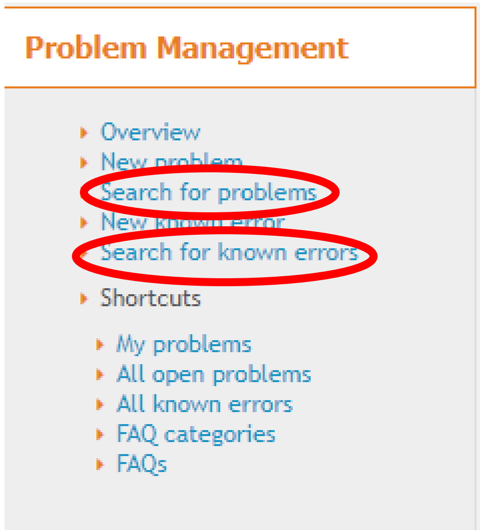
Adhoc report may have tabular data arrangement, a matrix as well as a chart. On Adhoc Wizard, you can also:

- Sort the data to re-arrange it in order of your choice
- Get totals / summaries
- Highlights selective records (when a condition is met)

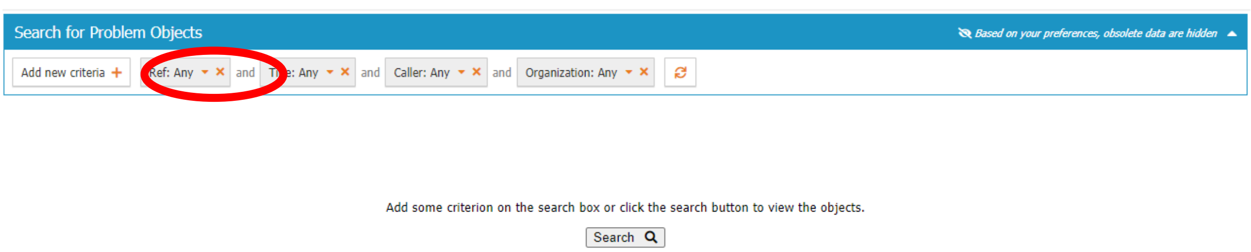
When you run an ad hoc report in output format, the viewer provides Power Viewer that helps altering the report on the viewer itself.

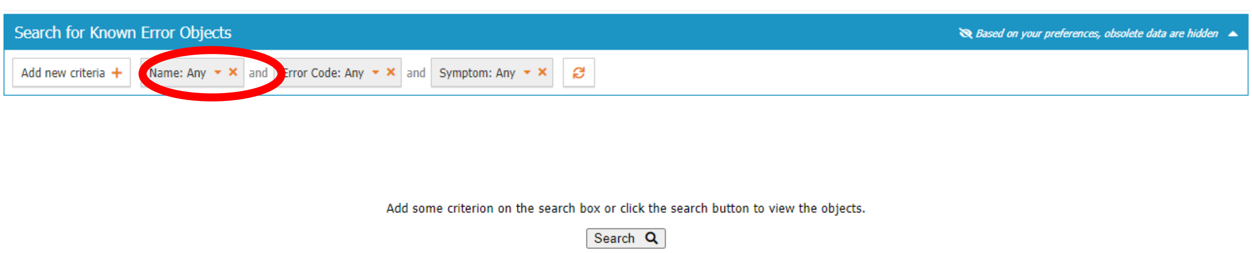
Steps for fetching Adhoc Report: -

1. Click on “Search for problems” or “Search for known errors”.

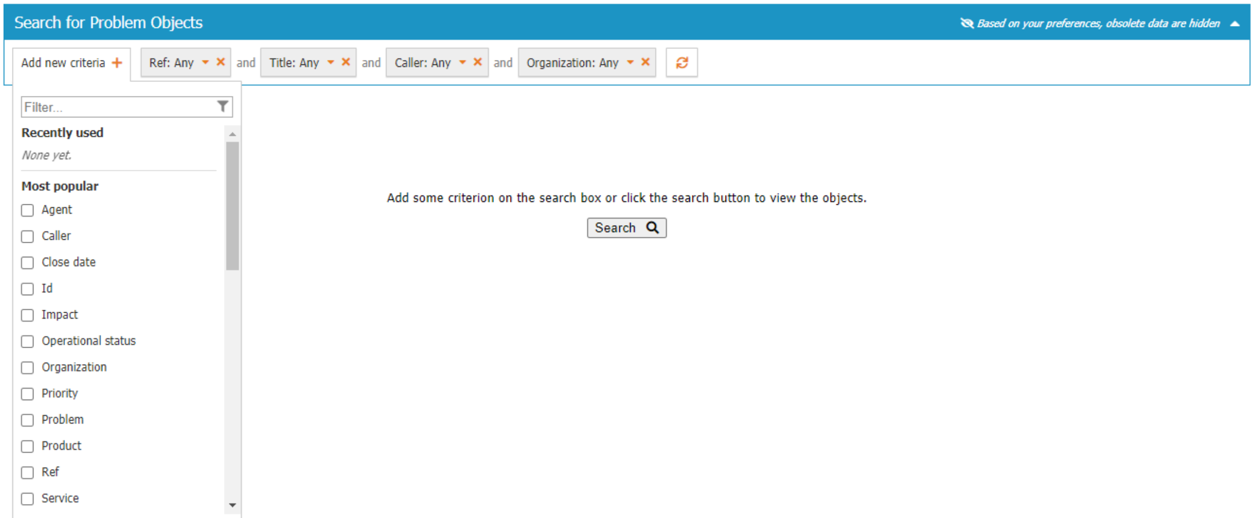


2. Click on “Add new criteria” on Problem Objects or Known Error Objects.

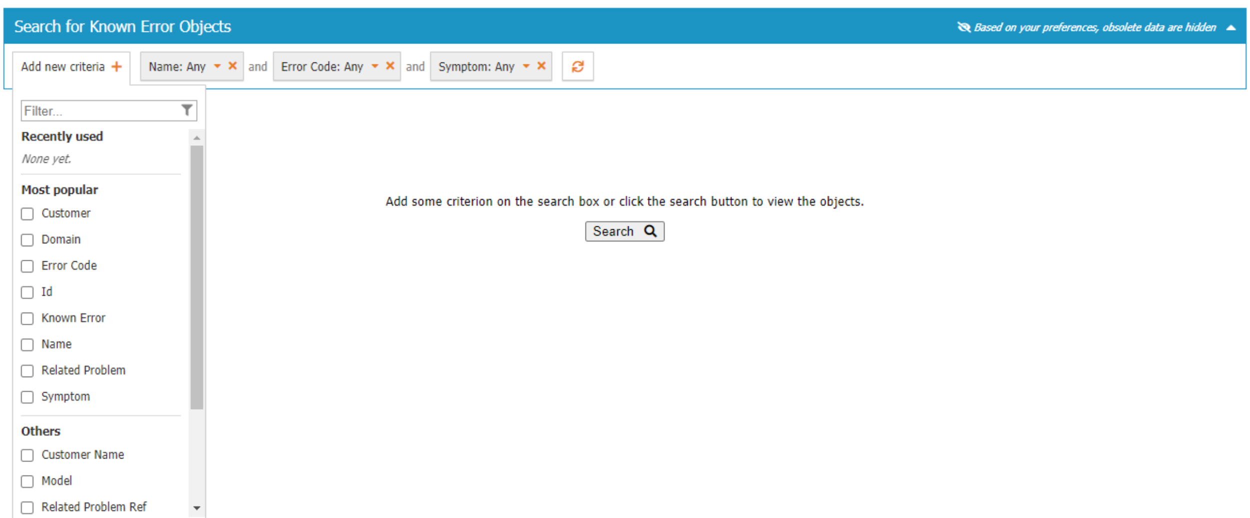




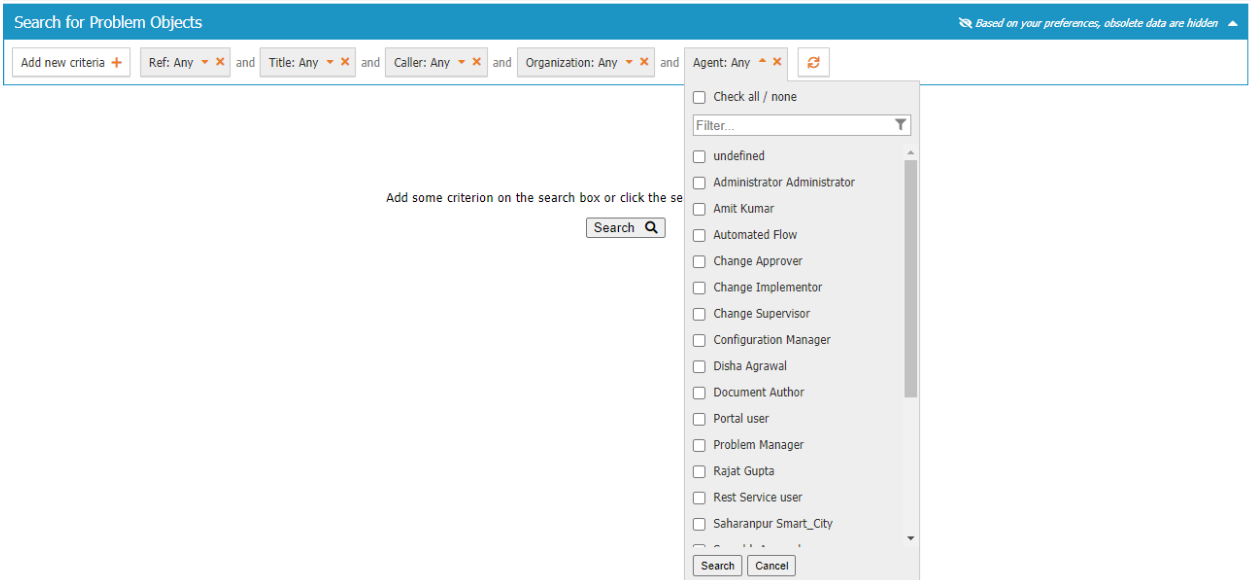
3. Select the Criteria.



Or

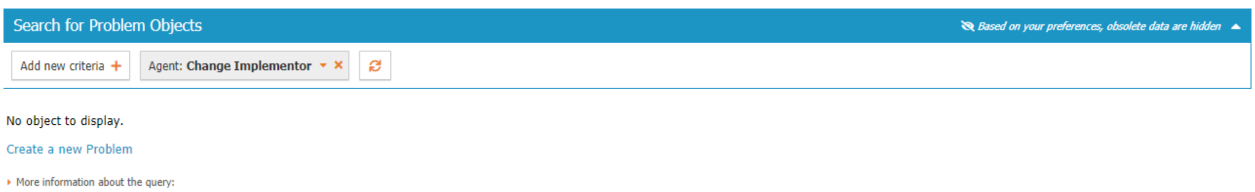


For example, if you select Agent, then you will get the display as:



4. Then select the values and then click on Search.

5. The display will be shown as below:



7. Change Management Module

A *change ticket* or simply a *change* keeps track of modifications planned in the IT:

- Patch installations
- System configuration changes
- OS updates
- Software installations

This way you can track all the modifications made in your IT. A lot of incidents are due to changes made in the IT environment. By documenting them, you can identify easily what changes had been made when an incident occurs and restore the service more quickly.

Moreover, this change management module allows you to analyze automatically the impact of the changes on the infrastructure and the application solutions. IT engineers can then better control the unavailability of the critical services in the enterprise, and improve customer satisfaction.

Change tickets are managed by the people having the profiles:

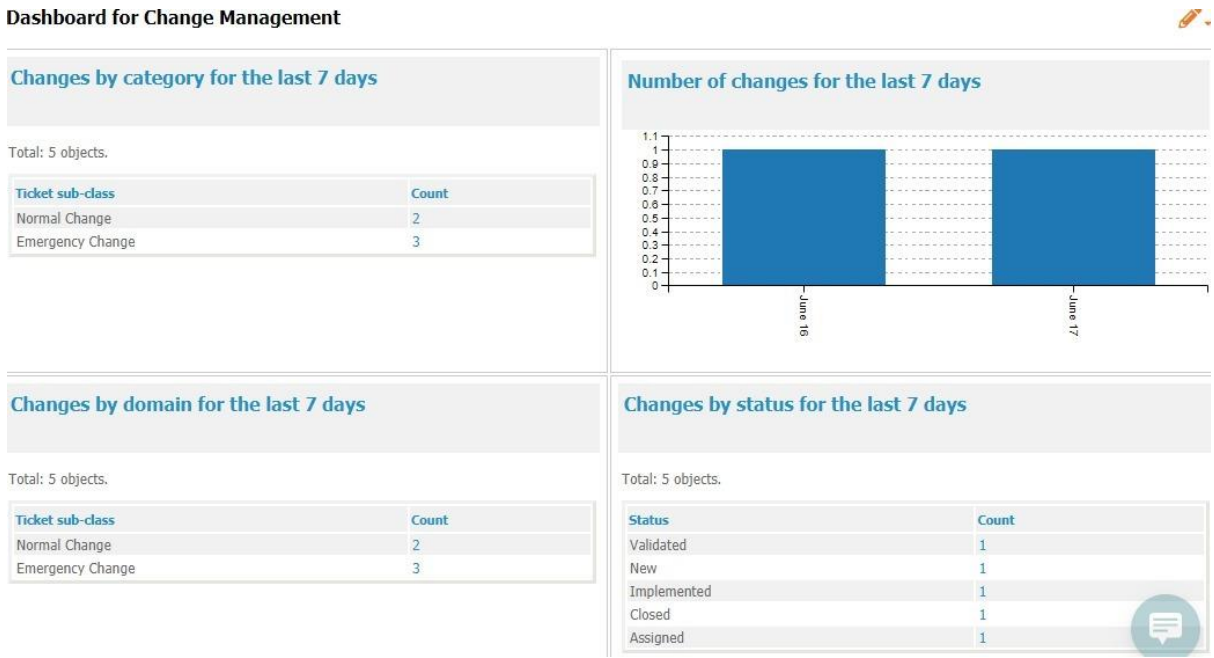
- **Change implementors** plan and implement the changes
- **Change supervisor** follow up with the changes
- **Change manager** approve the changes

When installing tbITSM v.1.0., you have the choice between two different modules for documenting changes. The module described here has been designed as a simple workflow to ease the management of changes:

- **Routine Changes**
- **Normal Changes**
- **Emergency Changes**

The differences between those three types of changes rely in their respective workflows.

Dashboard:



7.1. Emergency Changes

An ITIL V4 emergency change is the highest priority change that can be defined in an organization. Emergency changes are defined as changes that need to be evaluated, assessed and either rejected or approved in a short timeframe. Simply defining a change as an emergency does not automatically entail the change should be implemented. The Emergency Change Advisory Board (ECAB) will assess the change and provide advice to the delegated person responsible for approving or rejecting emergency changes.

Emergency Change Properties

Name	Type	Mandatory?
Ref	Alphanumeric string	Yes
Organization	Foreign key to a(n) Organization	Yes
Status	Possible values: Approved, Assigned, Closed, Implemented, Monitored, New, Not approved, Planned and scheduled, Rejected, Validated	No
Title	Alphanumeric string	Yes
Description	Multiline character string	Yes
Approval comment	Alphanumeric string	No
Caller	Foreign key to a(n) Person	No
Team	Foreign key to a(n) Team	No
Agent	Foreign key to a(n) Person	No
Supervisor team	Foreign key to a(n) Team	No
Supervisor	Foreign key to a(n) Person	No

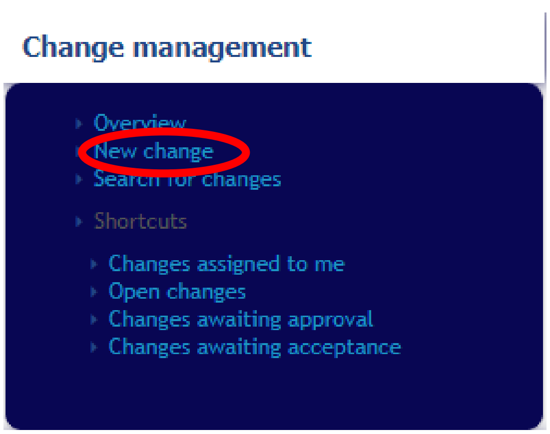
Manager team	Foreign key to a(n) Team	No
Manager	Foreign key to a(n) Person	No
Reject reason	Alphanumeric string	No
Impact	Alphanumeric string	No
Outage	Possible values: No, Yes	Yes
Fallback plan	Multiline character string	No
Parent change	Foreign key to a(n) Change	No
Creation date	Date and time (year-month-day hh: mm: ss)	No
Start date	Date and time (year-month-day hh: mm: ss)	No
End date	Date and time (year-month-day hh: mm: ss)	No
Last update	Date and time (year-month-day hh: mm: ss)	No
Approval Date	Date and time (year-month-day hh: mm: ss)	No
Close date	Date and time (year-month-day hh: mm: ss)	No

Tabs

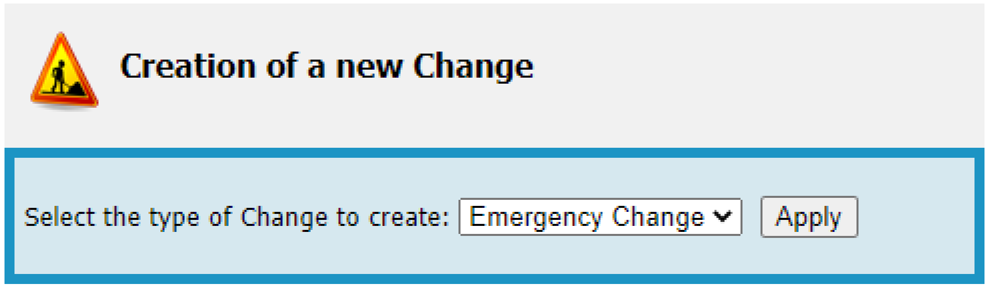
Tab	Description
CIs	All the configuration items impacted for this ticket
Contacts	All the contacts linked to this ticket
Work orders	All the work orders for this ticket
Related requests	All the user requests linked to this change
Related Incidents	All the incidents linked to this change
Related problems	All the problems linked to this change
Child changes	All the sub changes linked to this change

7.1.1. Creating an Emergency Change

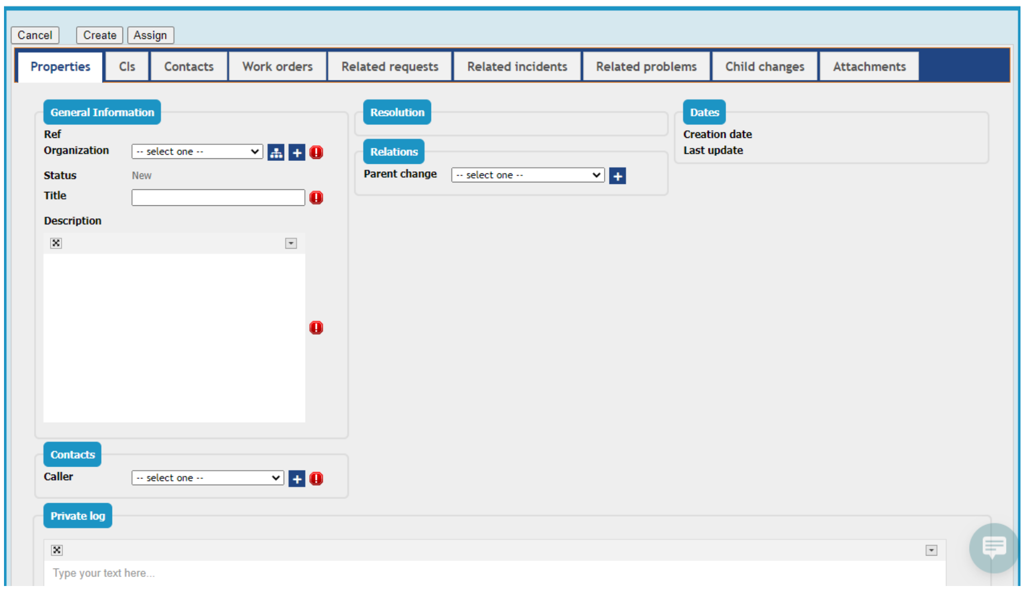
Click on the “New Change” menu:



Then select “Emergency Change” in the form below:

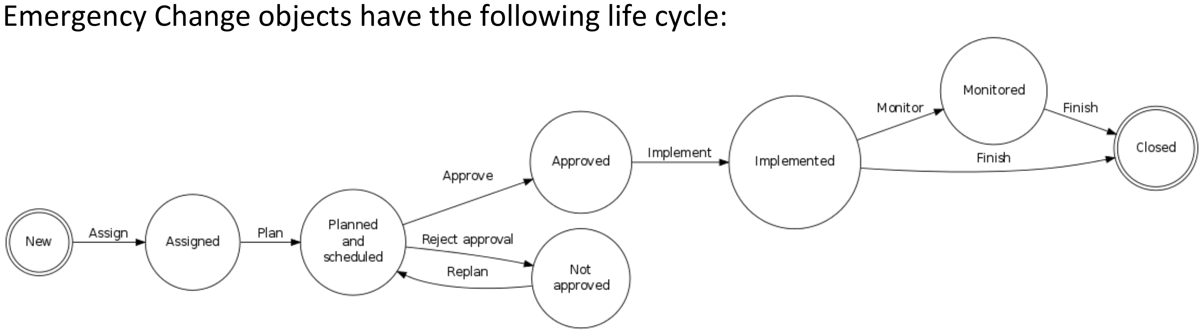


And click “Apply” to display the Emergency Change creation form:



7.1.2. Emergency Change Life Cycle

Emergency Change objects have the following life cycle:



Depending on the status of the object, the constraints on the properties vary as shown on the table below:

	New	Assigned	Planned and scheduled	Approved	Not approved	Impleme nted	Monit ored	Closed
Ref	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
Organization			R/O	R/O	R/O	R/O	R/O	R/O
Status	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
Title	M	M	M	M	M	R/O	R/O	R/O
Description		R/O	R/O	R/O	R/O	R/O	R/O	R/O
Approval comment	H	H	H	M	H	R/O	R/O	R/O
Caller							R/O	R/O
Team	H	M	M	M	M	M	R/O	R/O
Agent	H	M	M	M	M	M	R/O	R/O
Supervisor team	H	M	M	R/O	R/O	R/O	R/O	R/O
Supervisor	H	M	M	R/O	R/O	R/O	R/O	R/O
Manager team	H	M	M	R/O	R/O	R/O	R/O	R/O
Manager	H	M	M	R/O	R/O	R/O	R/O	R/O
Reject reason	H	R/O	R/O	R/O	M	R/O	R/O	R/O
Impact	H	H	M	R/O	R/O	R/O	R/O	R/O
Outage	H	H	M	R/O	M	R/O	R/O	R/O
Fallback plan	H	H	M	M	M	M	R/O	R/O
Parent change							R/O	R/O
Creation date	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
Start date	H	H	M	M	M	R/O	R/O	R/O
End date	H	H	M	M	M	M	R/O	R/O
Last update	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
Approval Date	H	H	H	M	H	R/O	R/O	R/O
Close date	H	H	H	H	H	H	H	R/O

Table key:

- H: hidden
- R/O: read-only
- M: mandatory

7.2. Normal Change

An ITIL V4 normal change refers to changes that must follow the complete change management process. By definition a normal change will proceed through all steps of the change management process and will eventually be reviewed by the Change

Advisory Board (CAB). The CAB will provide advice regarding the change to the person who is deemed responsible to approve or reject normal changes.

Normal Change Properties

Name	Type	Mandatory?
Ref	Alphanumeric string	Yes
Organization	Foreign key to a(n) Organization	Yes
Status	Possible values: Approved, Assigned, Closed, Implemented, Monitored, New, Not approved, Planned and scheduled, Rejected, Validated	No
Title	Alphanumeric string	Yes
Description	Multiline character string	Yes
Approval comment	Alphanumeric string	No
Acceptance comment	Multiline character string	No
Caller	Foreign key to a(n) Person	No
Team	Foreign key to a(n) Team	No
Agent	Foreign key to a(n) Person	No
Supervisor team	Foreign key to a(n) Team	No
Supervisor	Foreign key to a(n) Person	No
Manager team	Foreign key to a(n) Team	No
Manager	Foreign key to a(n) Person	No
Reject reason	Alphanumeric string	No
Impact	Alphanumeric string	No
Outage	Possible values: No, Yes	Yes
Fallback plan	Multiline character string	No
Parent change	Foreign key to a(n) Change	No
Creation date	Date and time (year-month-day hh: mm: ss)	No
Start date	Date and time (year-month-day hh: mm: ss)	No
End date	Date and time (year-month-day hh: mm: ss)	No
Last update	Date and time (year-month-day hh: mm: ss)	No
Approval Date	Date and time (year-month-day hh: mm: ss)	No
Acceptance date	Date and time (year-month-day hh: mm: ss)	No
Close date	Date and time (year-month-day hh: mm: ss)	No

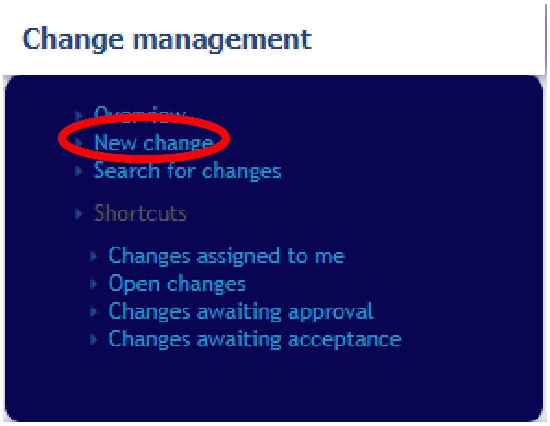
Tabs

Tab	Description
Cis	All the configuration items impacted for this ticket
Contacts	All the contacts linked to this ticket
Work orders	All the work orders for this ticket
Related requests	All the user requests linked to this change
Related incidents	All the incidents linked to this change

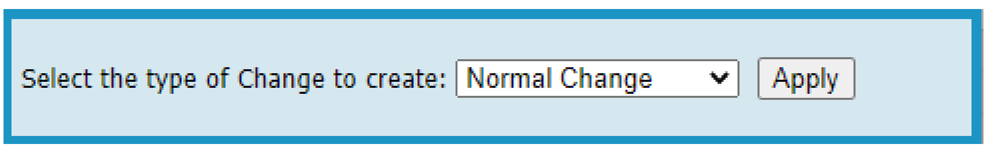
Related problems	All the problems linked to this change
Child changes	All the sub changes linked to this change

7.2.1. Creating a Normal Change

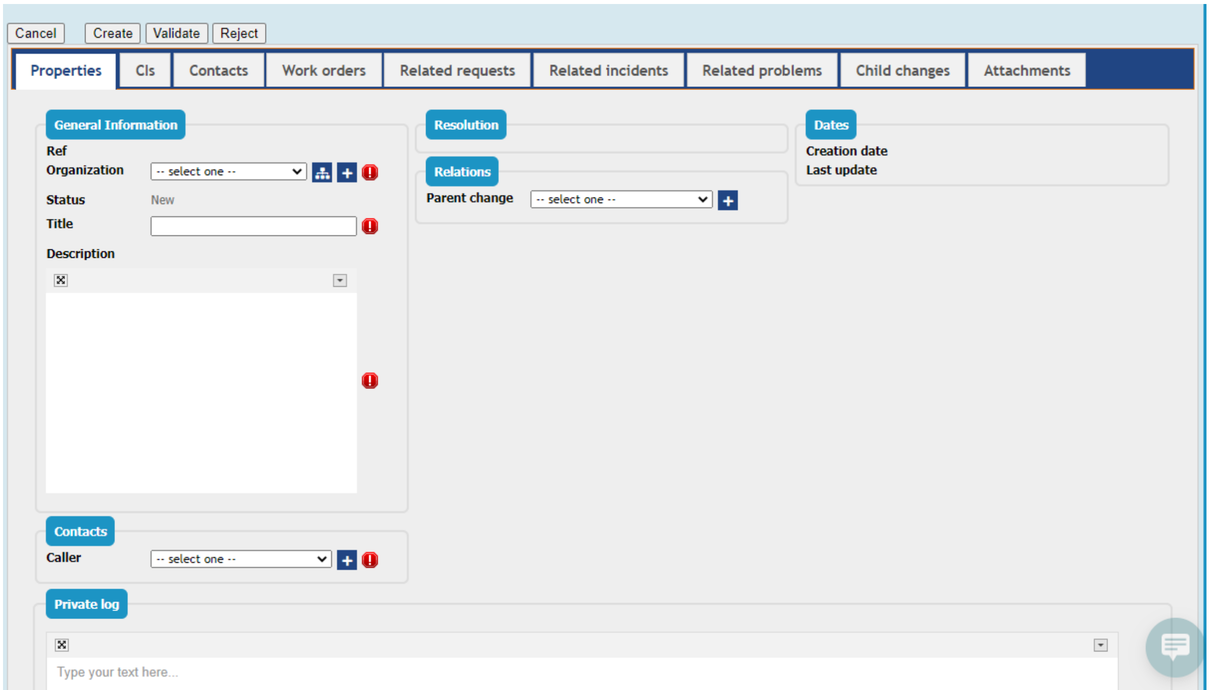
Click on the “New Change” menu:



Then select “Normal Change” in the form below:

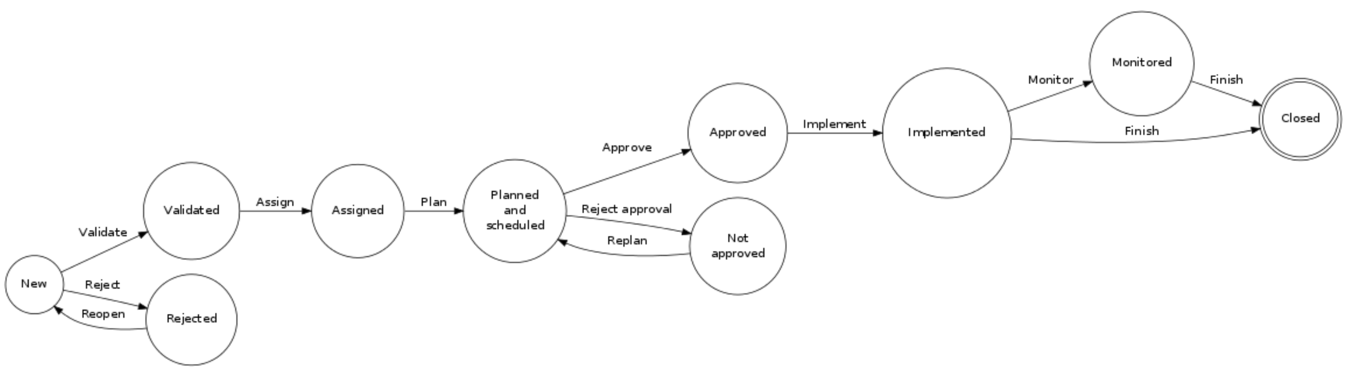


And click “Apply” to display the Normal Change creation form:



7.2.2. Normal Change Life cycle

Normal Change objects have the following life cycle:



Depending on the status of the object, the constraints on the properties vary as shown on the table below:

	New	Validated	Rejected	Assigned	Planned and scheduled	Approved	Not approved	Implemented	Monitored	Closed
Ref	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
Organization					R/O	R/O	R/O	R/O	R/O	R/O
Status										
Title	M	M	M	M	M	M	M	R/O	R/O	R/O
Description		R/O		R/O	R/O	R/O	R/O	R/O	R/O	R/O
Approval comment	H	H	H	H	H	M	H	R/O	R/O	R/O

Acceptance comment	H	M	H		R/O	R/O	R/O	R/O	R/O	R/O
Caller	M		M						R/O	R/O
Team	H	M							R/O	R/O
Agent	H	H	H	M	M	M	M	M	R/O	R/O
Supervisor team	H	M	H	M	M	R/O	R/O	R/O	R/O	R/O
Supervisor	H	H	H	M	M	R/O	R/O	R/O	R/O	R/O
Manager team	H	M	H	M	M	R/O	R/O	R/O	R/O	R/O
Manager	H	H	H	M	M	R/O	R/O	R/O	R/O	R/O
Reject reason	H	R/O	M	R/O	R/O	R/O	M	R/O	R/O	R/O
Impact	H	H	H	H	M	R/O	R/O	R/O	R/O	R/O
Outage	H	H	H	H	M	R/O	M	R/O	R/O	R/O
Fallback plan	H	H	H	H	M	M	M	M	R/O	R/O
Parent change									R/O	R/O
Creation date	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
Start date	H	H	H	H	M	M	M	R/O	R/O	R/O
End date	H	H	H	H	M	M	M	M	R/O	R/O
Last update	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
Approval Date	H	H	H	H	H	M	H	R/O	R/O	R/O
Acceptance date	H	M	H		R/O	R/O	R/O	R/O	R/O	R/O
Close date	H	H	H	H	H	H	H	H	H	R/O

Table key:

- H: hidden
- R/O: read-only
- M: mandatory

7.3. Routine Change

An ITIL V4 routine (standard) change quite simply refers to pre-approved changes. Pre-approved changes can be defined for a variety of tasks, but they will typically be low risk, low effort changes that have a low or known cost.

Routine Change Properties

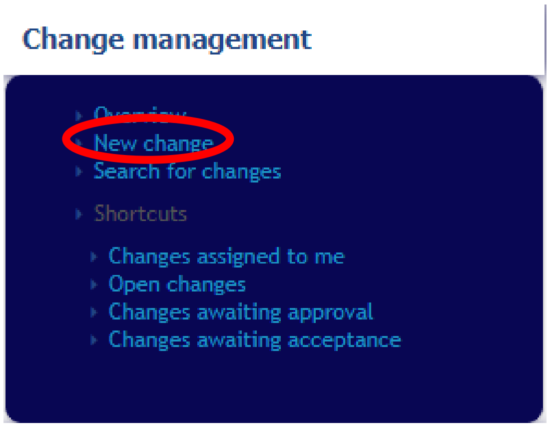
Name	Type	Mandatory?
Ref	Alphanumeric string	Yes
Organization	Foreign key to a(n) Organization	Yes
Status	Possible values: Approved, Assigned, Closed, Implemented, Monitored, New, Not approved, Planned and scheduled, Rejected, Validated	No
Title	Alphanumeric string	Yes
Description	Multiline character string	Yes
Caller	Foreign key to a(n) Person	No
Team	Foreign key to a(n) Team	No
Agent	Foreign key to a(n) Person	No
Supervisor team	Foreign key to a(n) Team	No
Supervisor	Foreign key to a(n) Person	No
Manager team	Foreign key to a(n) Team	No
Manager	Foreign key to a(n) Person	No
Reject reason	Alphanumeric string	No
Impact	Alphanumeric string	No
Outage	Possible values: No, Yes	Yes
Fallback plan	Multiline character string	No
Parent change	Foreign key to a(n) Change	No
Creation date	Date and time (year-month-day hh: mm: ss)	No
Start date	Date and time (year-month-day hh: mm: ss)	No
End date	Date and time (year-month-day hh: mm: ss)	No
Last update	Date and time (year-month-day hh: mm: ss)	No
Close date	Date and time (year-month-day hh: mm: ss)	No

Tabs

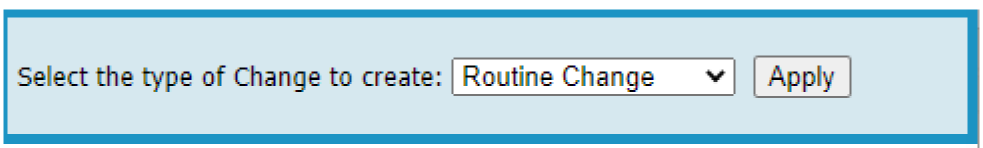
Tab	Description
CIs	All the configuration items impacted for this ticket
Contacts	All the contacts linked to this ticket
Work orders	All the work orders for this ticket
Related requests	All the user requests linked to this change
Related incidents	All the incidents linked to this change
Related problems	All the problems linked to this change
Child changes	All the sub changes linked to this change

7.3.1. Creating a Routine Change

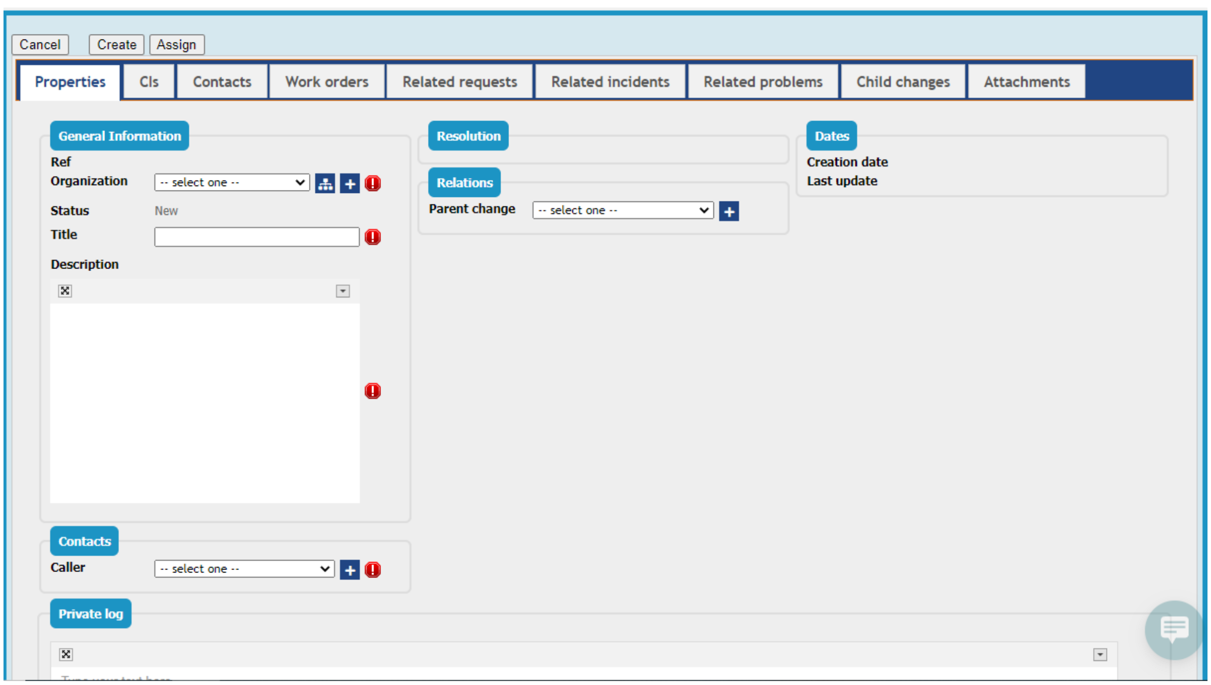
Click on the "New Change" menu:



Then select "Routine Change" in the form below:

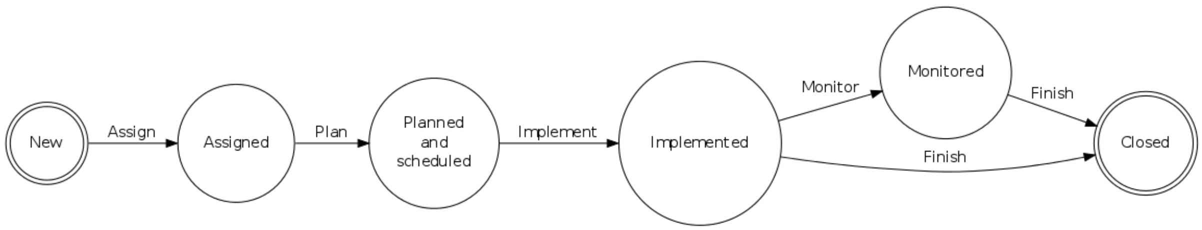


And click "Apply" to display the Routine Change creation form:



7.3.2. Routine Change Life Cycle

Routine Change objects have the following life cycle:



Depending on the status of the object, the constraints on the properties vary as shown on the table below:

	New	Assigned	Planned and scheduled	Implemented	Monitored	Closed
Ref	R/O	R/O	R/O	R/O	R/O	R/O
Organization			R/O	R/O	R/O	R/O
Status	R/O	R/O	R/O	R/O	R/O	R/O
Title	M	M	M	R/O	R/O	R/O
Description		R/O	R/O	R/O	R/O	R/O
Caller					R/O	R/O
Team	H	M	M	M	R/O	R/O
Agent	H	M	M	M	R/O	R/O
Supervisor team	H	M	M	R/O	R/O	R/O
Supervisor	H	M	M	R/O	R/O	R/O
Manager team	H	M	M	R/O	R/O	R/O
Manager	H	M	M	R/O	R/O	R/O
Reject reason	H	R/O	R/O	R/O	R/O	R/O
Impact	H	H	M	R/O	R/O	R/O
Outage	H	H	M	R/O	R/O	R/O
Fallback plan	H	H	M	M	R/O	R/O
Parent change					R/O	R/O
Creation date	R/O	R/O	R/O	R/O	R/O	R/O
Start date	H	H	M	R/O	R/O	R/O
End date	H	H	M	M	R/O	R/O
Last update	R/O	R/O	R/O	R/O	R/O	R/O
Close date	H	H	H	H	H	R/O

Table key:

- H: hidden
- R/O: read-only
- M: mandatory

7.4. Assigning a User Request to a Team and Agent

The list of teams to which you can assign a change is defined by the delivery model of the corresponding organization. When creating a change, the agent has to select the customer organization, then the list of teams is strictly limited to the teams defined for this customer. If a team is missing, the delivery model of the customer must be updated to reflect this need. See More about Delivery model for more information.

7.5. Managing Private Log

A change ticket only has a private log to document all the activities and communications related to it. This one is not visible on the end user portal.

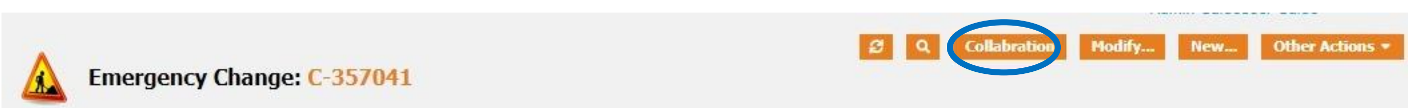
7.6. Managing impacted CIs and Contacts

This section is similar to the one of the Service Request modules. Please refer to it.

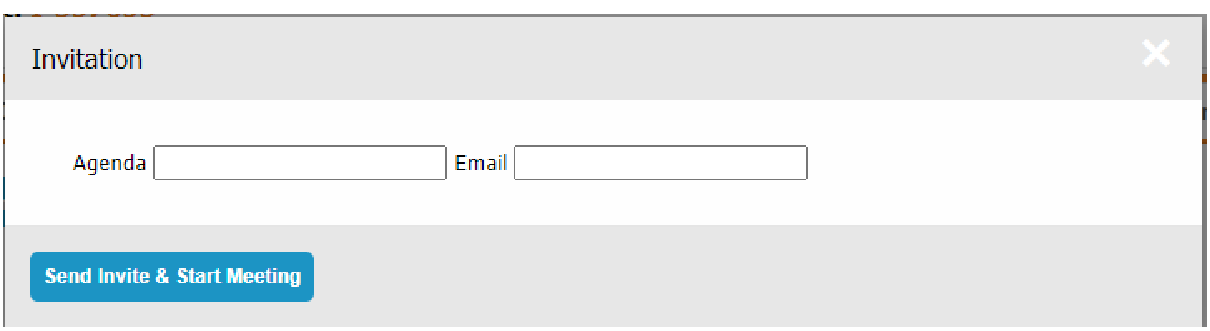
7.7. Collaboration and Communication

The tool is integrated with communication and collaboration tool to facilitate coordination and cooperation with onsite/remote users. It also helps in diagnosis and swarming.

- Click on Collaboration option.



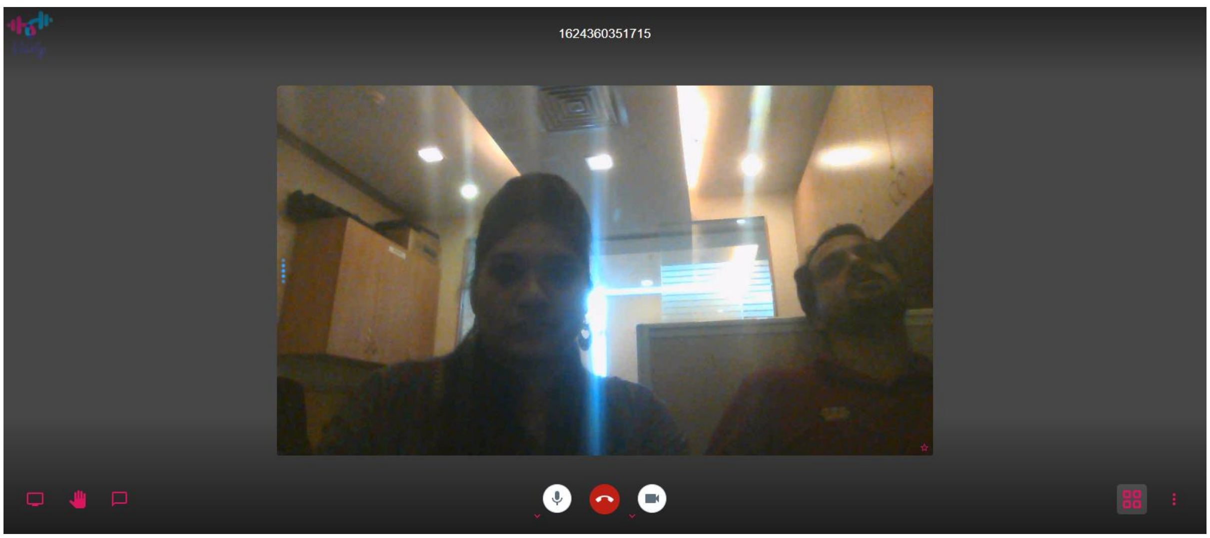
Then, the screen displays are:



- Then, fill the Agenda and mark all the email ids with whom communication is needed.

- Then click on "Send Invite & Start Meeting".

The screen displayed is



This is the interface where users can communicate and collaborate easily and efficiently.

7.8. Kanban Board

Kanban boards to track and visualize the flow of work.

7.9. Fetching of Adhoc Report

Adhoc report may have tabular data arrangement, a matrix as well as a chart. On Adhoc Wizard, you can also:

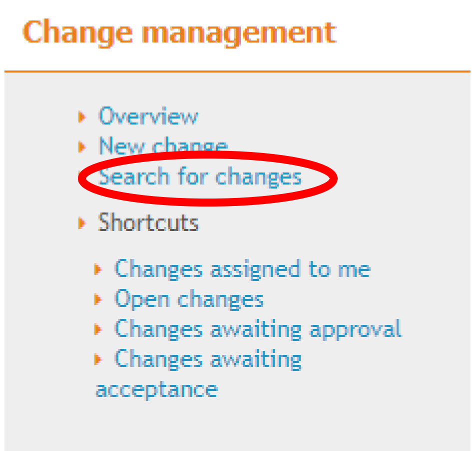
- Sort the data to re-arrange it in order of your choice

- Get totals / summaries
- Highlights selective records (when a condition is met)

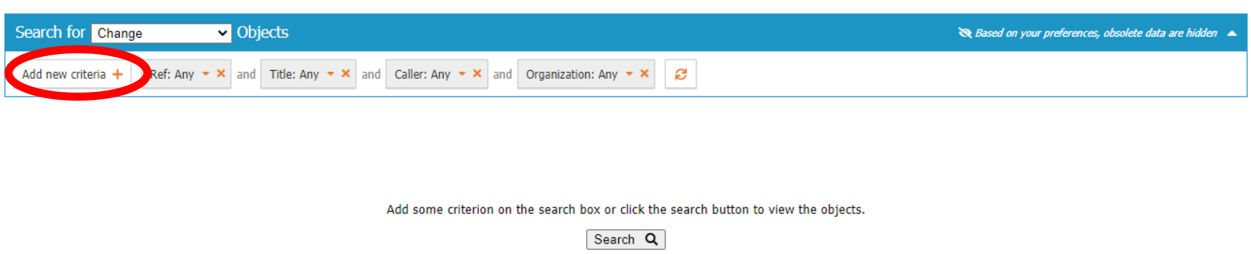
When you run an ad hoc report in output format, the viewer provides Power Viewer that helps altering the report on the viewer itself.

Steps for fetching Adhoc Report: -

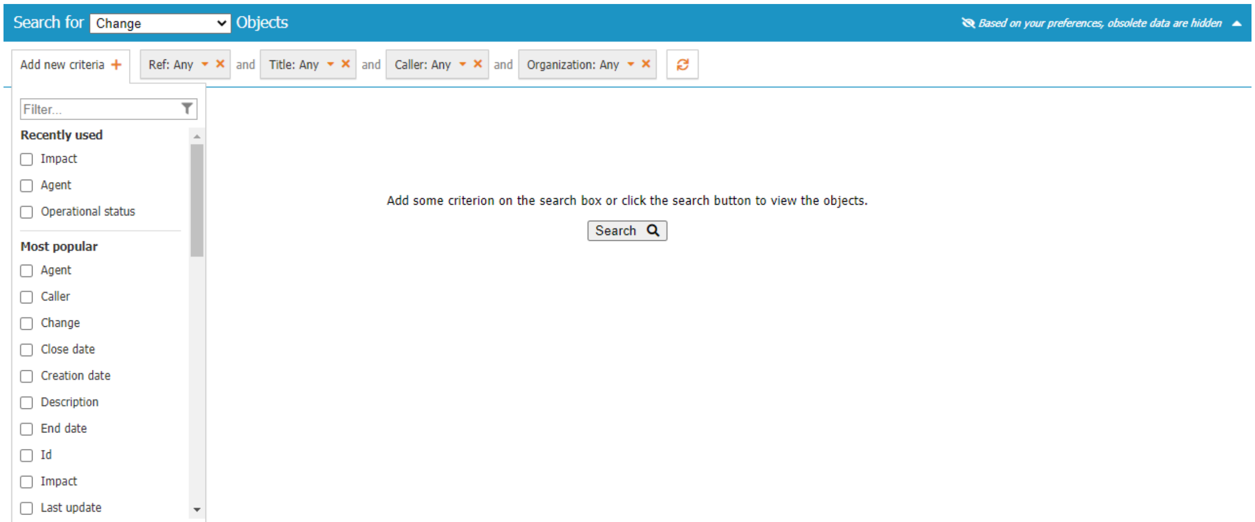
1. Click on “Search for changes”.



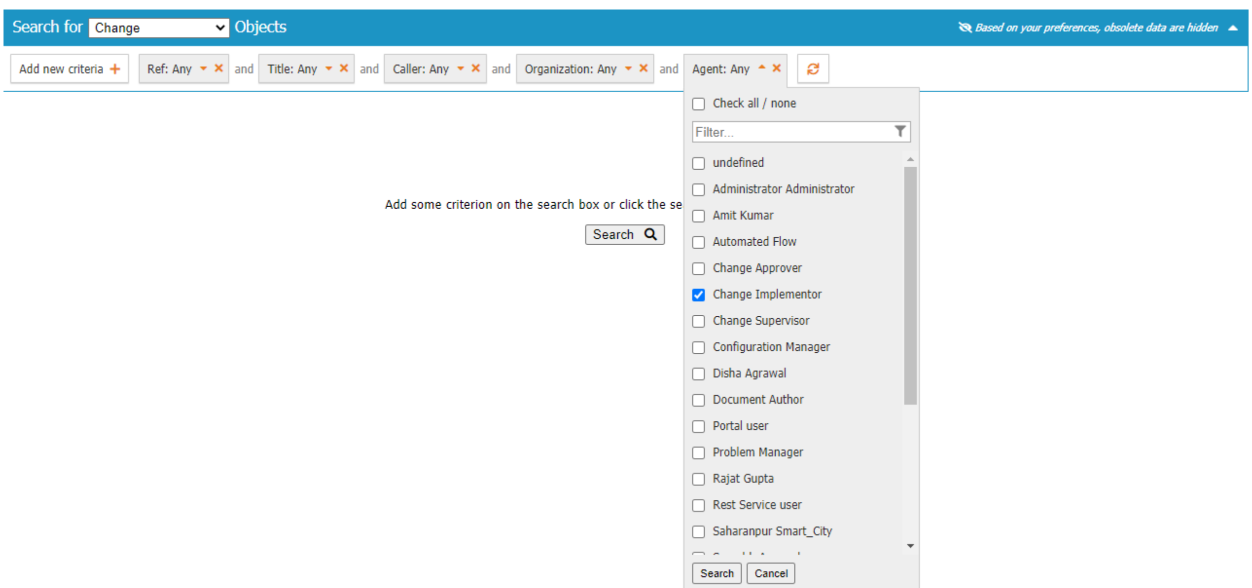
2. Click on “Add new criteria” on Problem Objects or Known Error Objects.



3. Select the Criteria.

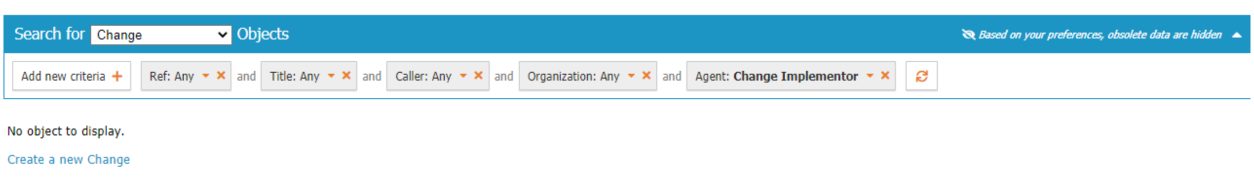


For example, if you select Agent, then you will get the display as:



4. Then select the values and then click on Search.

5. The display will be shown as below:



8. Service Level Management (services, SLAs, contracts) Module

IT Service Management (ITSM) is a process-based practice intended to align the delivery of IT services with needs of the enterprise. Part of this, is the management of the service catalog that defines *services*, *SLA* and *contracts* with the end users (or customers).

This module allows a service manager to define all the pieces of the service catalog:

- Service
- Service subcategories
- SLA & SLT
- Customer contracts
- Provider contracts

The *service management* is integrated with the *ticket management* system:

- When creating a ticket for a given customer, the agent can select the service amongst the list of services defined for this customer.
- Ticket deadlines are computed depending on the SLA signed with the customer.

There are two alternatives to manage the service catalog in tbITSM v.1.0.:

- Service Management (services, SLAs, contracts)
- Service Management (services, SLAs, contracts) for Service providers

The first one is used to manage service within a single company, whereas the second is designed for service providers. Both modules provide the same elements to be managed, the differences are the relationships between those elements. In the sections below we describe those differences.

Dashboard:


Dashboard for Service Management

Customer contracts to be renewed in 30 days

Total: 3 objects.

Customer Contract	Name	Status	Organization	Description	Start date
Everest	Everest	production	Techbridge		2021-05-17
PGCIL	PGCIL	implementation	Techbridge		2021-05-17
Tech-Bridge_coustomer	Tech-Bridge_coustomer	production	Techbridge	contract	2021-05-08

Provider contracts to be renewed in 30 days

Total: 2 objects.

Provider Contract	Name	Status	Organization	Description	Start date	End date	Provider
SLA_Client	SLA_Client	production	Techbridge	test	2021-06-08	2021-06-08	Techbridge
SLA_TB	SLA_TB	production	Techbridge		2021-06-08	2021-06-08	Techbridge

8.1. Contract Type

The contract type allows you to define a typology for the customer and provider contract in order to regroup them. For instance:

- Hardware contract
- Maintenance contract
- Outsourcing contract

Contract Type Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes

8.2. Customer Contract

A customer contract allows you to define which services have been purchased (requested) by a customer and what are the SLA corresponding to those services. By default, no coverage window is defined in tbITSM v.1.0.

It is possible to document several contracts for a given customer. For each customer contracts you can link documents, and contacts related to it.

Customer Contract Properties

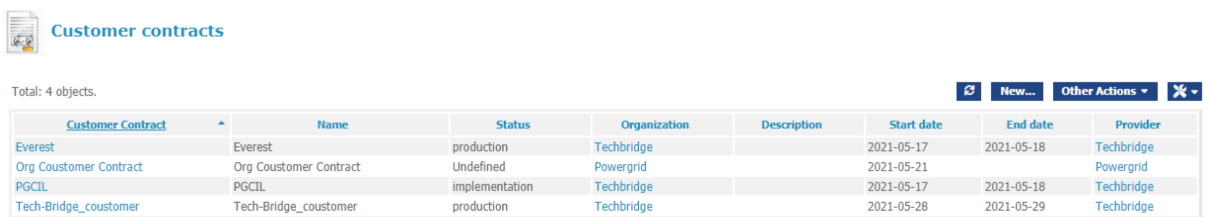
Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Customer	Foreign key to a(n) Organization	Yes
Description	Multiline character string	No
Start date	Date (year-month-day)	No
End date	Date (year-month-day)	No
Cost	Alphanumeric string	No
Cost Currency	Possible values: Dollars, Euros	No
Billing frequency	Alphanumeric string	No
Cost unit	Alphanumeric string	No
Provider	Foreign key to a(n) Organization	Yes
Status	Possible values: implementation, obsolete, production	No
Contract type	Foreign key to a(n) Contract Type	No

Tabs

Tab	Description
Contacts	All the contacts for this customer contract
Documents	All the documents for this customer contract
Services	All the services purchased for this contract

8.3. Creating a Customer Contract

The menu “Customer contract” provides a list of already defined customer contracts. Just click on the button “New” to create a new one.



Customer contracts

Total: 4 objects.

New... Other Actions

Customer Contract	Name	Status	Organization	Description	Start date	End date	Provider
Everest	Everest	production	Techbridge		2021-05-17	2021-05-18	Techbridge
Org Coustomer Contract	Org Coustomer Contract	Undefined	Powergrid		2021-05-21		Powergrid
PGCIL	PGCIL	implementation	Techbridge		2021-05-17	2021-05-18	Techbridge
Tech-Bridge_coustomer	Tech-Bridge_coustomer	production	Techbridge		2021-05-28	2021-05-29	Techbridge

8.4. Delivery Model

A delivery model defines which teams are providing support and services to a set of organizations (customers). It can be used also to document key people with their role:

- Service Manager
- Chief technology officer
- Helpdesk manager

Several customers can share the same delivery model. Nevertheless, it may be useful to define several delivery models, especially if you want to limit the number of teams to which you can assign a ticket.

The delivery model of a given customer is used to identify to which team you can assign a ticket for this one.

Delivery Model Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Organization	Foreign key to a(n) Organization	Yes

Description	Multiline character string	No

Tabs

Tab	Description
Contacts	All the contacts (Teams and Person) for this delivery model
Customers	All the customers having this delivering model

Creating a Delivery Model

The menu "Delivery models" provides a list of already defined *delivery models*. Just click on the button "New" to create a new one.

8.5. Provider Contract

A provider contract allows you to document all the contracts you have with your providers (Support, maintenance, hardware contracts, etc.) This helps you to retrieve quickly such information and better manage the contract renewal process.

You can also link *configuration items* that are covered by a provider contract. For each customer contracts you can link documents, and contacts related to it.

Provider Contract Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Customer	Foreign key to a(n) Organization	Yes
Description	Multiline character string	No
Start date	Date (year-month-day)	No
End date	Date (year-month-day)	No
Contract type	Foreign key to a(n) Contract Type	No
Cost	Alphanumeric string	No
Cost Currency	Possible values: Dollars, Euros	No
Billing frequency	Alphanumeric string	No
Cost unit	Alphanumeric string	No
Provider	Foreign key to a(n) Organization	Yes
SLA	Alphanumeric string	No
Service hours	Alphanumeric string	No
Status	Possible values: implementation, obsolete, production	No

Tabs

Tab	Description
Contacts	All the contacts for this customer contract
Documents	All the documents for this customer contract
Cis	All the configuration items covered by this provider contract

Creating a Provider Contract

The menu “Provider contract” provides a list of already defined provider contracts. Just click on the button “New” to create a new one.

8.6. SLA

A *service level agreement* (SLA) is a group of SLT. It defines the agreement between a provider and a customer for a given set of services.

Each IT department can define its own list of SLAs.

SLA Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Provider	Foreign key to a(n) Organization	Yes
description	Multiline character string	No

Tabs

Tab	Description
SLTs	All the service level targets for this SLA
Customer contracts	All the customer contracts using this SLA

Creating an SLA

The menu “SLA” provides a list of already defined service level agreements. Just click on the button “New” to create a new one.

8.7. SLT

A *service level target* (SLT) allows you to define metrics for agreements that have to be respected. By default, tbITSM v.1.0. is proposed with two types of metrics:

- *Time to own* (TTO): This is time between the creation of a ticket and the time to take it into account (assign it to an agent)
-
- *Time to resolve* (TTR): This is the time between the creation of a ticket and the time to resolve it

A SLT is defined by:

- a name
- the type of metric
- the type of request
- the priority of the request

SLT Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Priority	Possible values: critical, high, medium, low	No
Request type	Possible values: incident, service request	No
Metric	Possible values: TTO, TTR	No
Value	Numeric value (could be negative)	No
Unit	Possible values: hours, minutes	No

Creating a SLT

The menu "SLT" provides a list of already defined *service level targets*. Just click on the button "New" to create a new one.

8.8. OLAs

An Operational Level Agreement (OLA) is a contract defined between the various departments within an organization to ensure on-time delivery of Services to the Customer. OLAs are designed to address and solve the problem of IT silos by setting forth a specific set of criteria and defining the specific set of IT Services that each department is responsible for.

Applicability and uses:

- Applicable between Incident and Work Order, and Workgroups within an Incident.
- Applicable for Auto created Work Orders (only for the Work Orders created while Incidents are logged and not applicable for sequential Work Orders).
  If the OLA is within an Incident, it corresponds to the Transfer Time. **Example:** OLA between Workgroup 1 and Workgroup 2 is 30 min. An Incident is logged at 10 AM and in the que of Workgroup 1. If it is transferred to Workgroup 2 before 10:30 AM, the OLA is considered as met. If transferred after 10:30 AM, the OLA is considered as not met.

- If the OLA is made between the Incident and Work Order, it corresponds to Resolution Time.
  Example: OLA between Workgroup 1 and Workgroup 2 is 30 min. Incident logged at 10 AM and in the que of Workgroup 1. There is a Work Order created for this incident and is in the que of Workgroup 2. If the Work order is resolved by 10:30AM, the OLA is considered as met. If resolved after 10:30 AM, OLA is considered as not met.

8.9. Work Order

A Work Order (WO) refers to a task or set of tasks, which are part of an initial requirement (Incident or Service Request) that are created where the Incident or Service Request involves different Workgroups/Analysts. All the Work Orders created for the Incident or Service Request should be Resolved before resolving the Incident or Service Request. The Work Orders should meet the OLAs, otherwise, there is an OLA violation.

Example 1: An Incident, is created for a Slow System issue. Analyst A of Global IT Service Desk troubleshoots the Incident. On troubleshooting, it is found that the RAM of the system needs to be increased. Analyst A creates a Work Order, and assigns it to the Asset Management team. Analyst B of Asset Management team installs the RAM in the user’s system and resolves the Work Order. Analyst A validates the performance of the system and resolves the Incident.

Example 2: The HR creates a Service Request (SR) for a new employee joining the organization at Grade 3. After the Project Manager approves the SR, the SR is assigned to the Global IT Service Desk. Multiple Work Orders are created for different activities and assigned to different Workgroups: Work Order for Windows team to create a Windows login id and e-mail id, Work Order for Asset Management team to allocate a laptop, Work Order for SAM team to install appropriate software, and Work Order to Facilities team to provide a cubicle. WO can run in parallel and can be Resolved independently. WO can start only after the completion of WO 224. After all the WOs are completed, the SR gets closed automatically (based on configuration) or Resolved by the Service Desk manually.

Work Orders can be created for Incidents and Service Catalogs. Analyst can create a Work Order for Incident from option. For Service Catalogs, after a Service Request is created for the Service Catalog for which Work Order is configured, the Work Order becomes active automatically and needs to be resolved before resolving the Service Request.

Advantages

- If a user's request cannot be fulfilled by one Workgroup, a Work Order can be created for the concerned Workgroup to fulfill the user's requirement.

- The End User or Analysts do not need to create multiple Incidents or Service Requests for related issues. Especially, for the End User, it is just one Incident or Service Request, which fulfills their requirements (although it may involve multiple Work Orders for the support teams).

8.10. Service

The services are the basis of the *service catalog*. They are used to document all the services that can be provided by one or several IT Departments. For example:

- Application management
- Desktop and End user management
- Network management
- System management

Each IT provider can define its own list of services. Services are used to characterize a user request or an incident.

Services **must** be grouped into *service families* as soon as you use the Enhanced Portal.

For each service you can document:

- the configuration items required to deliver the service
- the provider contracts, on which the service is depending

Service Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Provider	Foreign key to a(n) Organization	Yes
Service Family	Foreign key to a(n) Service Family	No
Description	Multiline character string	No
Status	Possible values: implementation, obsolete, production	No

Tabs

Tab	Description
Service sub categories	All the sub categories for this service
Documents	All the documents linked to the service
Contacts	All the contacts for this service
Customer contracts	All the customer contracts that have purchased this service
Provider contracts	All the provider contracts to support this service
Depends on CIs	All the configuration items that are used to provide this service

Creating a Service

The menu “Service” provides a list of services. Just click on the button “New” to create a new one.

8.11. Service Family

Service Family is the first level in the tree structure of your Service Catalog.

Service Family Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes

Tabs

Tab	Description
Services	All the services in this category

8.12. Service Subcategory

The *service subcategories* are used to define more precisely a service. For example, you can define the following subcategory for the service System management:

- Troubleshooting
- Order a new server
- Configure a new virtual machine
- Repair a server

A *service subcategory* is related to a type of user request (Incident or Service request). This is done to automate the qualification of a user request or an incident.

Service Subcategory Properties

Name	Type	Mandatory?
Name	Alphanumeric string	Yes
Service	Foreign key to a(n) Service	Yes
Provider	Field mapped to an external key	Yes
Status	Possible values: implementation, obsolete, production	No
Request type	Possible values: incident, service request	Yes
Provider Name	Field mapped to an external key	Yes
Description	Multiline character string	No

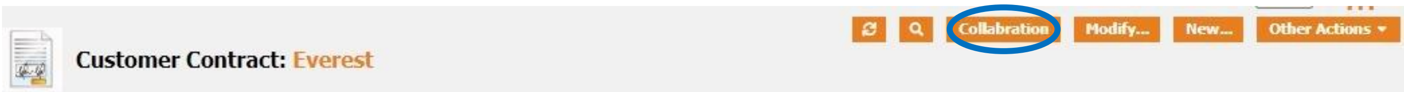
Creating a Service Subcategory

The menu “Service subcategories” provides a list of service subcategories. Just click on the button “New” to create a new one.

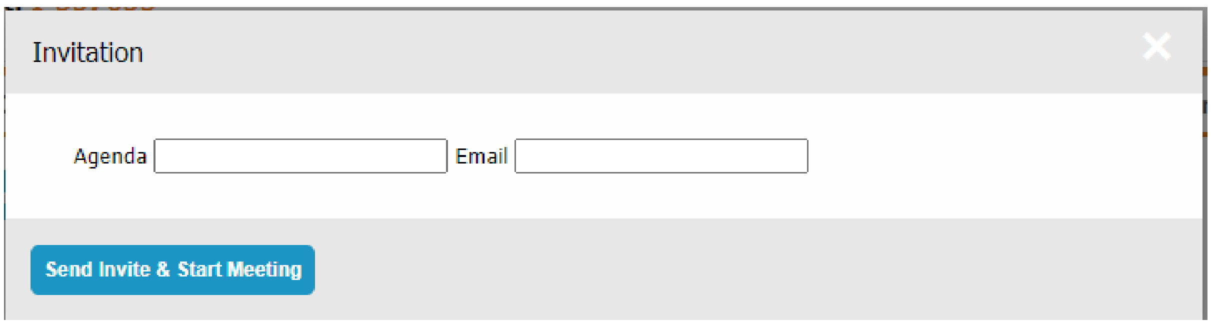
8.13. Collaboration and Communication

The tool is integrated with communication and collaboration tool to facilitate coordination and cooperation with onsite/remote users. It also helps in diagnosis and swarming.

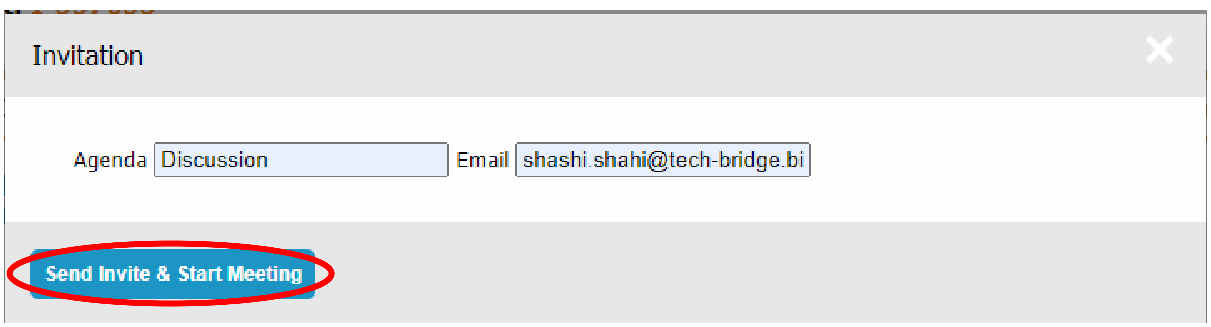
- Click on Collaboration option.



Then, the screen displays are:



- Then, fill the Agenda and mark all the email ids with whom communication is needed.



- Then click on "Send Invite & Start Meeting".

The screen displayed is

This is the interface where users can communicate and collaborate easily and efficiently.

8.14. Fetching of Ad-hoc Report

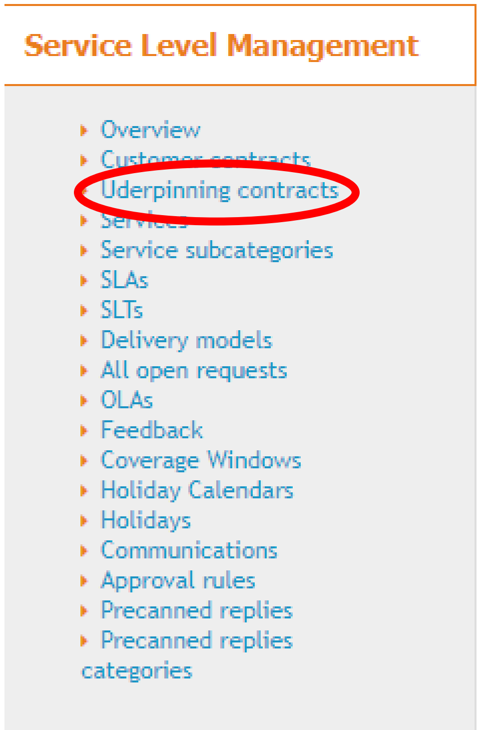
Ad-hoc report may have tabular data arrangement, a matrix as well as a chart. On Ad-hoc Wizard, you can also:

- Sort the data to re-arrange it in order of your choice
- Get totals / summaries
- Highlights selective records (when a condition is met)

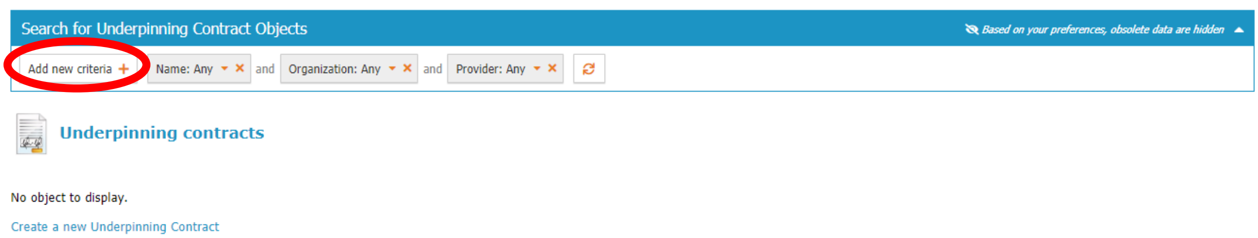
When you run an ad hoc report in output format, the viewer provides Power Viewer that helps altering the report on the viewer itself.

Steps for fetching Ad-hoc Report: -

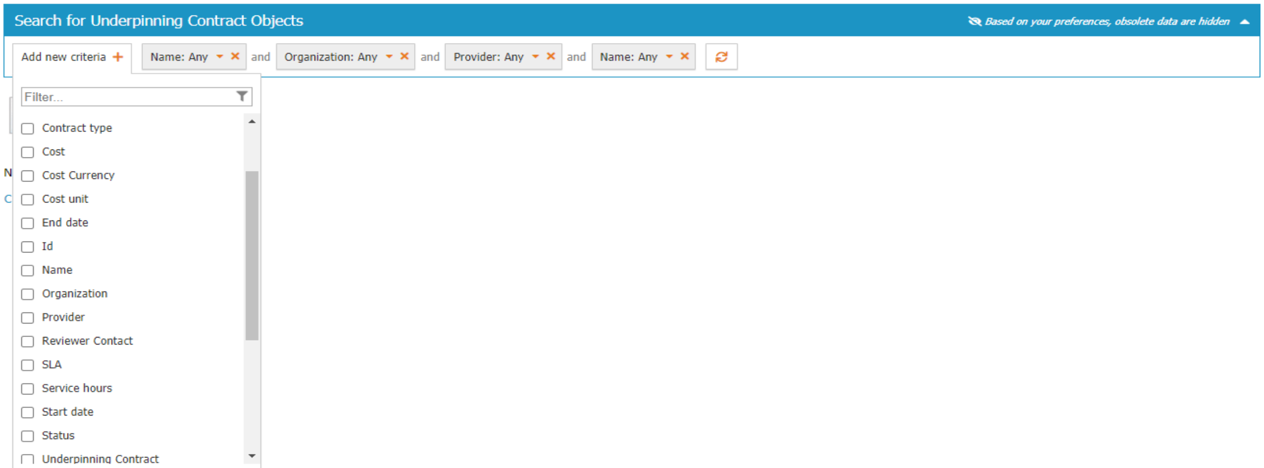
1. Click on any one of the options given “Underpinning contracts”.



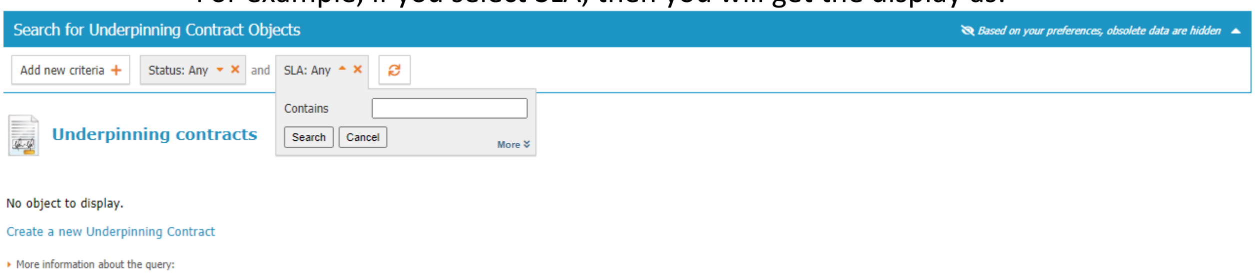
2. Click on “Add new criteria” on Underpinning Contract Objects.



3. Select the Criteria.

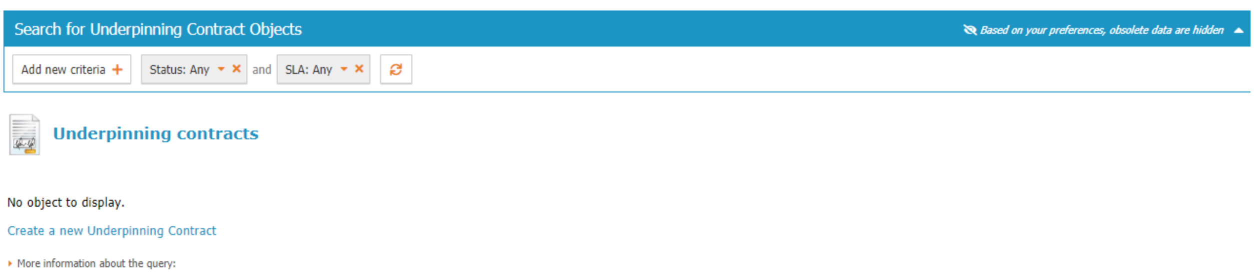


For example, if you select SLA, then you will get the display as:



4. Then select the values and then click on Search.

5. The display will be shown as below:



8.15. Service Improvement

Continual Service Improvement (CSI) uses a metrics-driven approach to identifying opportunities for improvement and to measure the impact of improvement efforts. Although CSI is a phase of the lifecycle and is documented in a separate ITIL publication, CSI can be effective only if it is integrated throughout the lifecycle, creating a culture of continual improvement. CSI should ensure that all participants in service delivery understand that identifying opportunities for improvement is their responsibility.

An important task for CSI is to identify which metrics out of the thousands that are created daily should be monitored. This is done by identifying, for each service or process, what the critical success factors (CSFs) are. CSFs must be present if a process or service is to succeed. It is recommended that each process or service have identified no more than three to five CSFs (one or two in the early life of a service or process).

CSI uses a 7-step process to guide how data is collected and used:

- Define the objectives.
- Determine what to measure.
- Collect the data.
- Process the data.
- Analyze the data.
- Present and use the information.
- Implement improvement.

If CSI is performing its role properly, there will be improvement suggestions arising from all parts of service delivery. The organization is unlikely to have enough resources to implement all of the suggestions, so it is necessary to capture the improvement opportunities, understand their impact, scope, and resource requirements, and prioritize their implementation. CSI uses the CSI register as a tool to document, analyze, and plan for improvements.

As businesses depend more on IT services, it is vital that IT organizations continually evaluate and improve their IT services and the IT service management processes that enable those IT services. A formal, proactive continual service improvement (CSI) practice is required to meet and achieve service level agreements.

To implement CSI, organizations need to instill the right attitude and drive the right behaviors until they become second nature. IT providers must embed a culture of measurement that continually tests the value, quality, performance, and compliance of the services within their portfolio and implements improvement initiatives that enable the desired business outcomes.

By definition, alignment requires bringing together two separate entities, often with disparate goals and objectives. However, in today's complex IT environment, it is becoming more difficult to determine the difference between an IT service and a business service.

As the lines blur, alignment is no longer enough; IT must become an integral part of the business. Rather than having two separate organizations with disconnected goals and objectives, there must be a single, integrated business operation that functions with appropriate technology.

Many people have a personal continual improvement plan (although they may not call it that). For example, perhaps they decide they want to become more physically fit. First, they must define what “more physically fit” means to them. Then they measure their current fitness level and set goals for where they want to be. They may then decide to start walking half a mile every morning. Once they accomplish that for two weeks, they might increase the distance to one mile. After a month of walking one mile, they might increase the distance to a mile and a half, and so on. They measure their progress at regular intervals to determine where they are in relation to their goals. Once they meet their physical fitness goals, they select something else in their lives that they want to improve, and the improvement process begins again.

CSI for IT is just that: finding an IT area that is important to the business and looking for ways to improve it. ITIL defines CSI as “[a] stage in the lifecycle of a service. CSI ensures that you align services with changing business needs by identifying and implementing improvements to IT services that support business processes. You continually measure the performance of the IT service provider and then make improvements to processes, services, and infrastructure to increase efficiency and effectiveness.

8.15.1. Creating Service Quality Plan

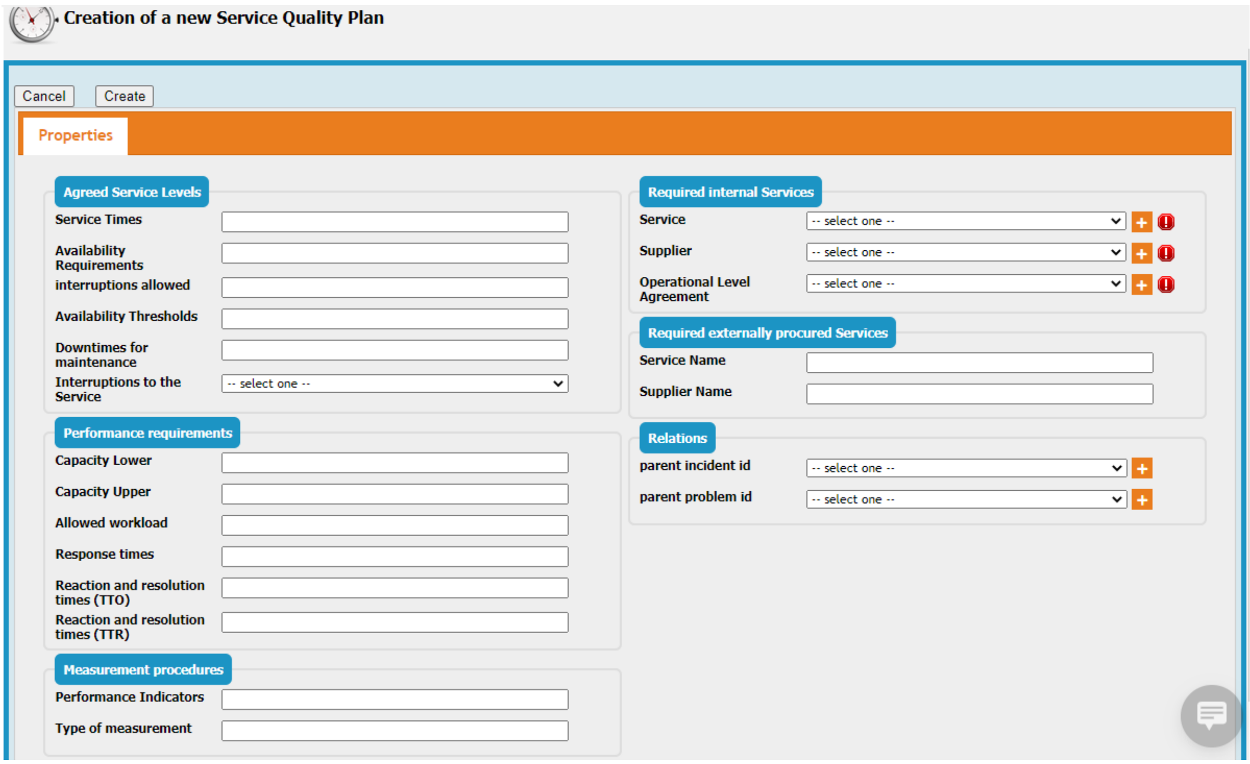
You can create a Service Quality Plan (SQP) using following methods:

- By viewing information such as incidents or problems related to services
- By creating a new record to link related incident or problem records

Administrators can add specific record types and forms such as service improvement plan or service quality plan to collect the information required by the business.

Steps to Create a new Service Quality Plan

- Click on the Service Quality Plan in the Service Level Management.
- Then the below dialog box appears:



Then click on Create.

8.16. Service Availability

According to ITIL, availability refers to the ability of a configuration item or IT service to perform its agreed function when required. It is calculated by using this equation:

$$Availability\ \% = \frac{(Agreed\ service\ time -\ Downtime)}{Agreed\ service\ time}$$

Agreed service time is the expected time the service will be in operation. If your service level specifies that users must have access to an ERP system from 6:00 AM to Midnight on workdays, your agreed service time is 18 hours/1,080 minutes/64,800 seconds per workday.

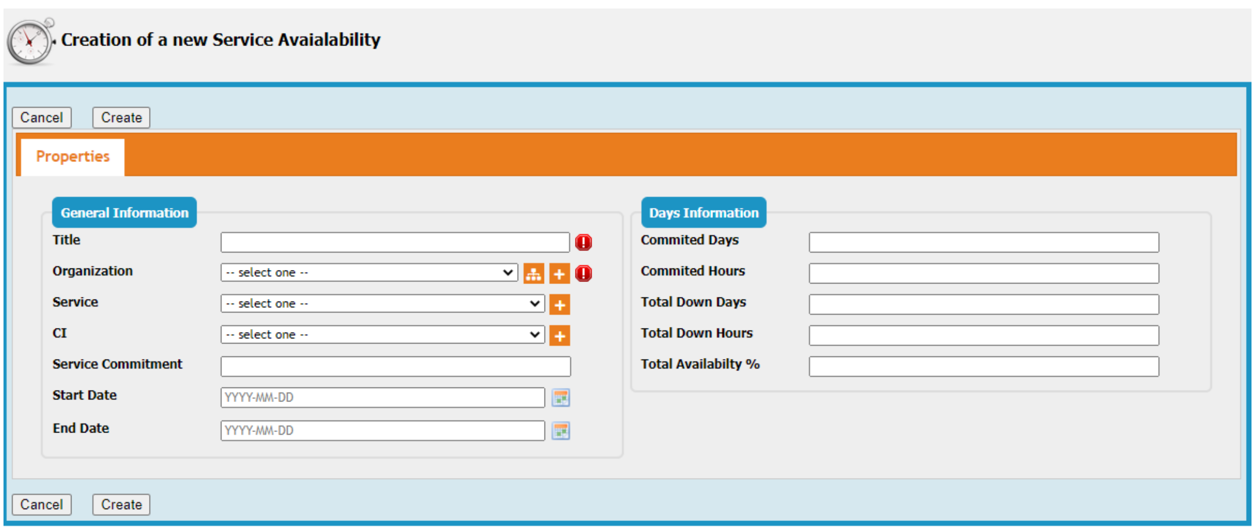
Downtime is the amount of time during the agreed service time that the service is not available.

Availability is measured as the percentage of time your service or configuration item is available. It reports on the past and estimates the future of a service. It tells you how well a service performed over the measurement period.

Availability metrics also estimate how well a service will perform in the future. In our ERP availability example, an average availability of 99.99% would predict we could expect an average uptime for our service of 17.9982 hours/1079.892 minutes/64,793.52 seconds per day. We can compare calculated against promised availability to determine if we are meeting our business goals.

8.16.1. Creating a new Service Availability

- Click on Service Availability from the Service Level Management Menu.
- Then, the below click on Create new Service Availability.
- The below dialog box will appear as:



Then click on Create.

8.16.2. Creating Service Improvement Plan

You can create a Service Improvement Plan (SIP) and link it to the continual service improvement register using following methods:

- By viewing information such as survey results
- By creating a new record to link related incident or problem records

Administrators can add specific record types and forms such as service improvement plan or service quality plan to collect the information required by the business.

9. Project Management

This simple version allows already project manager to manage:

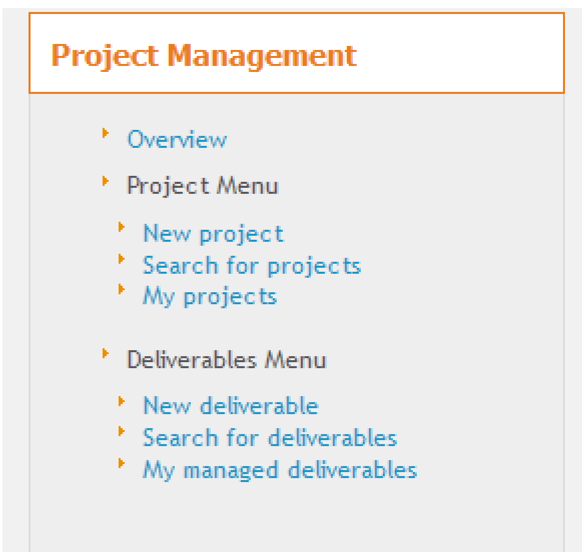
- **Projects**
- **Deliverables** with a Gantt View.
- **Stakeholders** involved on project

Project Manager will follow them through a **dashboard providing at a glance projects status & cost**.

Project Management capabilities can be limited to users needing it.

9.1. Usage

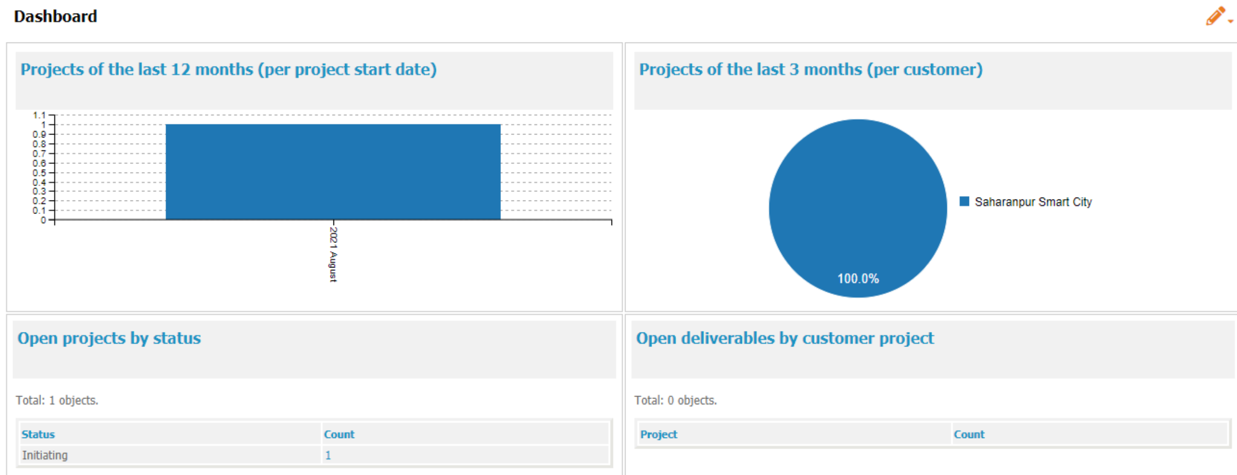
After installation, you will see Project Management group Menu as below:



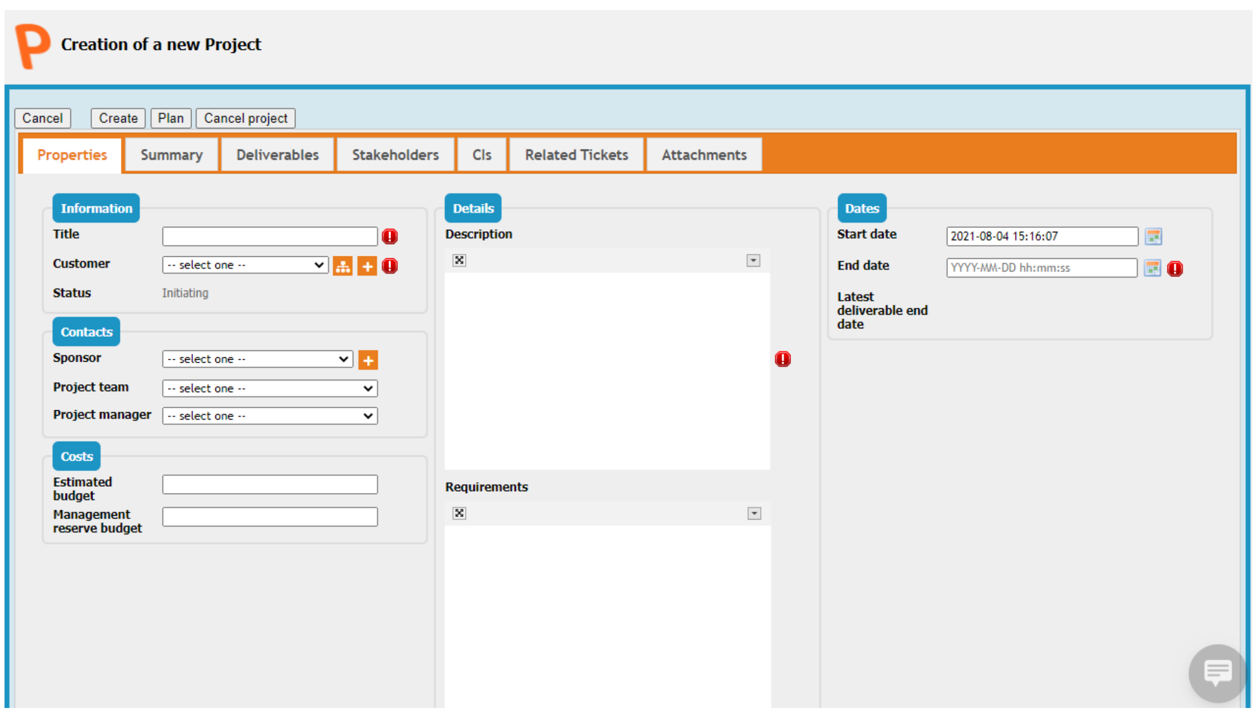
As Project Manager, you will

- create a project describing its goal, its scope, its constraints and its actors.
- then, you will start to divide your overall project on small deliverables.
- For each deliverable, you will assign a deliver owner who will define planning, costs, stakeholders.

Dashboard:



9.2. New Project



9.2.1. Project properties

A Project has the following attributes:

Field	Type	Comment	Mandatory?
Project Name	Alphanumeric string		Yes
Project for Customer	Foreign key to an Organization	Project is related to an organization	Yes

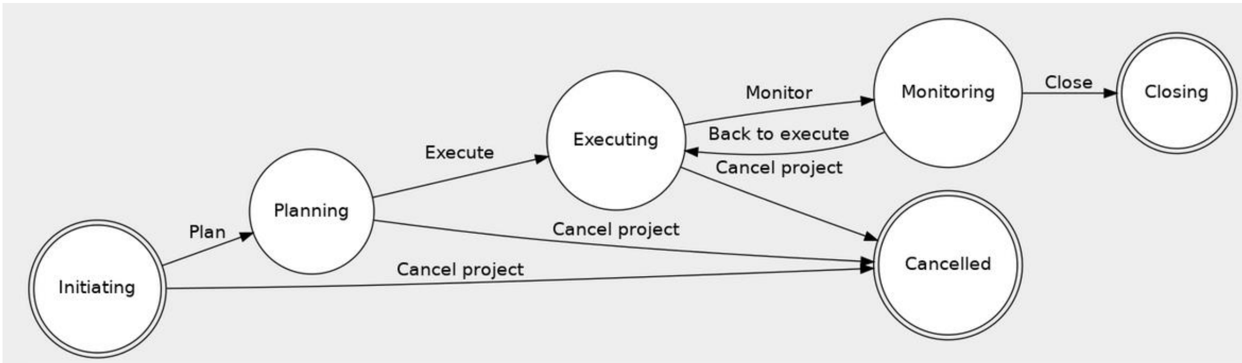
Status	Enum List	Possible values: Initiating, Planning, Executing, Monitoring, Closing	Yes
Project Description	Text (HTML)	description of project	No
Project Requirements	Text (HTML)	What is absolutely required to be done for this project	No
Project Constraints	Text (HTML)	Describe constraints that project will have to consider	No
Project Exclusions	Text (HTML)	Describe what is out of scope of your project	No
Project Sponsor	Foreign key to a Person	Authorizes the project	No
Project Manager	Foreign key to a Person	Executes the project	No
Estimated budget	Decimal	Amount of your budget to execute project	No
Management reserve budget	Decimal	Financial reserve of Sponsor to manage escalations	No
Deliverables costs	Decimal	Costs of all deliverables attached to this project (calculated)	Read-only
Project start date	Date (year-month-day)	Enter Start date of your project	Yes
Project end date	Date (year-month-day)	Enter planned end date of your project. This one can't be before start date	Yes
Revised end date	Date (year-month-day)	Enter revised end date of your project in case of rebase line. This one can't be before latest date of deliverable end date. if no date is entered, latest end date of deliverable is copied	No
Latest deliverable end date	Date (year-month-day)	Consider the latest deliverable's end date attached to this project	Read-only
Creation date	Date (year-month-day)	creation date of Project's ticket	Read-only
Last Update	Date (year-month-day)	last update date of Project's ticket	Read-only

9.2.2. Tabs

Tab	Description
Summary	Calculated Summary in one page of your project
Deliverables	All the deliverables belonging to this project
Stakeholders	All the contacts linked to this project with their interest, power, allocation in % and allocation in days
CI	All the CIs linked to this project, if applicable
Gantt	Gantt View of all deliverables attached to this project
Attachments	Documents related to this project

9.3. Project Lifecycle

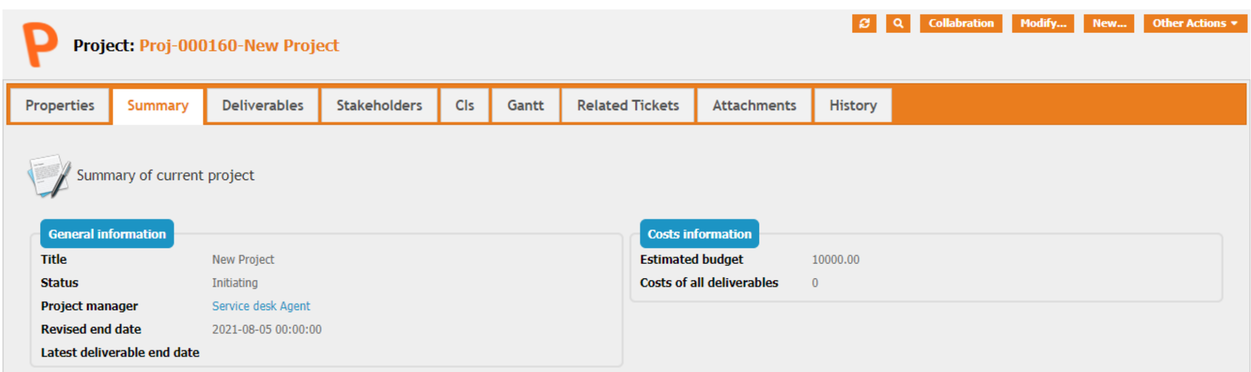
Project has lifecycle implemented to go through usual states of a project.



9.4. Project Summary

A calculated tab “Summary Project” is displayed near to properties tab. On this tab, you will find consolidated view:

- Project dates information
- Sum of all deliverable’s costs (even completed ones)
- list of involved people (either directly on project level or on deliverable level)
- list of opened deliverables
- Gantt View of project's deliverables



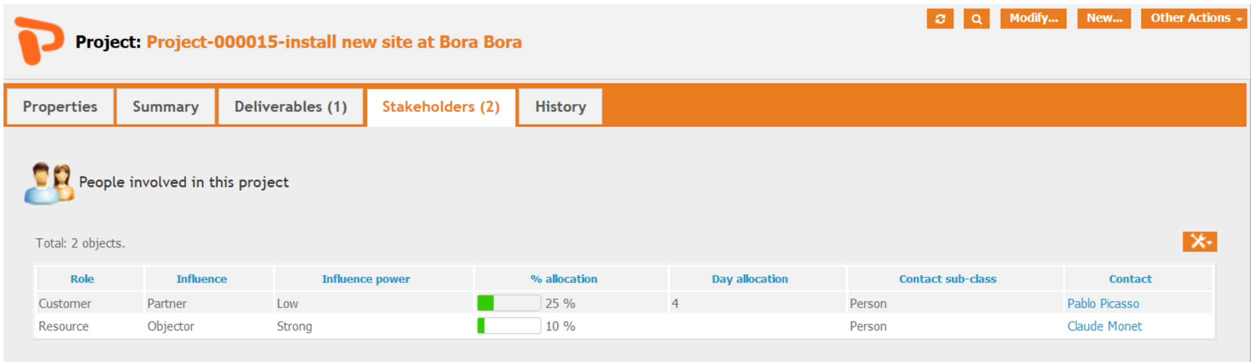
9.5. Project Stakeholders

In this tab, you will link all actors of project (customers or doers). You can define for each of them their

Role	Possible values: Customer, Resource
Influence	Possible values: Partner, Objector
Influence Power	Possible values: Low, Strong
% Allocation	provide a % of time resource or customer is allocated on project
Day allocation	provide number of days a resource or a customer is allocated on

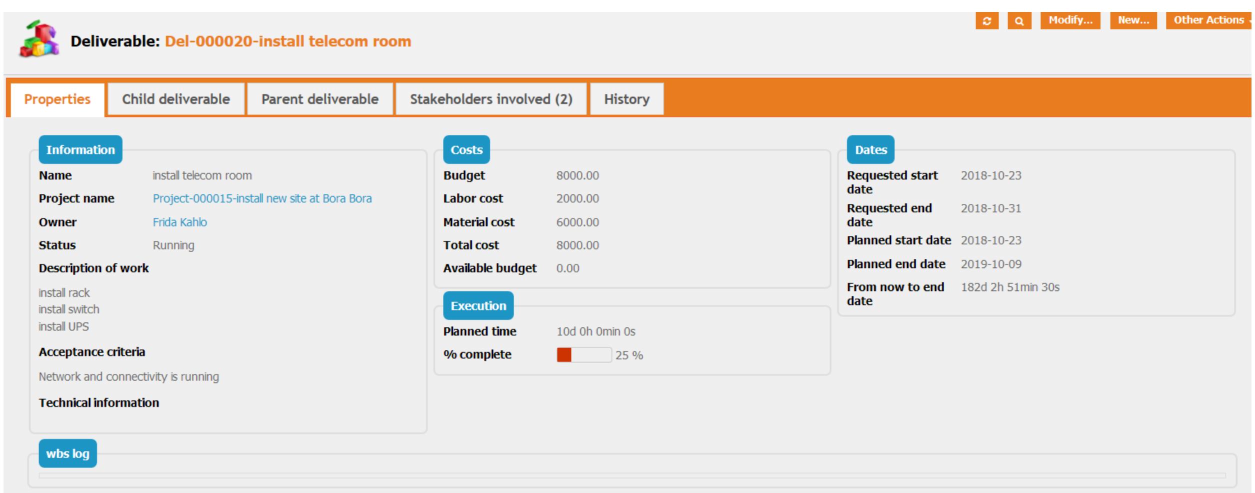
	project

This information will describe how you must manage and communicate with these stakeholders and plan the load of your resources.



9.6. Deliverables

To deliver medium to big project, you will split it in several deliverables which may have dependency between each other. Each deliverable will have costs, start date, end date and may be dependent between each other. In that case, you will declare parent deliverable.



9.6.1. Deliverable properties

Deliverable has following attributes:

Field	Type	Comment	Mandatory?
Deliverable Name	Alphanumeric string	Name of deliverable	Yes
Project name	Foreign key to a	Deliverable is linked to a project (directly or	Yes

	Project	not))	
Deliverable owner	Foreign key to a Person	Person who will execute the deliverable	No
Status	Enum List	Possible values: Running, Pending parent deliverable, Cancelled, Completed	No
Description of work	Text (HTML)	Describe the deliverable	No
Acceptance criteria	Text (HTML)	Describe the Acceptance criteria for resolution of deliverable	No
Technical information	Text (HTML)	Describe technical steps of the deliverable	No
Deliverable Budget	Decimal	Total budget to execute this deliverable	No
Labor cost	Decimal	Human and Software costs	No
Material cost	Decimal	Hardware costs	No
Deliverable Total cost	Decimal	Automatic sum of Labor and Material costs	Read-only
Available budget	Decimal	Amount of available budget (Deliverable Budget -Deliverable Total cost). Calculated on creation or update of deliverable	Read-only
Time planned	Duration	Effective time required for deliverable's execution	No
% Complete	Percentage	% of completion of deliverable	No
Requested start date	Date (year-month-day)	initial requested start date of deliverable	Yes
Requested end date	Date (year-month-day)	initial requested end date of deliverable	Yes
Planned start date	Date (year-month-day)	Reviewed start date of deliverable - Can't be before end date of parent deliverable. if you put a start date before parent's end date, it will automatically postpone child's “Planned start date” when you save ticket.	No
Planned end date	Date (year-month-day)	Reviewed end date of deliverable. If “Planned start date” has been postponed, “Planned end date” will be automatically postponed, as well, including initial delay between start and end date.	No
from Now to end date	Duration	Remaining time between now and re-estimated end date	Read-only

9.6.2. Deliverable automatic calculations

As soon as you update some attributes on deliverable, automatic calculations will run:

- if costs have been updated, “Deliverable Total cost” and “Available budget” will be recalculated
- if planned start or end date is updated, then all child deliverable's dates will be recalculated.

Tabs

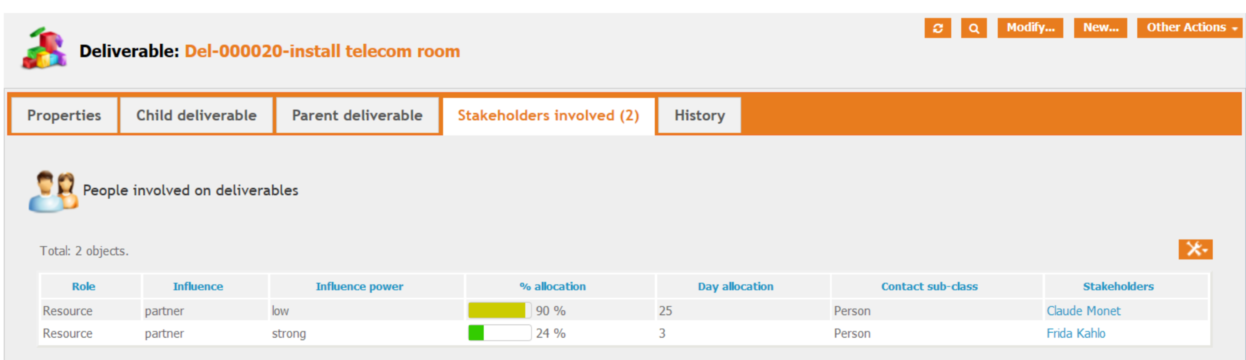
Tab	Description
Child Deliverable	Child deliverable - Child can't start before current deliverable's end date
Parent Deliverable	Parent deliverable - current deliverables can't start before end date of parents attached
Stakeholders involved	All the contacts linked to this deliverable with their interest, power, allocation in % and allocation in days

9.6.3. Deliverable Stakeholders

In this tab, you will link all actors of project (customers or doers). You can define for each of them:

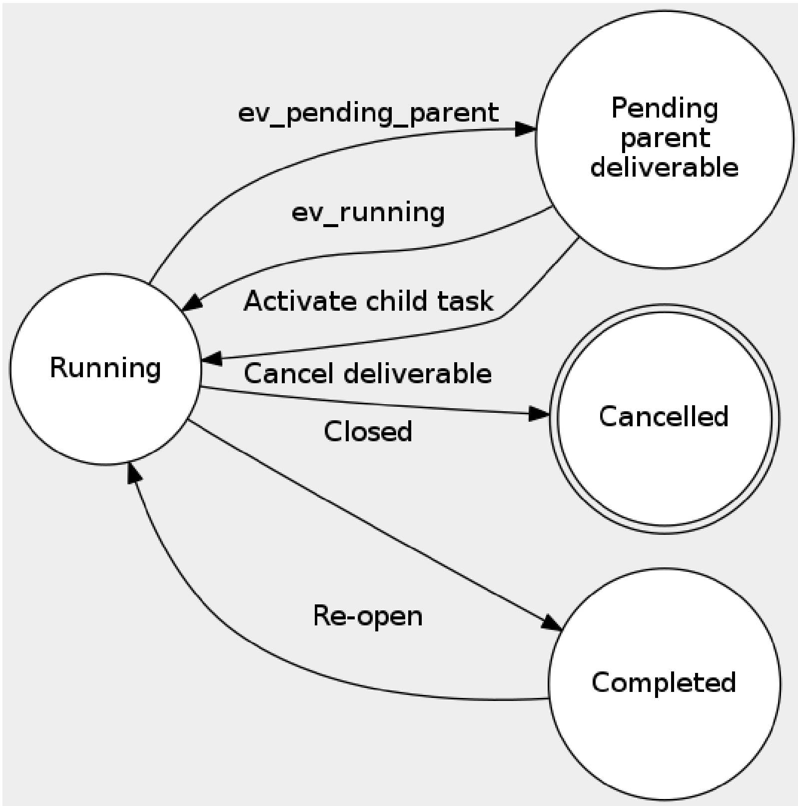
Role	Possible values: Customer, Resource
Influence	Possible values: Partner, Objector
Influence Power	Possible values: Low, Strong
% Allocation	provide a % of time resource or customer is allocated on project
Day allocation	provide number of days a resource or a customer is allocated on project

This information will describe how you must manage and communicate with these stakeholders and plan the load of your resources.



9.6.4. Deliverable Lifecycle

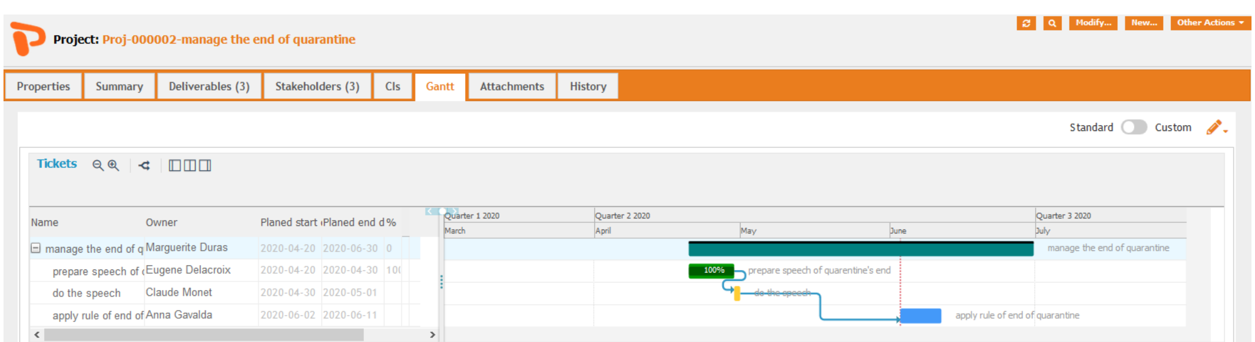
Deliverable has a simple lifecycle implemented to manage dependency between parent and child deliverable (child deliverable can't start before its parents' end).



As soon as deliverable has a parent, it will be created in status "pending parent deliverable" and will be re-activated in "running" status, automatically, as soon as parent deliverable is "completed".

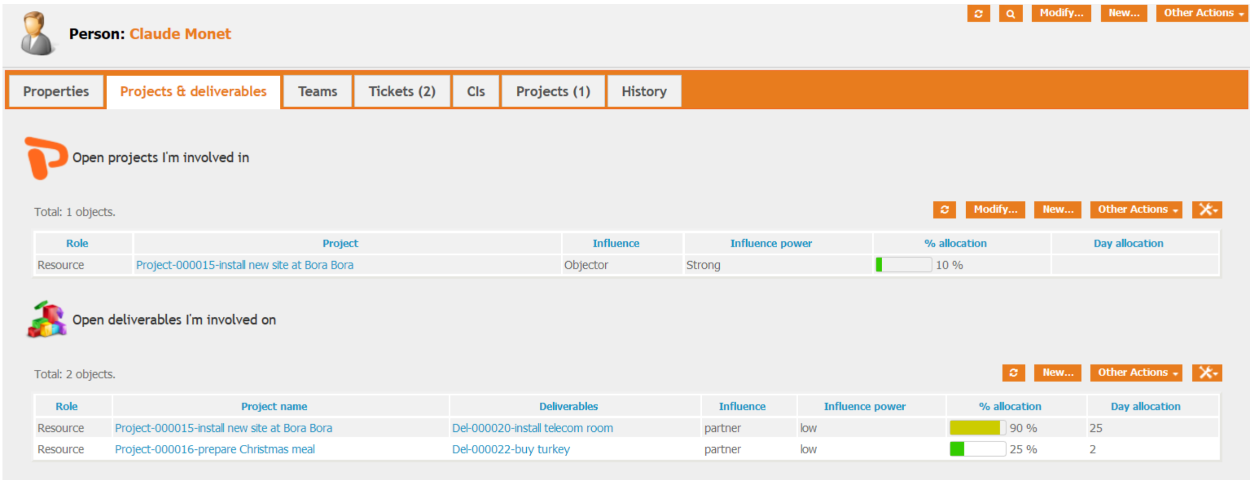
9.6.5. Gantt View

In Gantt tab, you will see graphically all deliverables of your project. In green, you will have the closed deliverables, in blue all running deliverables.



9.6.6. Stakeholder view from Contact Class

As soon as an tbITSM contact is involved on a project or on a deliverable, you will see appearing tab “Project& deliverable”. This calculated tab will display all projects and deliverable, the contact is involved on.



10. Knowledge Management

Knowledge management is the process of gathering, analyzing, storing, and sharing knowledge that is created within an IT service desk. It is designed to assist service desk teams to make the right decisions throughout the service life cycle and the incident resolution process by efficiently controlling and handling the flow of information.

ITIL defines knowledge management as the one central process responsible for providing knowledge to all other IT service management (ITSM) processes. It lays the groundwork needed for integrating knowledge management with all other processes in the ITSM framework.

10.1. Benefits of Knowledge Management

By implementing knowledge management practices in the workplace, IT managers can improve their teams and provide better service. Consider these benefits:

- Decreases the amount of time that must be spent on training employees. Since your team will constantly receive accurate and up-to-date information, they will require fewer formal training sessions to continue to meet and exceed performance expectations.

- Reduces the number of errors that are made by team members. If your team has the knowledge that they need to make decisions, they are far less likely to make mistakes.

- Reduces the need to complete steps in the service process more than once. When everyone is informed about the process, it is less likely that a step will be performed incorrectly. By implementing knowledge management, you give your team the power to reduce service process time by eliminating the need to repeat steps.

- Allows IT professionals to respond to customer needs faster and more effectively. Your employees will know how to answer customer questions more often, so you will not have to intervene or respond to negative feedback about team members' lack of knowledge.

10.2. Who uses Knowledge Management?

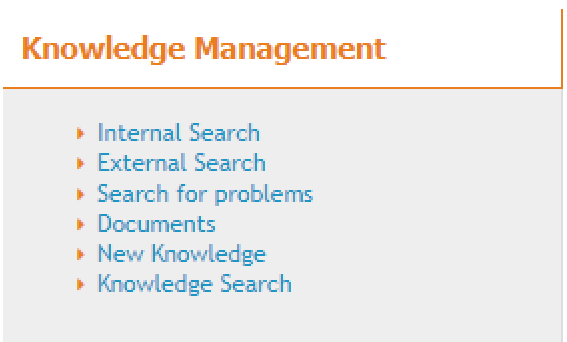
In one form or another, knowledge management is used by everyone in your organization. Your job as an IT manager is to ensure that data is properly processed into information, which is stored in a location that is accessible to the people who require specific knowledge to successfully complete job tasks. Your team will run more efficiently if you use knowledge management properly.

At your team's level, knowledge management creates knowledge and access to this knowledge when necessary. All of your team members will be responsible for identifying the need to translate information into knowledge, while you will need to manage the databases and manuals that are available to your team members.

Outside of your IT team, knowledge needs to be available to users through self-service portals. To improve the efficiency of your team, basic knowledge should be available to users who are willing to troubleshoot their own problems. When users can find solutions on their own, your team has opportunity to focus on escalated issues that require special skills and knowledge that is not available to those who are not familiar with IT applications, processes, and knowledge.

10.3. Usage

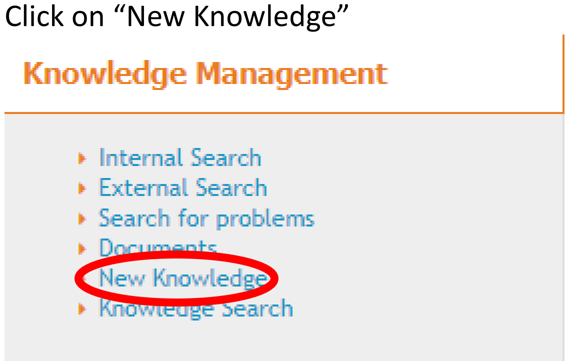
After installation, you will see Knowledge Management group Menu as below:



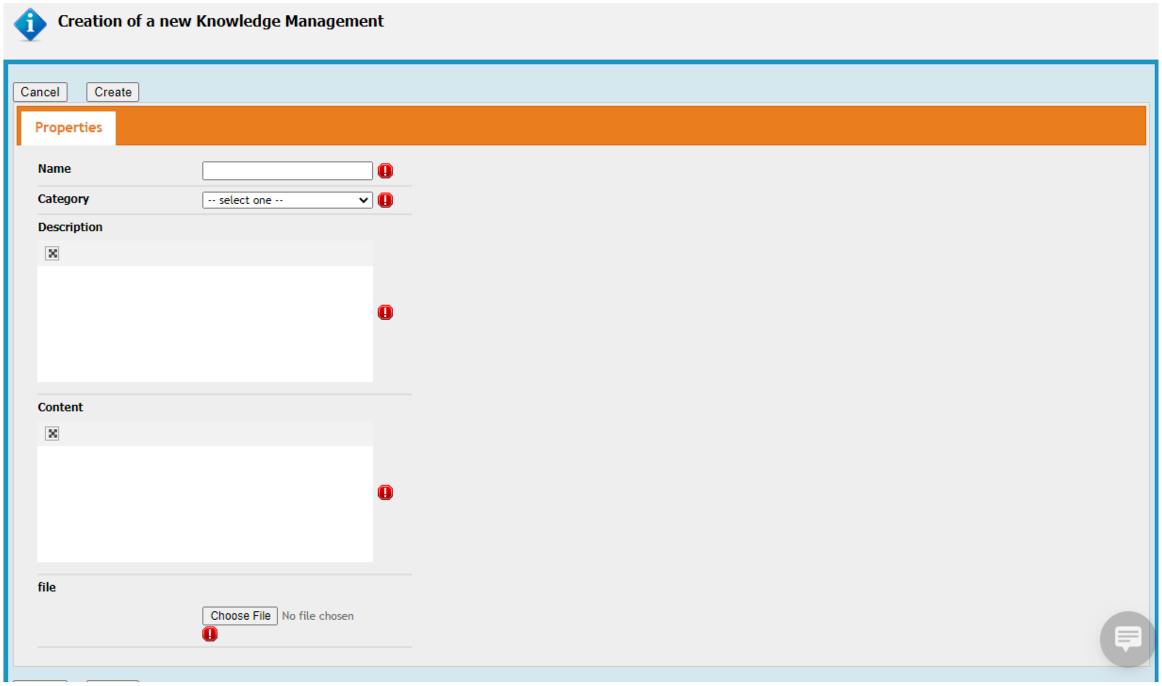
In Knowledge Management, you can easily gather, analyze, store and share knowledge within the Organization.

10.3.1. Creating a New Knowledge

Click on "New Knowledge"



Then fill the values in the given fields, then click on Create.



10.3.2. Internal Search

It is used to search internally within the Organization.

10.3.3. External Search

It is used to search externally/globally.

10.3.4. Documents

The documents that are being uploaded in the portal are shown here.

11. Approval Process Automation

This extension provides the capability to handle a two levels approval process for a user request, with several approvers in parallel on each level.

11.1. Features

- Automated, rule based, approval mechanism
- Two levels of approbation with several approvers in parallel at each level
- Approvers can approve or reject a request in one click from an email
- Passive or active approval
- Configurable timeout delay
- Graphical view of the approval status on each ticket
- Configurable approval email message, with multi-language capabilities.

11.2. Configuration

The following settings are available to configure the module:

Module	Parameter	Type	Description	Default Value
approval-base	email sender	string	Sender email address, as seen in the approval email. If left blank, sending the email will fail.	
approval-base	email_reply_to	string	Default "reply to" email address for the approval email. If empty defaults to email sender	
approval-base	comment_attcode	string	Attribute into which the user comments will be reported. Can be a case log or text. The comments are all aggregated. Note: the comment can also be viewed as a tooltip.	(optional)
approval-base	list_last_first	Boolean	In case several executions occur, drives the order in which the results of the executions are displayed (vertically).	false
approval-base	enable_reminder	Boolean	Enable the feature "send a reminder".	true
approval-extended	enable-admin-abort	Boolean	Allow defined profiles to bypass the approval process. Profiles allowed to bypass the process are defined in the variable bypass_profiles.	true
approval-extended	target_state	string	state that may trigger the approval process. The	new

			event ev_wait_for_approval must exist on this state, ending on wait_for_approval state.	
approval-extended	bypass_profiles	string	CSV list of profiles. Having any of the given profiles is sufficient to be allowed to bypass approval processes. Set to an empty string to deny the feature to anybody.	Administrator, Service Manager
approval-extended	reuse_previous_answers	Boolean	If a person is approver at both levels, then his level 1 decision is reused at level 2	true

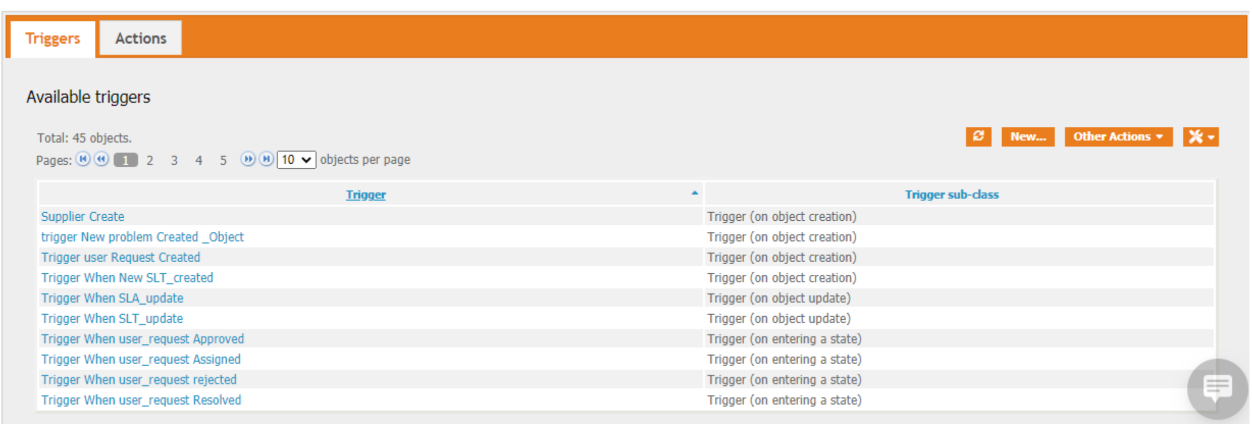
The following standard settings might be of interest when setting up the approval feature:

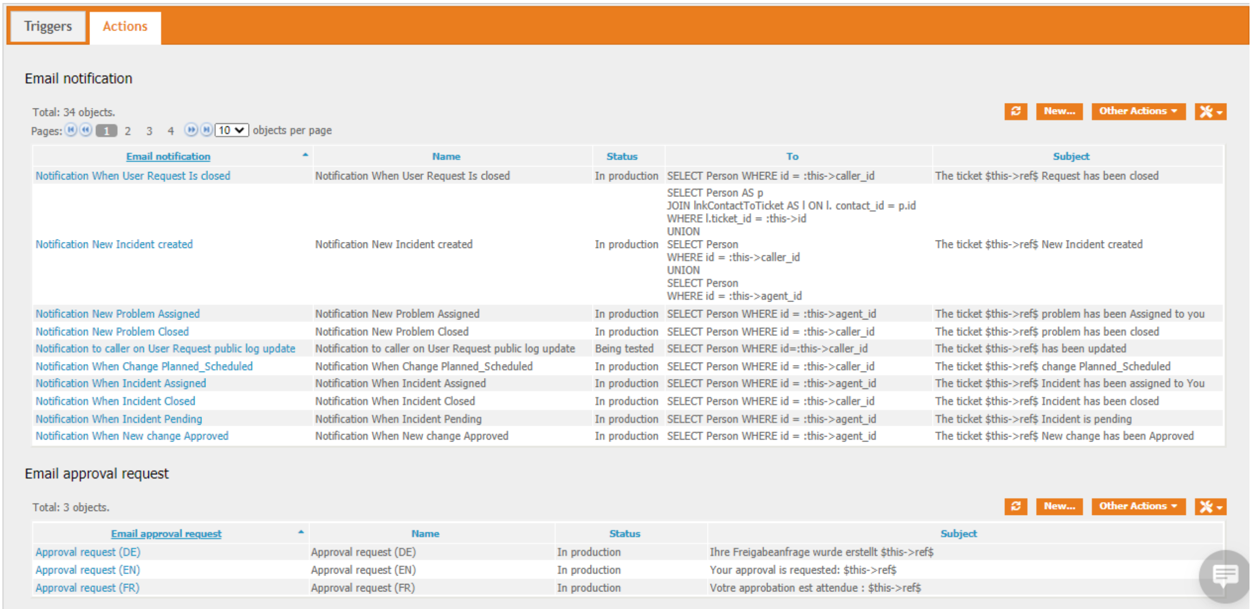
- email_asynchronous
- email_transport

11.3. Notification(trigger/action)

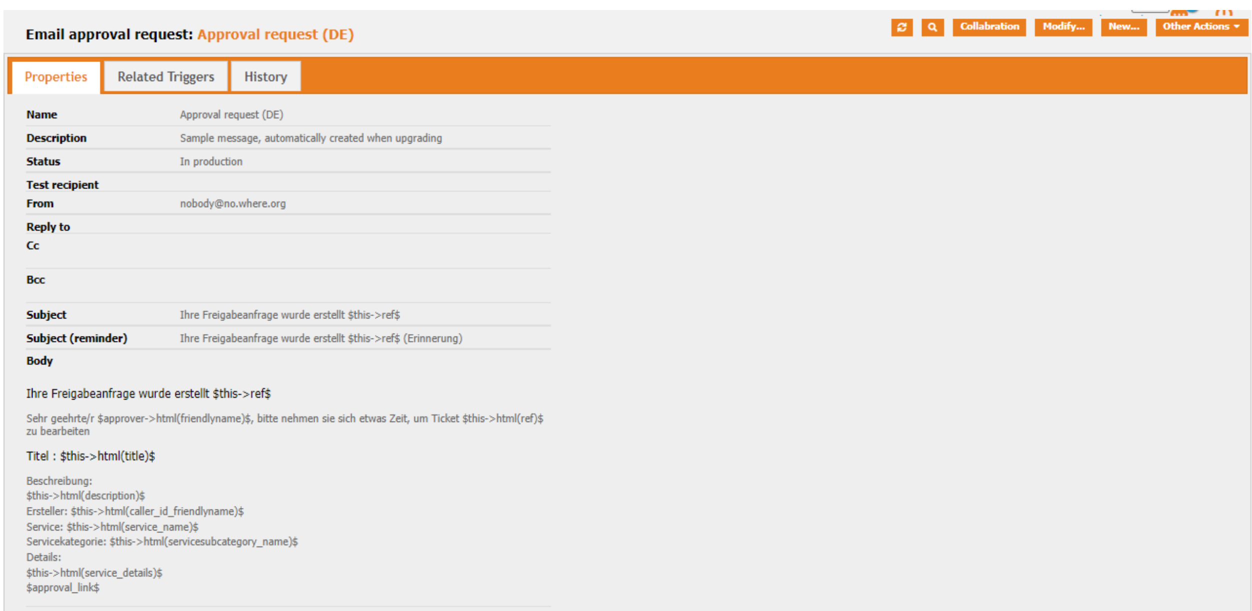
Email notification is based on Trigger/Action and email content can be tailored to your need with HTML format and placeholders.

A default trigger is created at installation

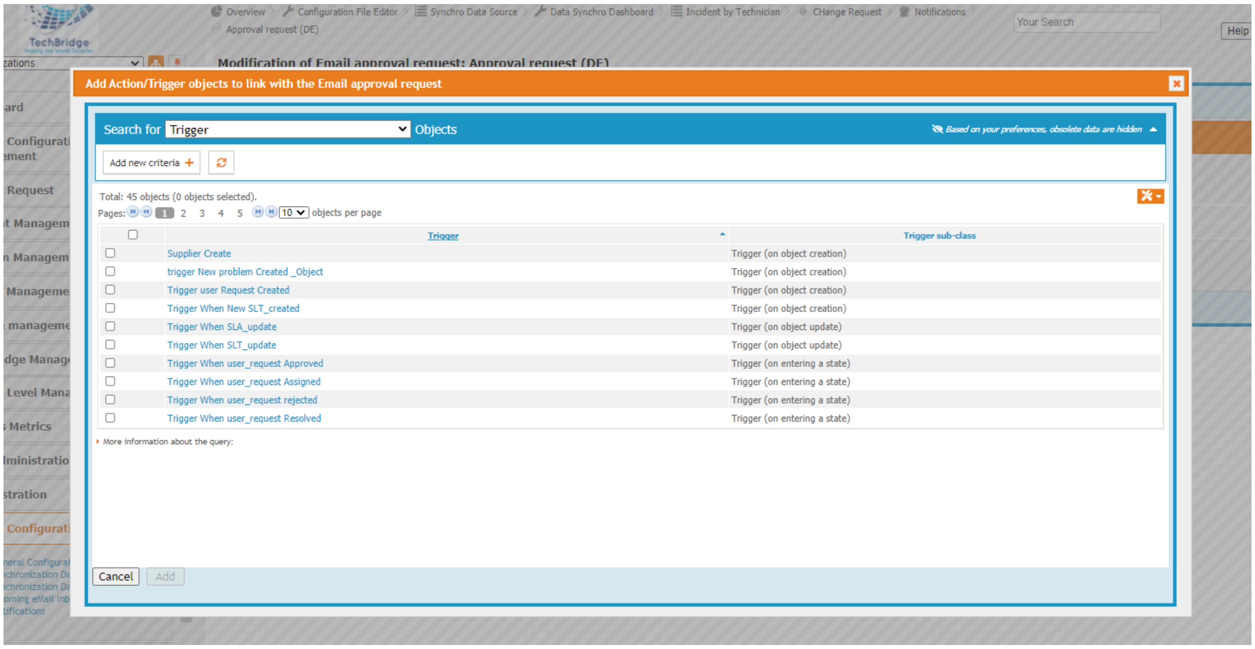




You can of course edit this message to make it yours, here is the English default version for example of possible placeholders:



You must **create the linkage between trigger and action** of the language you want to use.



11.4. User Experience

11.4.1. Cinematics

When a User Request is entering the state *new*, the approval engine verifies if there is an approval rule defined for the corresponding service subcategory. If yes, then the state of the user request is set to *wait for approval* and a notification is sent to the approvers defined in the approval rule. Only the approvers corresponding to the first level are notified. Once the request is approved, and if a second level has been defined, then the second level approvers are notified. Should the approval be rejected (by an approver, or on timeout), then the process finishes in any case.

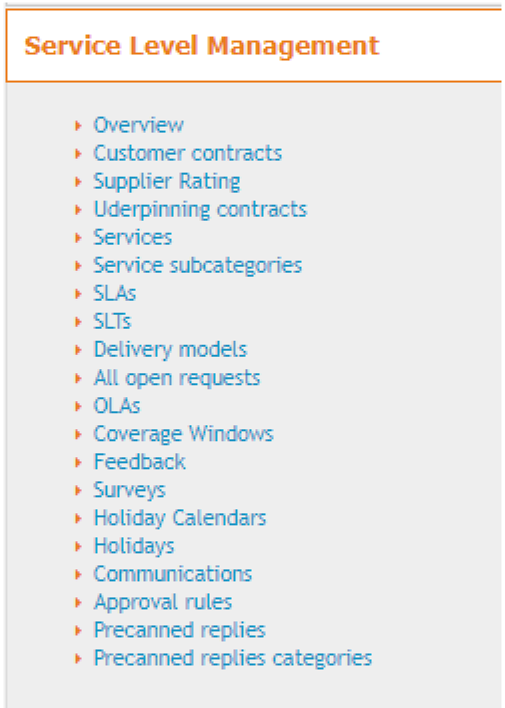
The email contains a web link to display the web page to approve or reject the request.

The approvers will have a delay defined in the approval rule to give their answers. This delay takes into account the coverage window defined in the approval rule.

Once the approval delay has expired, the approval process terminates. The result (approved or rejected) is then taken in the property *Automatically approved if no answer at Level 1/2* of the approval rule.

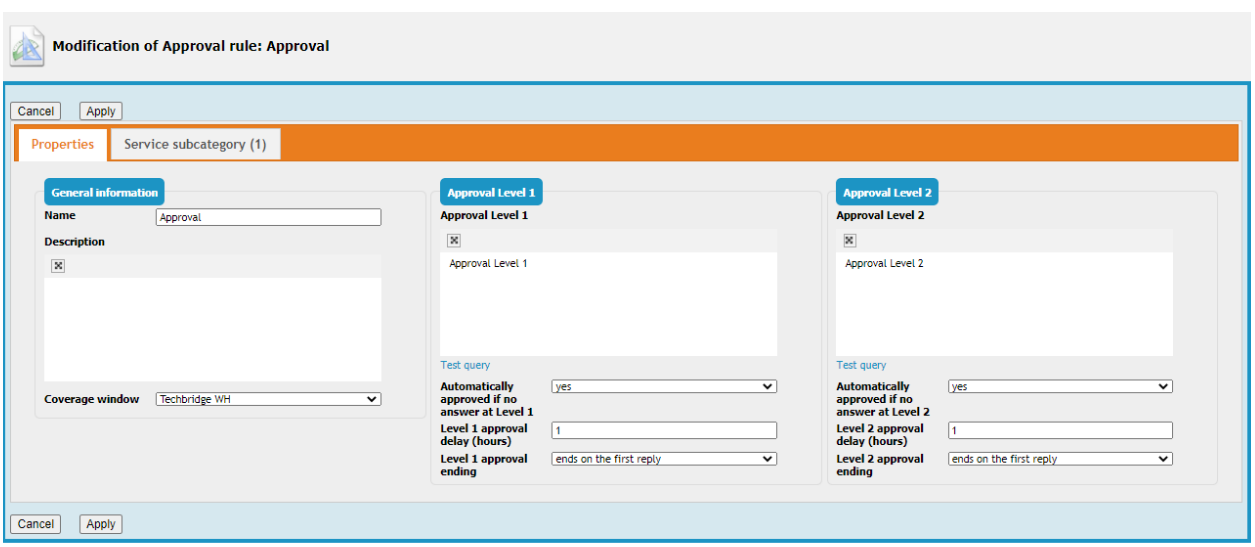
The User Request will then continue its way through its lifecycle, depending on the approval status: rejected or approved.

11.4.2. Create approval rules



From the Service Management menu, click on Approval rules:

The pages show a list of already defined approval rules. Click on the button “new” to create a new one:



An approval rule is identifier by its **name**. The **coverage window** is used to compute the approval delay.

The fields **Approval level** 1 and 2 define, via an OQL query, the list of approvers for each approval level. These queries must return elements having an email attribute (e.g., Person or Team). It is possible to use the place holders: this->attribute that make reference to the attributes of the user request.

For instance: this->caller_id for the caller: this->service_id for the service … All attributes defined in the data model for a service request can be used.

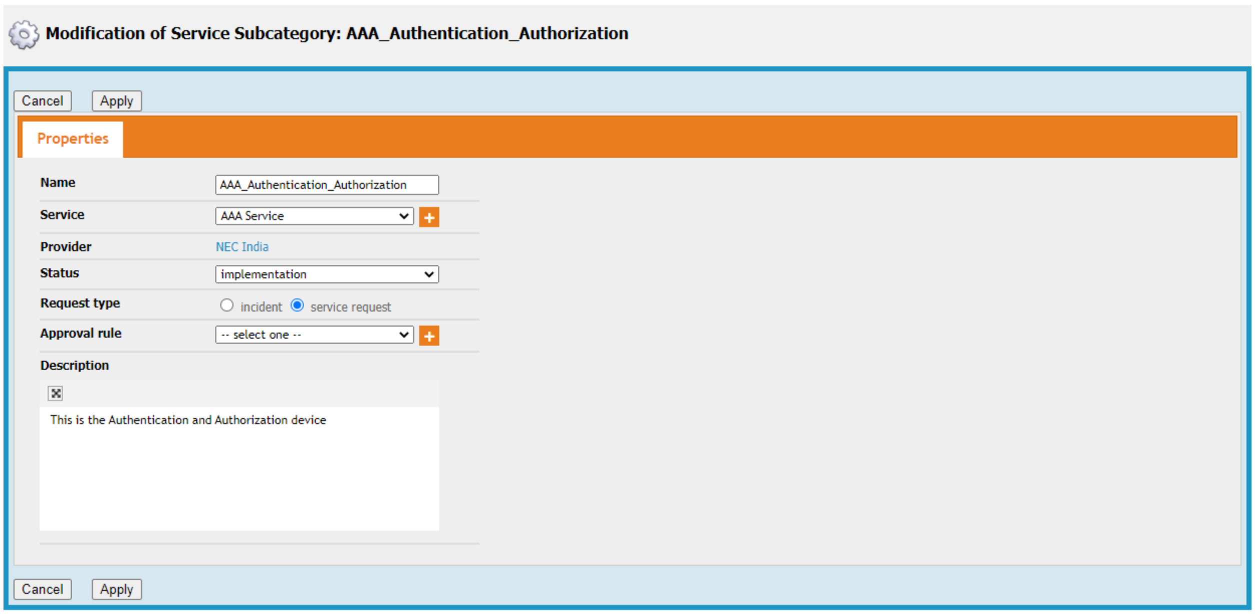
The field **Automatically approved if no answer** determines if the request will be approved or rejected if there is no answer after the defined delay.

The field **approval delay (hours)** defines the delay in hours for each level of approval, only hours within the coverage windows are counted. If you leave it empty or set it to 0, User Request will stay forever in waiting for approval state, until one of the approvers accept or reject it.

The field **approval ending** is useful with multiple approvers to determine, how to behave with multiple actions:

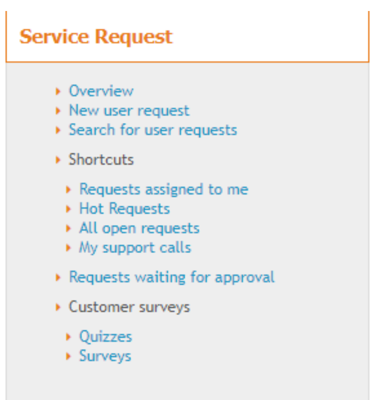
- **ends on first approve***:* All must reject for Request to be rejected.
  - Multiple rejected then one approved ⇒ ends and approved
  - multiple rejects but not all then delay expired ⇒ ends and "reject or approved" depends on field Automatically approved if no answer
- **ends on first reject *previous default behavior*****:** all must approve.
  - Multiple approval then one reject ⇒ ends and reject
  - multiple approvals but not all then delay expired ⇒ ends and "reject or approved" depends on field Automatically approved if no answer
- **ends on first reply**: first to act is the decision maker.

Once the approval rule has been defined, you can assign it to a service subcategory, either from the approval rule itself (tab *Service subcategory*) or from the service subcategory:

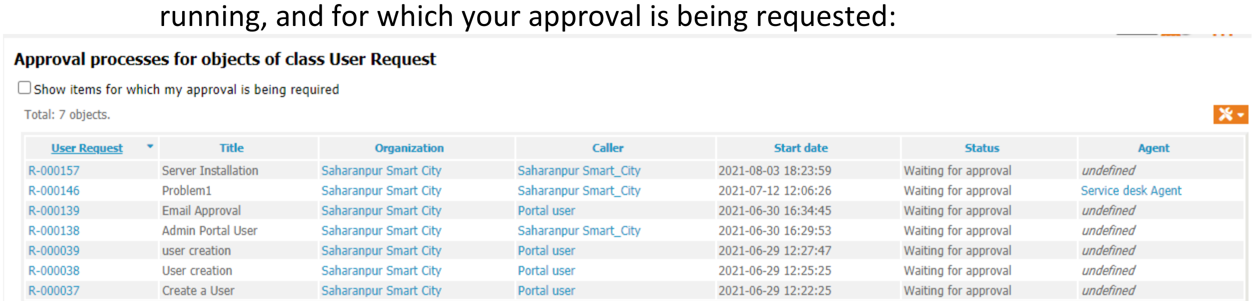


11.4.3. Approving process

Approver with an tbITSM login can connect anytime and check if they have Requests pending their approval. From the Service Request menu, click on "Requests waiting for approval":



The page shows a list of the User Requests having an approval process running, and for which your approval is being requested:


Approval processes for objects of class User Request

Show items for which my approval is being required

Total: 7 objects.

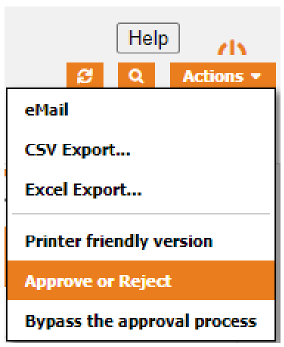
User Request	Title	Organization	Caller	Start date	Status	Agent
R-000157	Server Installation	Saharanpur Smart City	Saharanpur Smart_City	2021-08-03 18:23:59	Waiting for approval	*undefined*
R-000146	Problem1	Saharanpur Smart City	Saharanpur Smart_City	2021-07-12 12:06:26	Waiting for approval	Service desk Agent
R-000139	Email Approval	Saharanpur Smart City	Portal user	2021-06-30 16:34:45	Waiting for approval	*undefined*
R-000138	Admin Portal User	Saharanpur Smart City	Saharanpur Smart_City	2021-06-30 16:29:53	Waiting for approval	*undefined*
R-000039	user creation	Saharanpur Smart City	Portal user	2021-06-29 12:27:47	Waiting for approval	*undefined*
R-000038	User creation	Saharanpur Smart City	Portal user	2021-06-29 12:25:25	Waiting for approval	*undefined*
R-000037	Create a User	Saharanpur Smart City	Portal user	2021-06-29 12:22:25	Waiting for approval	*undefined*

Approver without tbITSM login, can still approve or reject Request, but they must have received the notification for this. The email provided link allows him to approve or reject without tbITSM login.

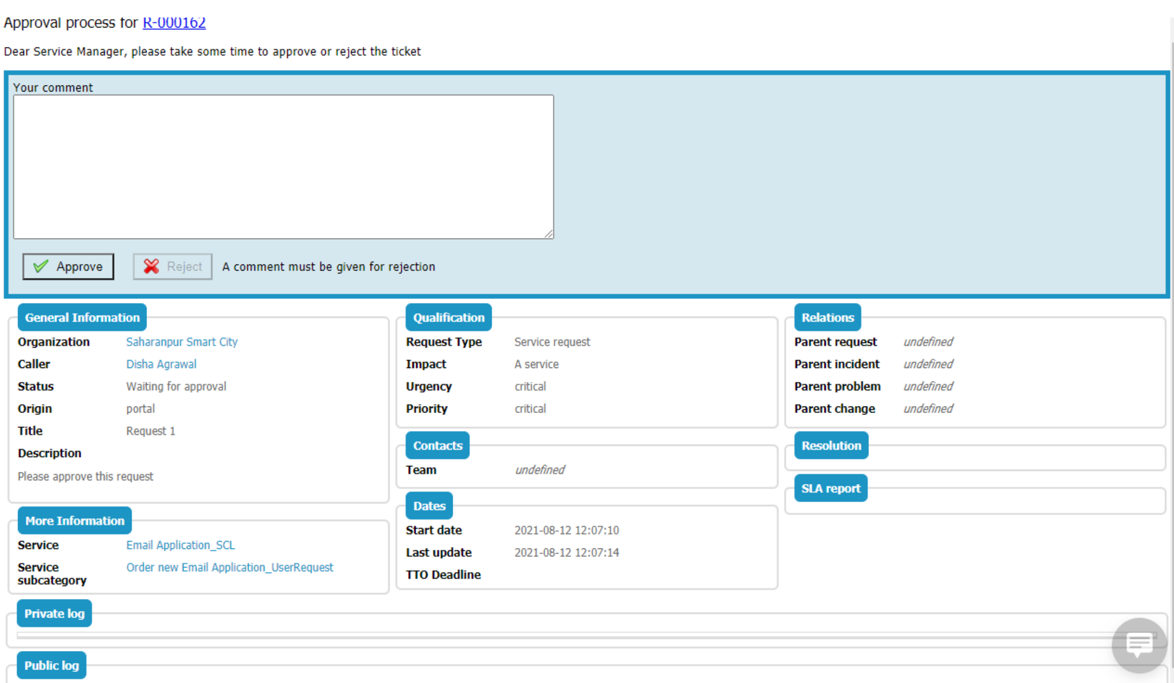
If the approver has an tbITSM login then the email provided link points him to tbITSM (Console or Portal depending on his access). In that case, the link provided in the email can only be used with the approver tbITSM login.

11.4.4. Approve or reject

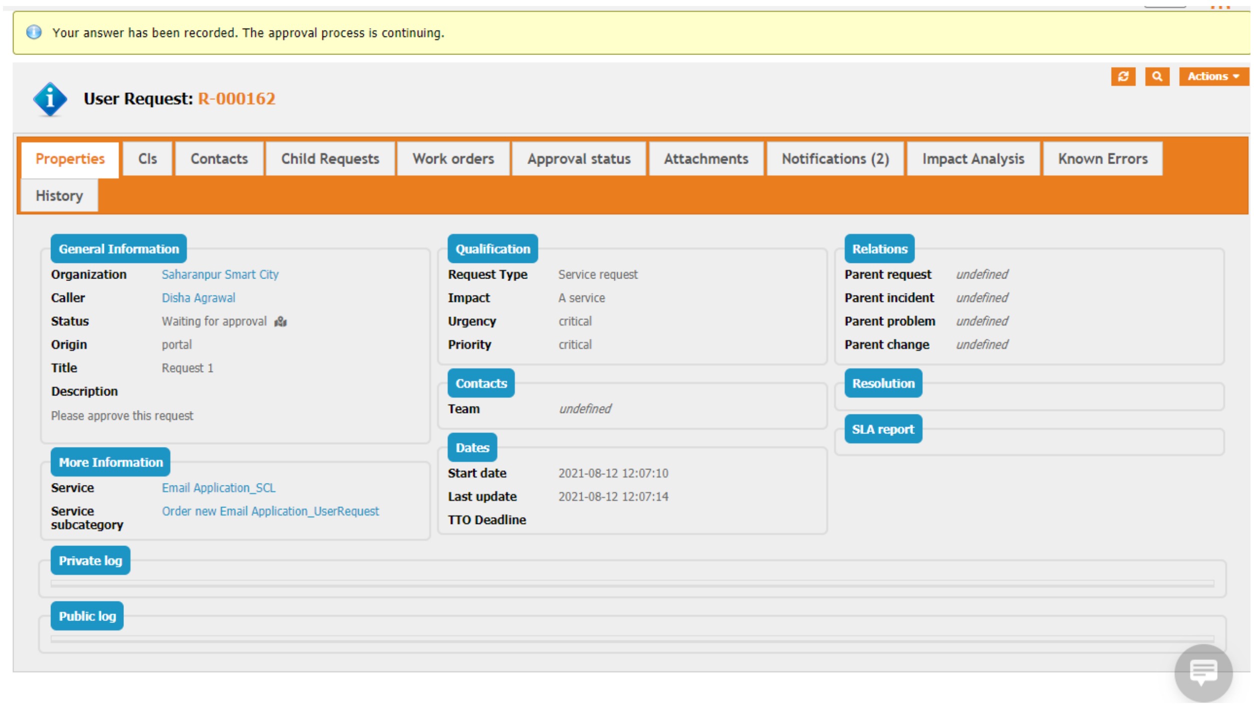
From the user request, open the **Other actions** menu and select **Approve or Reject**:



The approval form is displayed:

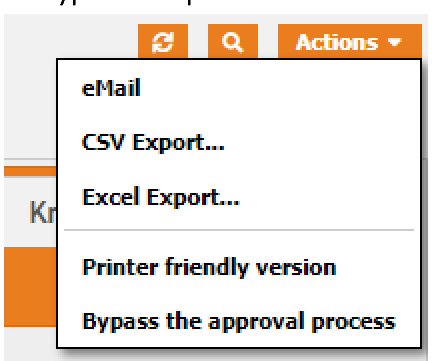


After the reply has been given, you are redirected to the user request and a banner reminds you the outcome of your reply.

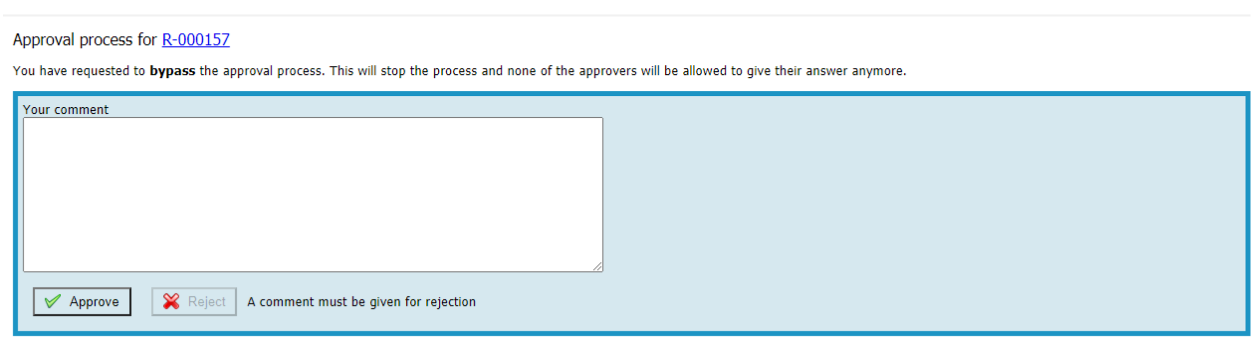


11.4.5. Bypass the approval process

If you are an administrator, and if the setup allows it, then you have a menu to bypass the process:

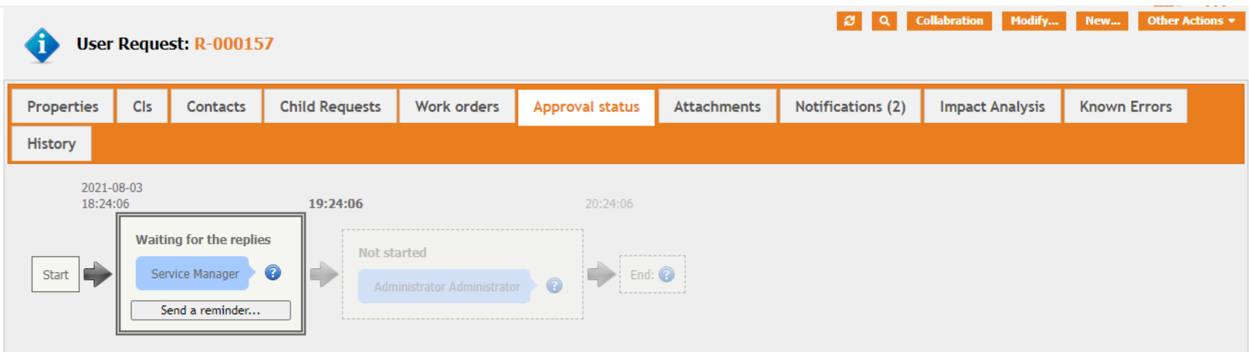


The approval form is then a little different than the standard reply form: it reminds you that bypassing the process is a little different.



11.4.6. Status

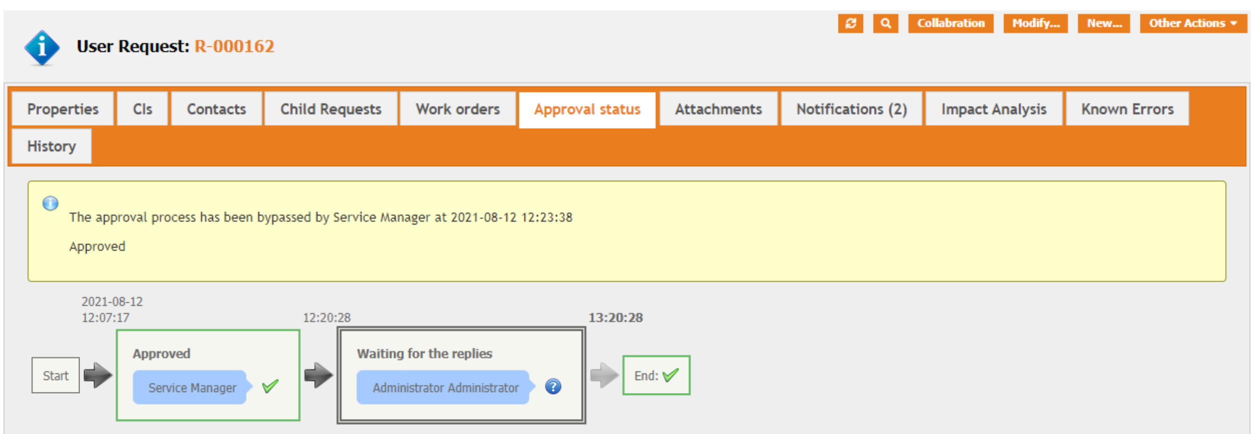
As soon as a user request has been through an approval process, the tab **Approval status** shows detailed information about the ongoing or terminated approval.



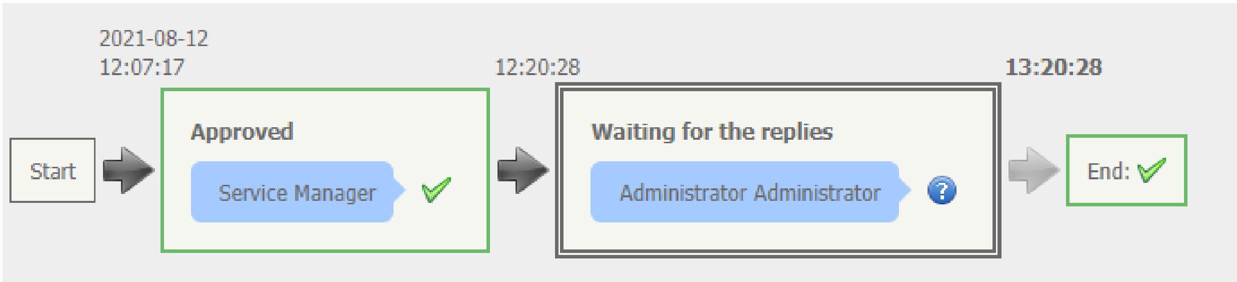
In the above example, the deadline is displayed in bold: 3rd of August at 19:24:06.

Click on the button "send a reminder" to send a new message to the approver (confirmation required). This feature can be disabled by setting the parameter *enable_reminder* to false.

After the reply has been given, the status appears in clear:



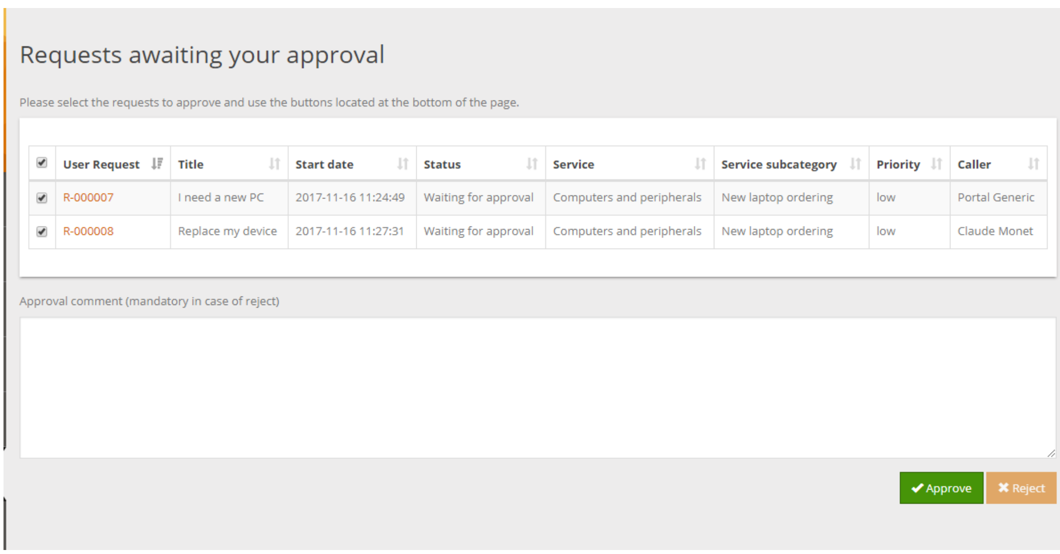
Move your mouse over the image next to the approver's name, and you will get the date of the answer and her comment if any has been given:



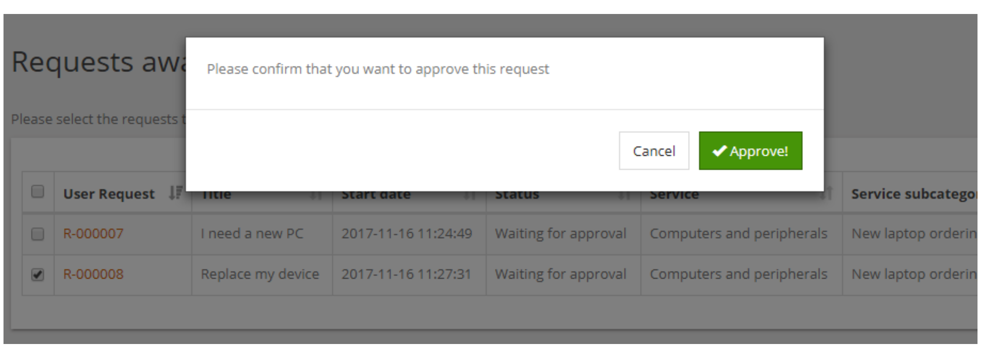
The status will be entirely reset anytime the user request enters the state "waiting for approval".

11.4.7. Approval in portals

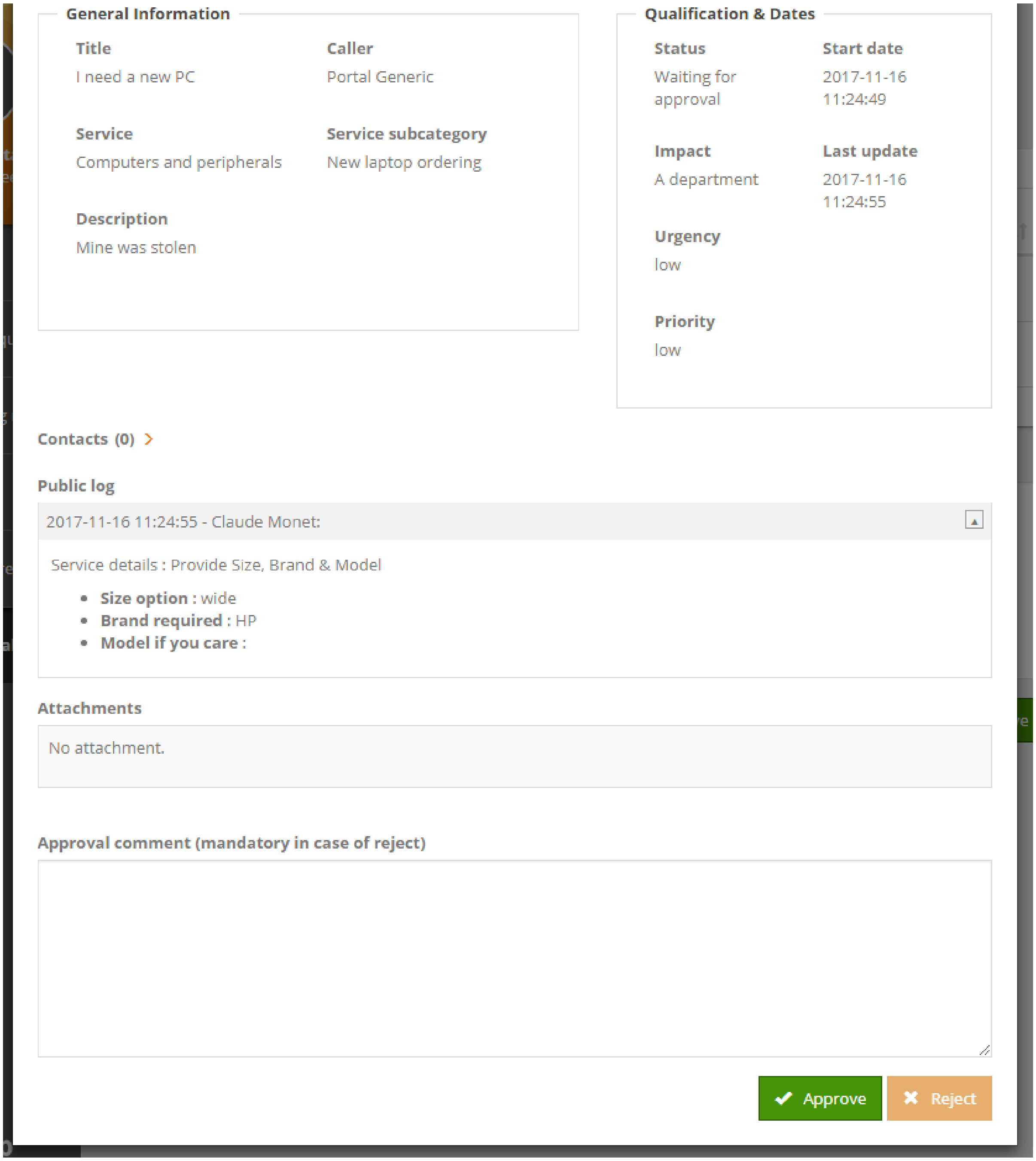
A new menu appears in the User Portal, which allow an approver to retrieve all User Requests waiting for her approval, and one by one or in bulk mode, accept or reject them.



A comment is required in case of rejection, so the button is disabled as long as the comment is empty. A confirmation is required in both cases.



When clicking on the link to a User Request, the ticket's details are displayed with an extra comment field and two buttons at the bottom of the details to accept or reject it:



12. Approval Process Light

This module provides the capability to handle a simple approval process for a user request. It notifies a selected approver by email in order to approve or reject a user request. This approver does not need a login in tbITSM to approve the request.

Only one level of approbation is supported by this module, and the approver has to be selected manually by a support agent.

12.1. Features

- Simple approval mechanism with one approver per ticket
- Approvers can approve or reject a request in one click (no need to have an tbITSM account)
- Tickets waiting for Approval and bulk approval available in Portal.
- Passive or active approval
- Configurable timeout delay
- Graphical view of the approval status on each ticket

12.2. Configuration

After installing this module, configure a proper email_sender and configure trigger/action to ensure “Approval emails” delivery.

The following settings are available to configure the module:

Module	Parameter	Type	Description	Default Value
approval-base	email_sender	string	Sender email address, as seen in the approval email. If left blank, sending the email will likely fail.	
approval-base	email_reply_to	string	Default “reply to” email address for the approval email.	(optional) defaults to email_sender
approval-base	comment_attcode	string	Attribute into which the user comments will be reported. Can be a case log or text. The comments are all aggregated. Note: the comment can also be viewed as a tooltip.	(optional)
approval-base	list_last_first	Boolean	In case several executions occur, drives the order in which the executions are displayed (vertically).	false

approval-base	enable_reminder	Boolean	Enable the feature “send a reminder”.	true
approval-light	approval_timeout_delay	int	Delay to get the answers given in days. Use 0 to disable the timeout (= infinite duration to approve or reject the request). Note the first negative answer marks the request as rejected without waiting for the further answers.	5
approval-light	approve_on_timeout	Boolean	Set to true for a passive approval scheme, false for an active approval scheme.	false
approval-light	approver_select	string	OQL to display the possible approvers (must define a set of objects derived from the class Contact). Use: this->*att_code* to add conditions based on the user request's properties.	SELECT Person AS p WHERE id =: this->org_id
approval-light	bypass_profiles	string	CSV list of profiles. Having any of the given profiles is sufficient to be allowed to bypass approval processes. Set to an empty string to deny the feature to anybody.	Administrator, Service Manager

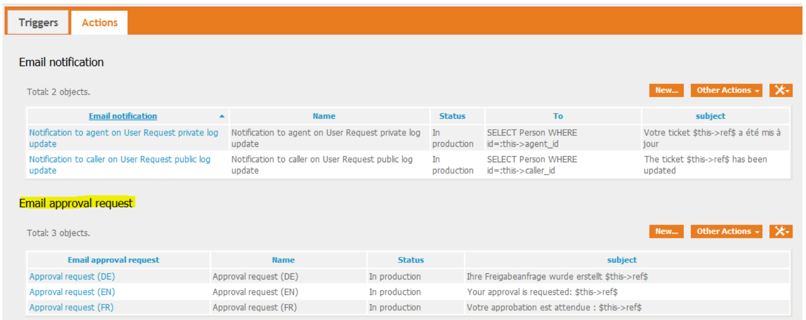
The following standard settings might be of interest when setting up the approval feature:

- email_asynchronous
- email_transport

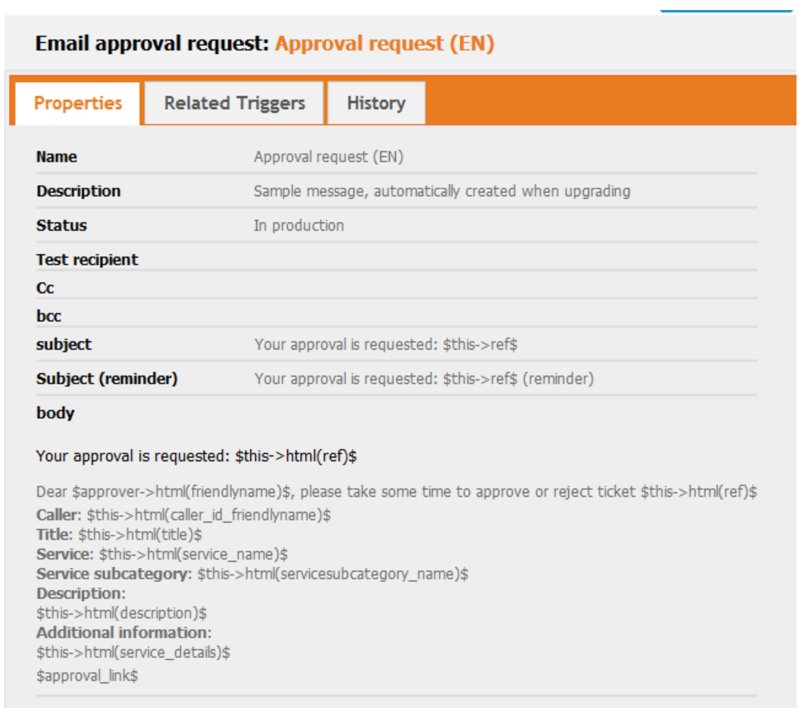
12.3. Notification (trigger/action)

Email notification is based on Trigger/Action and email content can be tailored to your need with HTML format and placeholders.

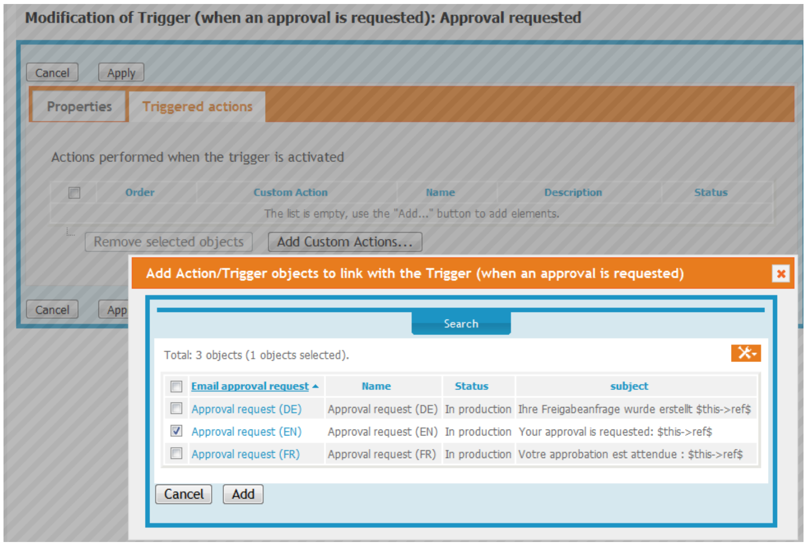
A default trigger is created at installation



You can of course edit this message to make it yours, here is the English default version for example of possible placeholders:



You must **create the linkage between trigger and action** of the language you want to use.



You can create your own trigger and action.

If you need to send different notification depending on the organization of the caller, the service, the service family, or any data available on the Ticket, this can be done by creating multiple trigger/action couples, using a filter on the Trigger.

12.4. Usage

12.4.1. Cinematics

A specific action *Wait for approval* is available on all User Requests in state *New*.

When the user selects this action, she will be prompted to select the Approver contact.

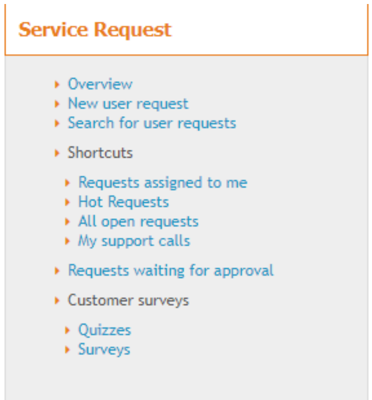
Then the User Request enters the state *Waiting for approval* and a notification is sent to the Approver. The Approver can approve or reject the request by clicking on the link provided in the email (an tbITSM login is not mandatory for this action). Alternatively, she can approve or reject from within tbITSM.

If there is no answer within 5 days (configurable), the answer defaults to *Rejected* (configurable).

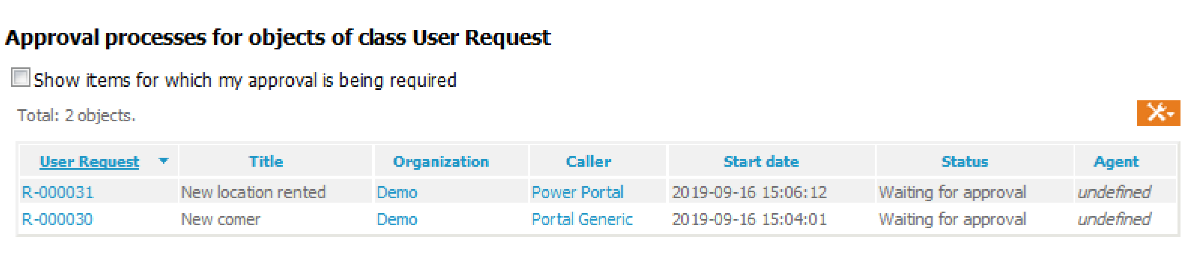
The User Request will then continue its way through its lifecycle, depending on the approval status: *Rejected* or *Approved*.

My ongoing approvals

From the Service Request menu, click on “Requests waiting for approval”:



The page shows a list of the User Requests having an approval process running, and for which your approval is being requested:



Approval processes for objects of class User Request

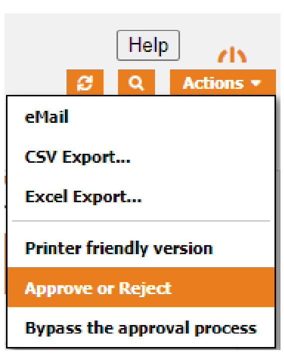
Show items for which my approval is being required

Total: 2 objects.

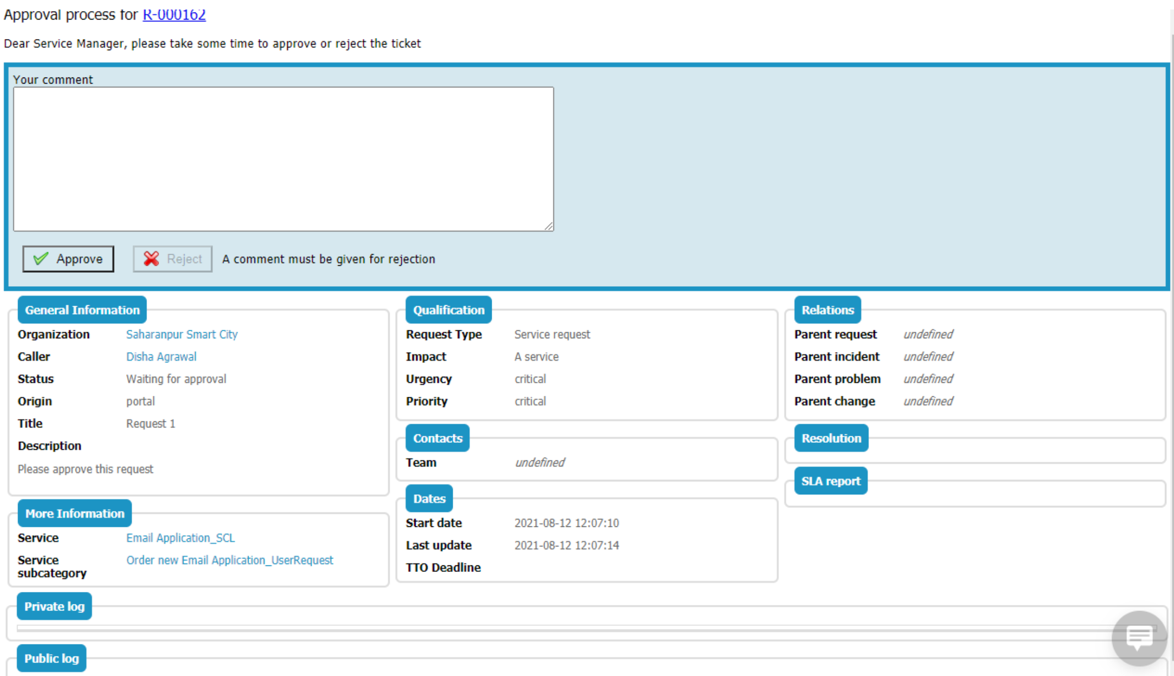
User Request	Title	Organization	Caller	Start date	Status	Agent
R-000031	New location rented	Demo	Power Portal	2019-09-16 15:06:12	Waiting for approval	*undefined*
R-000030	New comer	Demo	Portal Generic	2019-09-16 15:04:01	Waiting for approval	*undefined*

12.4.2. Approve or reject

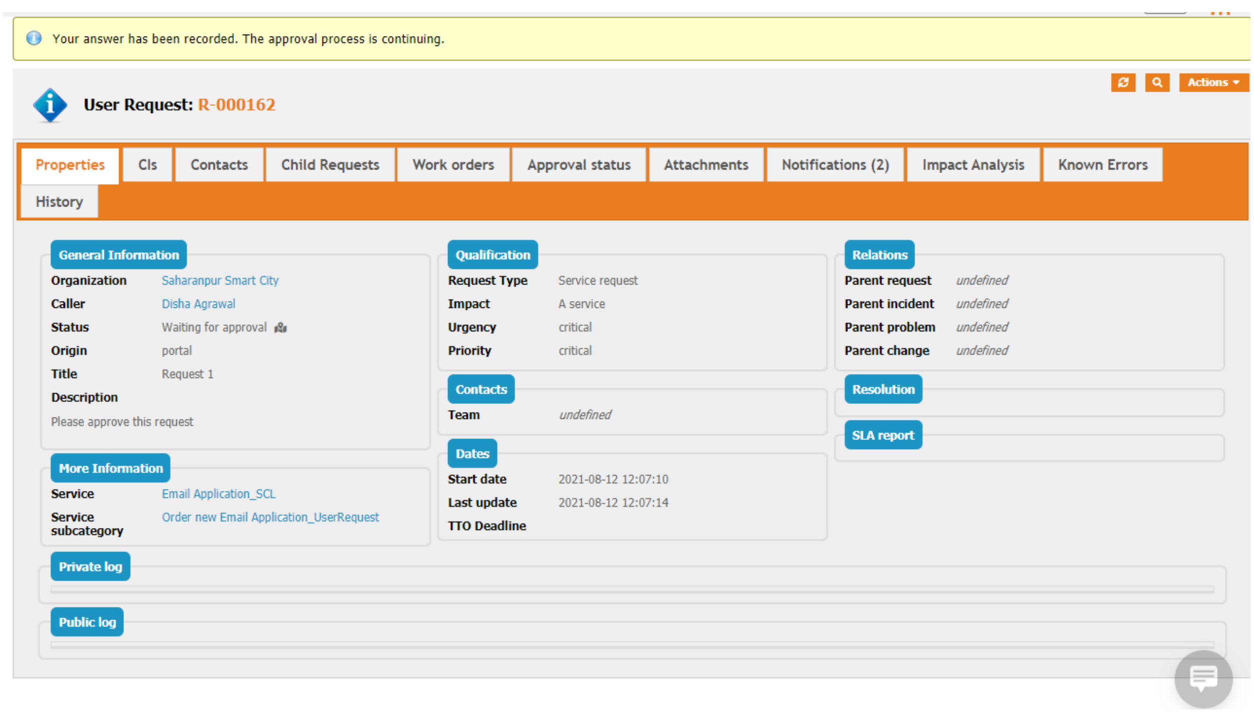
From the user request, open the **Other actions** menu and select **Approve or Reject**:



The approval form is displayed:

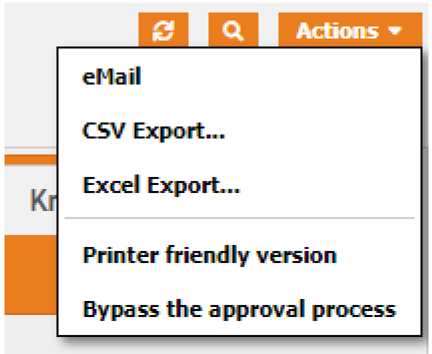


After the reply has been given, you are redirected to the user request and a banner reminds you the outcome of your reply.

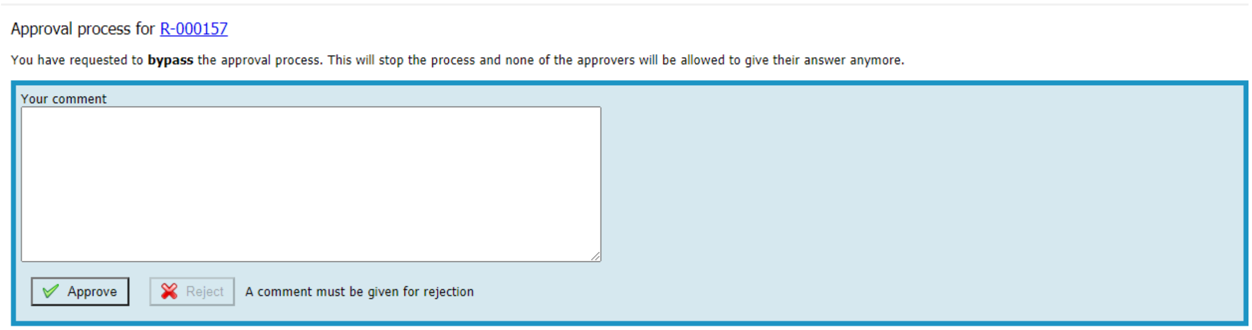


12.4.3. Bypass the approval process

If you are an administrator, and if the setup allows it, then you have a menu to bypass the process:



The approval form is then a little different than the standard reply form: it reminds you that bypassing the process is a little different.



If you are both an approver and allowed to bypass the process, then both menus are allowed. Using one or the other will just change the way the approval process result gets recorded and further displayed in the status tab.

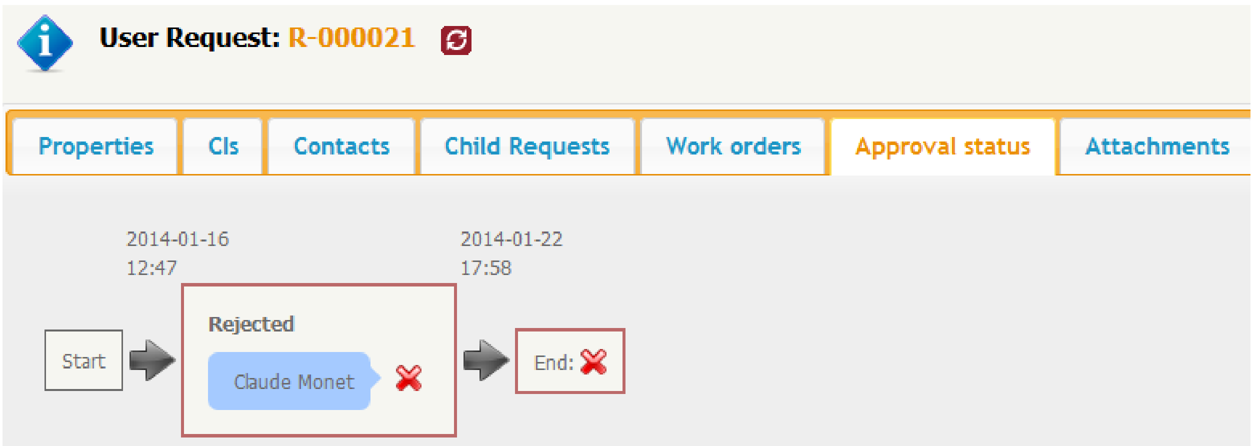
12.4.4. Status

As soon as a user request has been through an approval process, the tab **Approval status** shows detailed information about the ongoing or terminated approval.

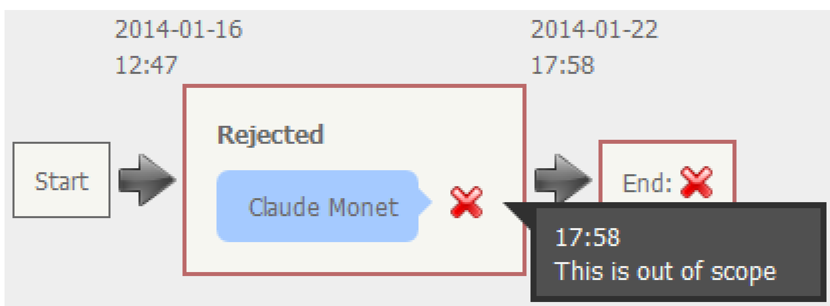
In the above example, the deadline is displayed in bold: 21st of January at 12:47.

Click on the button “send a reminder” to send a new message to the approver (confirmation required). This feature can be disabled by setting the parameter *enable_reminder* to false.

After the reply has been given, the status appears in clear:



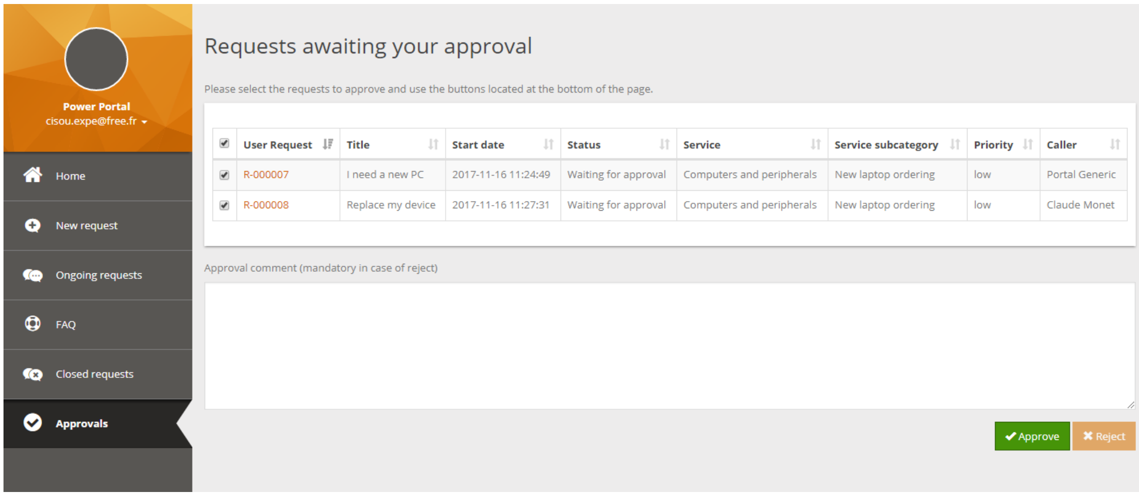
Move your mouse over the image next to the approver's name, and you will get the date of the answer and her comment if any has been given:



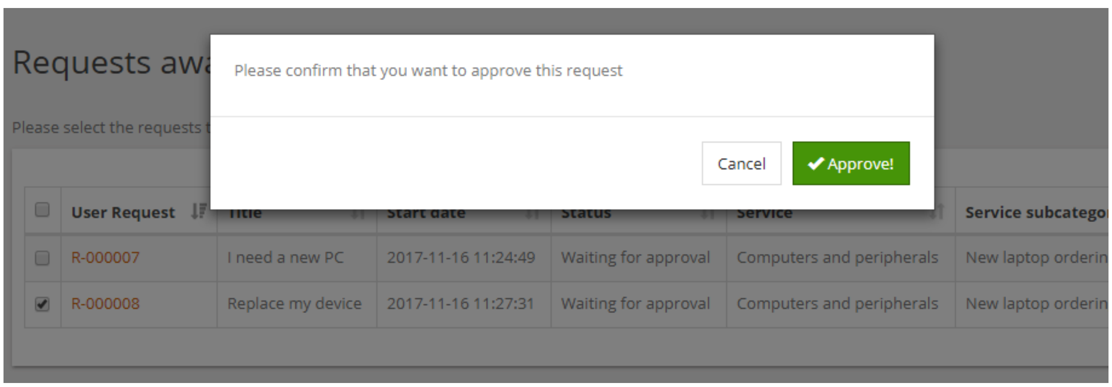
The status will be entirely reset anytime the user request enters the state “waiting for approval”.

12.4.5. Approval in portals

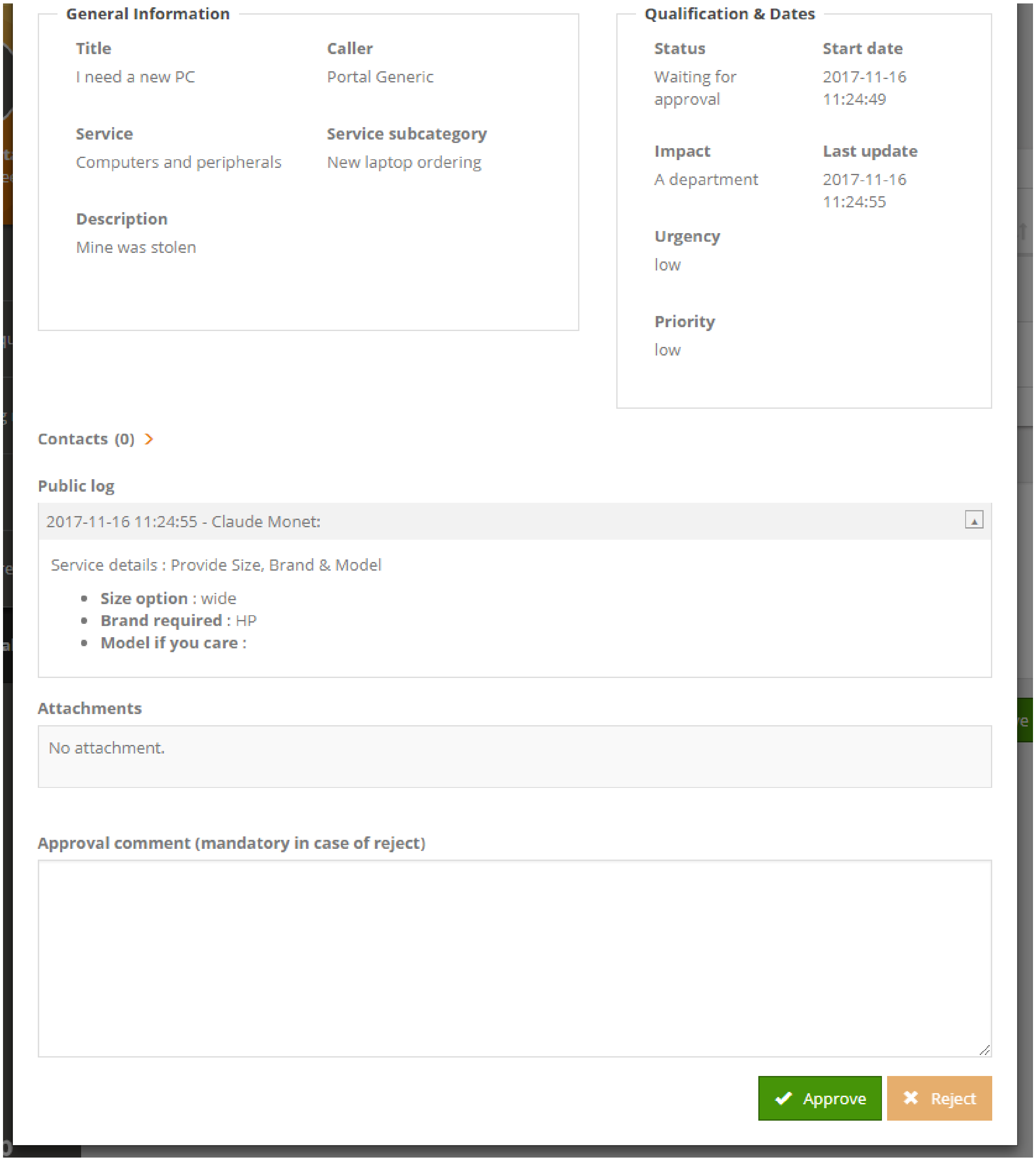
A new menu appears in the Enhanced Portal, which allow an approver to retrieve all User Requests waiting for her approval, and one by one or in bulk mode, accept or reject them.



A comment is required in case of rejection, so the button is disabled as long as the comment is empty. A confirmation is required in both cases.



When clicking on the link to a User Request, the ticket's details are displayed with an extra comment field and two buttons at the bottom of the details to accept or reject it:


General Information
Title
I need a new PC
Caller
Portal Generic
Service
Computers and peripherals
Service subcategory
New laptop ordering
Description
Mine was stolen
Qualification & Dates
Status
Waiting for approval
Start date
2017-11-16 11:24:49
Impact
A department
Last update
2017-11-16 11:24:55
Urgency
low
Priority
low
Contacts (0)
Public log
2017-11-16 11:24:55 - Claude Monet:
Service details : Provide Size, Brand & Model
Size option : wide
Brand required : HP
Model if you care :
Attachments
No attachment.
Approval comment (mandatory in case of reject)
Approve
Reject

13. Auto dispatch tickets to a Team

Automatically dispatch tickets to teams when entering a state.

13.1. Features

Allow to **dispatch automatically** Ticket based on predefined Dispatch rules, **to a team** and trigger a transition.

Each time a Ticket enter a state, tbITSM searched for Dispatch rules which apply to this class and state. If it finds one, then it uses each Team rules in order to retrieve a team. The Ticket is assigned to the first team found and Ticket is moved to a different state. If no team is found, the Ticket is left unchanged.

Example: when a Ticket is created, it is automatically dispatched to the team with role 'Support level1' defined on the customer Delivery Model, and moved to status 'Dispatch'.

This extension allows to define:

- different rules for different sub-class of Ticket,
- ordered queries to retrieve the team to use,
- force an automatic transition, if a team was set *(thus triggering notification)*
- Team can be assigned based on:
- customer of the ticket
- service or sub-service of the ticket
- location of the caller
- current time and applicable coverage windows
- or any other logic you may need…
- And any combination of those.
- If more than one team is returned, first will be used without guarantee that it will always be the same one.

13.2. Configuration

There is no parameter defined in the tbITSM configuration file.

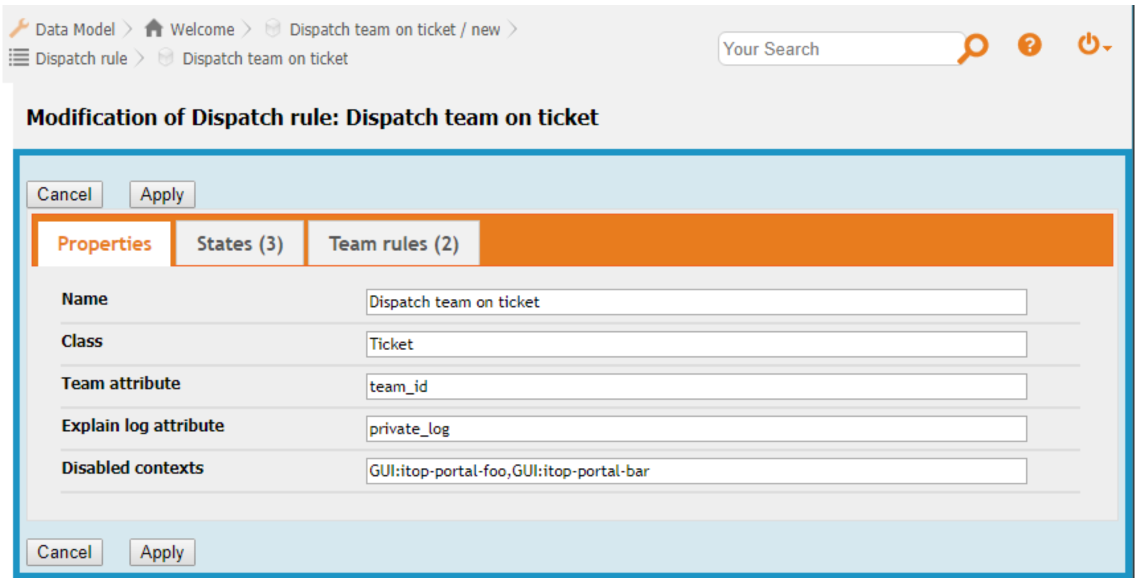
13.3. Usage

This extension does nothing out of the box, until you have created Dispatch rules to express when a Ticket should automatically assign to a team and which team to use.

13.3.1. Dispatch Rule

A dispatch rule is defined for one Class of Ticket. It must contain at least one Team rule and one State rule.

First create a new *Dispatch rule* by opening the corresponding menu under *Service Management*.



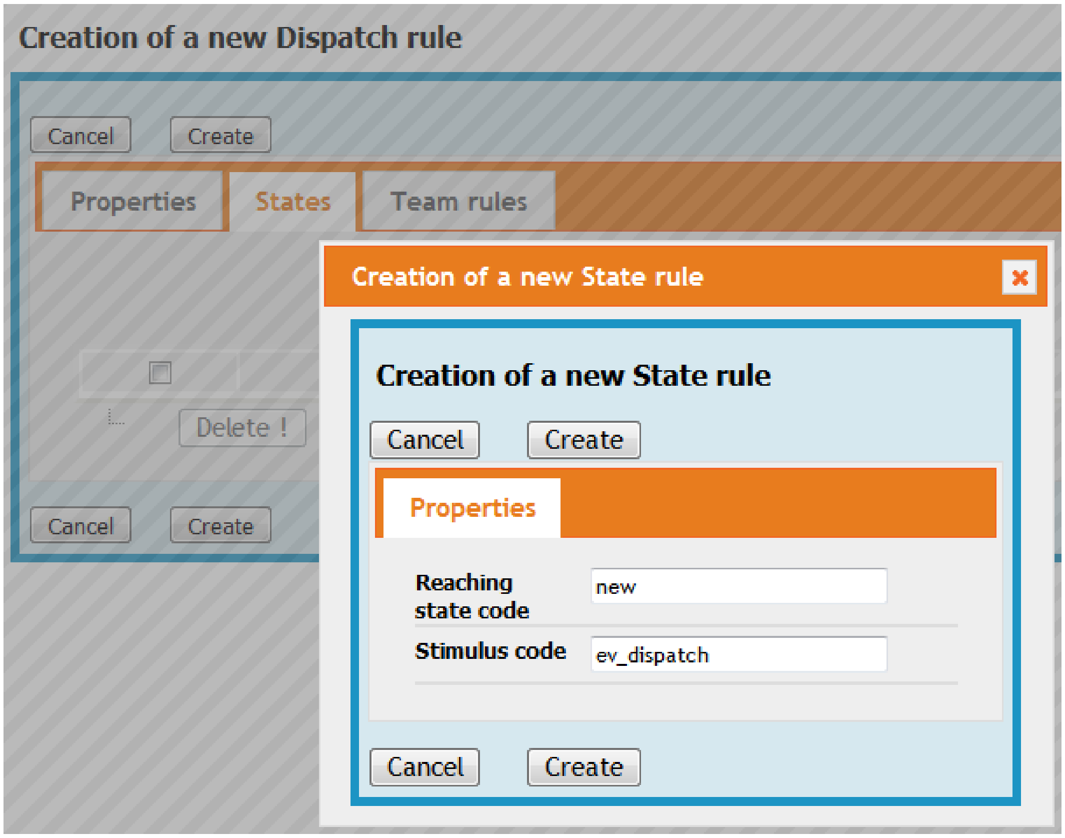
Name	Mandatory	Description
Name	Mandatory	A name describing the dispatch rule
Class	Mandatory	Ticket class the rule will apply to
Team attribute	Mandatory	Attribute code of the *Ticket Class* that will be set by the matching *Team rule*
Explain log attribute	Optional	Attribute code of the *Ticket Class* that will be set with a text explaining how a *Team rule* matched
Disabled contexts	Optional	A CSV list of context tags in which the *Dispatch rule* will be inactive. Typically, a portal (“GUI: Portal”), cron tab (“CRON”), a rest/json call (“REST/JSON”), …

Do not enter any space between contexts values, which must be coma separated.

13.3.2. State Rule

A State rule defines the state in which you want to auto-assign a team, and which transition must be applied if a team was found.

Create at least one *State rule* in the *States* tab.

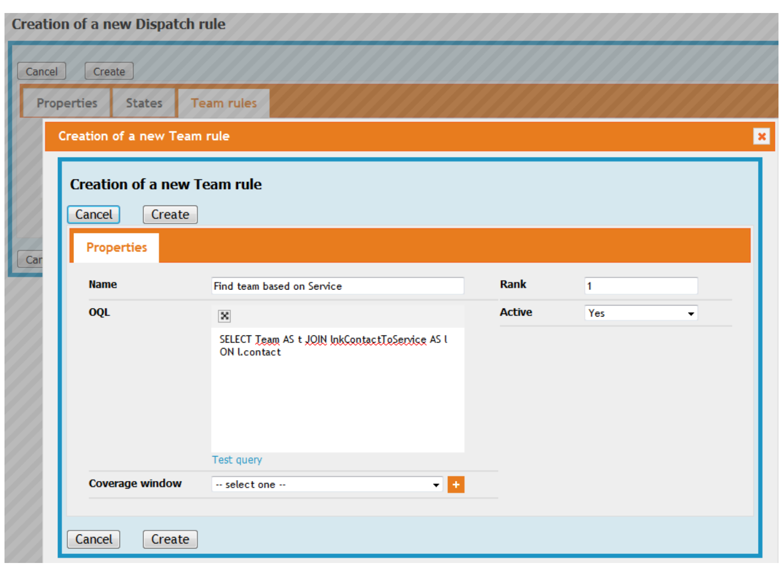


Name	Mandatory	Description
State code	Mandatory	Code of the Ticket state. Entering this state will trigger the *Dispatch rule*
Stimulus code	Mandatory	Code of the *Stimulus* that will be applied if a team is found

13.3.3. Team Rule

A Team rule defines how to retrieve the team to assign.

Create at least one *Team rule* in the corresponding tab.



Name	Mandatory	Description
Name	Mandatory	Free name of the rule, for humans.
OQL	Mandatory	Query which must return *Team* objects
Coverage window	Optional	The *Team rule* will be used only if we are within the *Coverage window*
Rank	Mandatory	Order for using the Team rules, lowest number first.
Active	Mandatory	Allow to prepare *Team rules*, without activating them. Rules with 'No' will not be used.

14. Auto-close Tickets

This page describes how the extension tbitsm auto close allows tbITSM to automatically close resolved User Request and Incident after a configured delay.

14.1. Features

This extension adds a new background process handled by cron.php to automatically close User Request and Incident according to the auto closure delay defined in the tbITSM configuration file.

14.2. Usage

When a user request or an incident is closed automatically, the user satisfaction is set to the default value defined in the data model: “Very satisfied”

15. Customer Survey

Integrates with the tbITSM CMDB so that you can send a satisfaction survey to the customers documented in tbITSM.

15.1. Features

A quiz is made of a list of questions of the following type:

- a free text comments
- a choice within a list
    - either a list defined once for all in your quiz (usually a rating, from *bad* to *good*)
    - or a list defined only for that question (e.g., types of products)

The quiz can be split on several pages. It has an introduction and a footer message.

A quiz will be used in one or more survey campaigns.

The survey defines to whom the quiz will be sent (scope given as an OQL query). Customers will receive an email with an URL to reply to the quiz.

The results can be viewed and exported to an Excel file.

By default, the module keeps track of the link between the reply and the contact. As this is prohibited in some countries, depending on local regulations, the module has an anonymous mode.

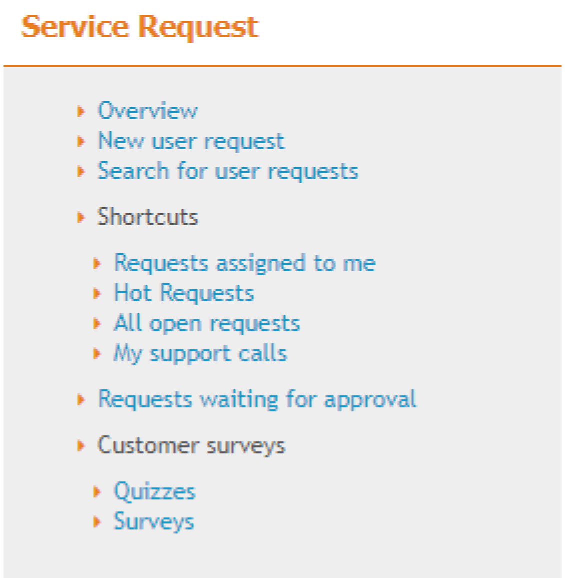
15.2. Configuration

The following settings can be adjusted in the tbITSM configuration file, in the section customer-survey:

Parameter	Type	Description	Default Value
anonymous_survey	Boolean	Set to true to guarantee that the module will not keep track of who did a given reply	false
quiz_scale	String	CSV list of labels (5 must be given?) that define the default scale labels	Very bad, Bad, Average, Good, Very good

15.3. Usage

This module creates two new entries in the menu group *Service Request*:



Quizzes and surveys are two new types of data:

- A quiz is a series of questions in a given language.
- A survey defines the targeted people. It is also the way to follow-up on the progress and get a report on the replies.

Create a quiz
The first step is to create a Quiz, i.e., a series of questions.

Properties

In the tab “Properties”, you will have to specify the following properties:

Property	Description
Name	A name for internal identification of the quiz
Description	A description for internal usage
Quiz Language	The language in which the quiz must be displayed. This choice has an effect on the labels of the buttons and other things like “this question is mandatory”
Default Quiz Choices	Standard choices for questions of type “predefined choices”. Leave it blank to use the setting from the configuration file: *quiz_scale*
Title	The first page title: *<title> page 1 of 2*
Introduction	Just below the title, an introduction message (optional)
Conclusion Message	Displayed after the quiz has been completed and submitted. This is the place to say thank you!

Questions
In the tab “Questions”, enter your questions and page breaks.

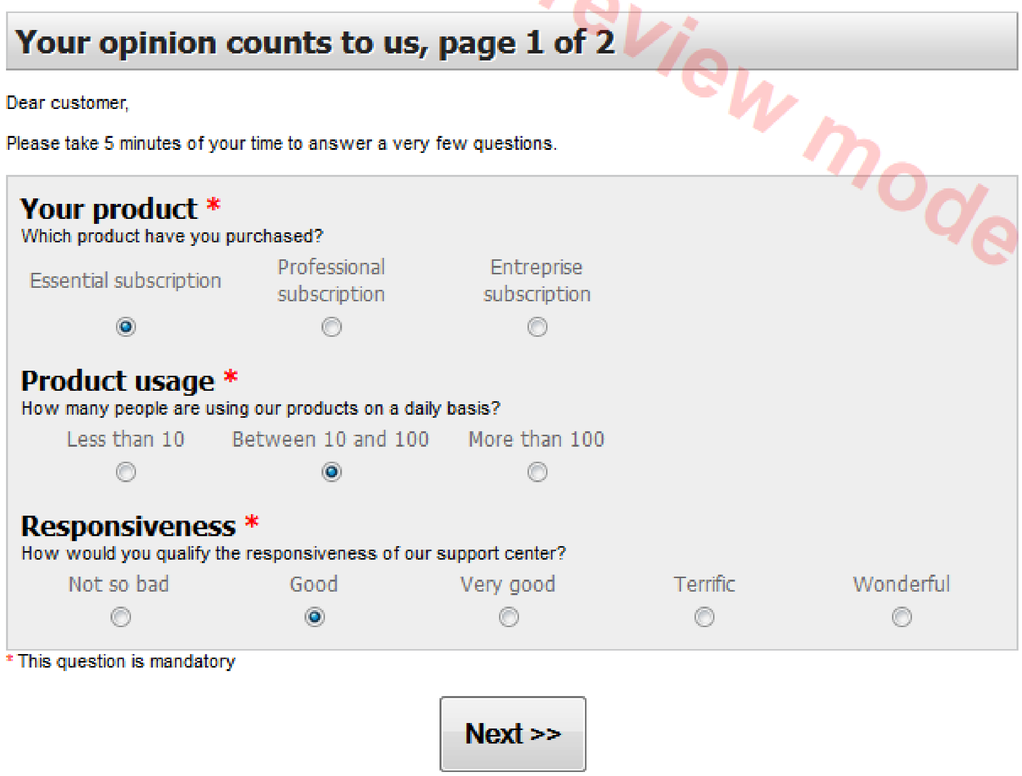
To modify a question, save the Quiz first, then from the "Questions" tab, click on the link to display the details of the desired question. You can then click on the "Modify" button to alter the question, as you would do for any element in tbITSM.

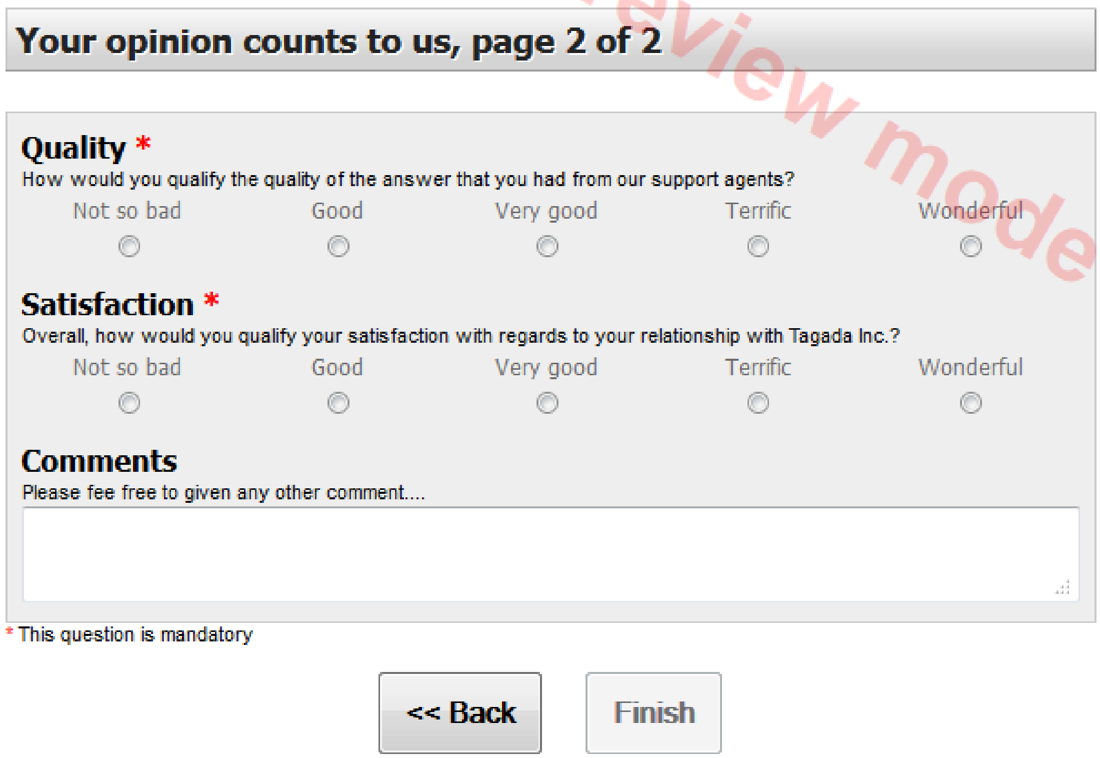
We recommend to use non-consecutive ordering numbers (e.g., 100, 200, 300) in order to easily insert a new question without having to renumber (i.e., modify) all the questions.

Type of question	Description
Free text	This is the place for a user comment.
Predefined choices	The user is asked to select an item out of a list defined for the whole quiz, in the property *Default Quiz Choices*. This is the way to have a standard defined at the quiz level or at the application level.
Specific choices	The user is asked to select an item out of a list defined solely for this question.
Page break	Though this is not a real question, a page break is inserted into the stream of questions. The title and description of the page break replace respectively the quiz title and introduction, for the page that comes AFTER the break.

Preview

Once the quiz has been created, click on the tab "Preview" to display the quiz as it will be seen by your customers:





Start a survey

Create the survey

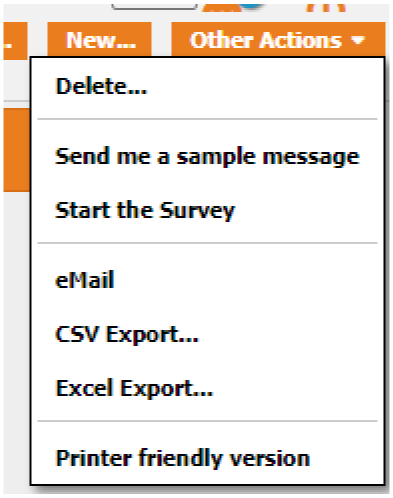
In the tab “Properties”, set the following properties:

Property	Description
Quiz	Select the quiz amongst the already created quizzes. Note: the language is displayed right above: it will be updated after the quiz is saved.
On behalf of	The sender of the survey. The email will be sent with *from* set to this person. The list proposes any person documented in tbITSM.
Email on completion	If set to *yes*, and if the sender has an email address, then she will receive an email each time a recipient completes the survey
Email subject	The email title.
Email body	The email body, in HTML.
Recipients of the survey	An OQL phrase, from the *Query phrasebook* (in the menu *Admin tools*)

It is possible to complete the list of target contacts defined by the OQL: go into the Additional contacts tab, and add them one by one.

Tune the survey
Creating the survey does not send the message. You still have the possibility to adjust the settings, review the quiz.

In the *Other Actions* menu, click on *Send me a sample message*.



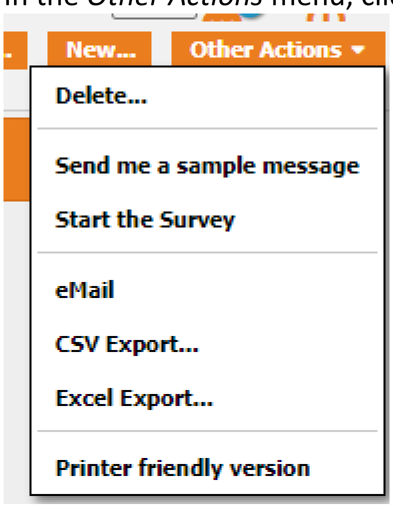
It reminds you the settings of the survey.

Confirm by clicking the *Send me a sample message* button.

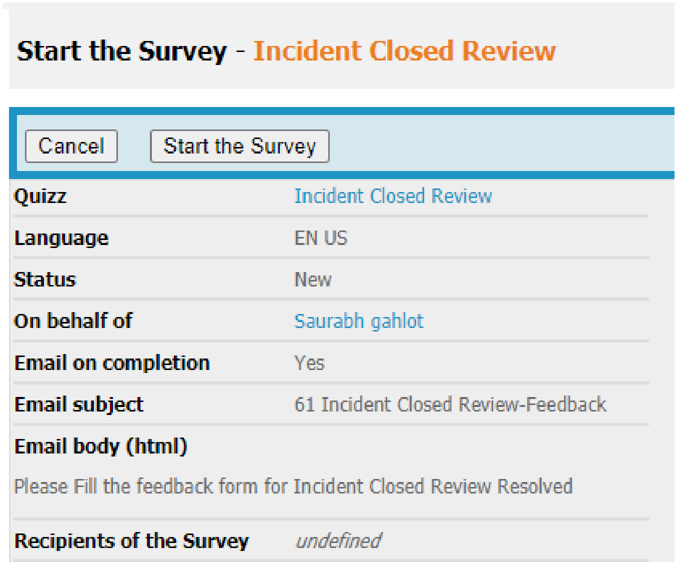
You will receive a sample email, and see what your customers should receive. In particular, this is the opportunity to check language consistency.

Send the survey

In the *Other Actions* menu, click on *Start the survey*.



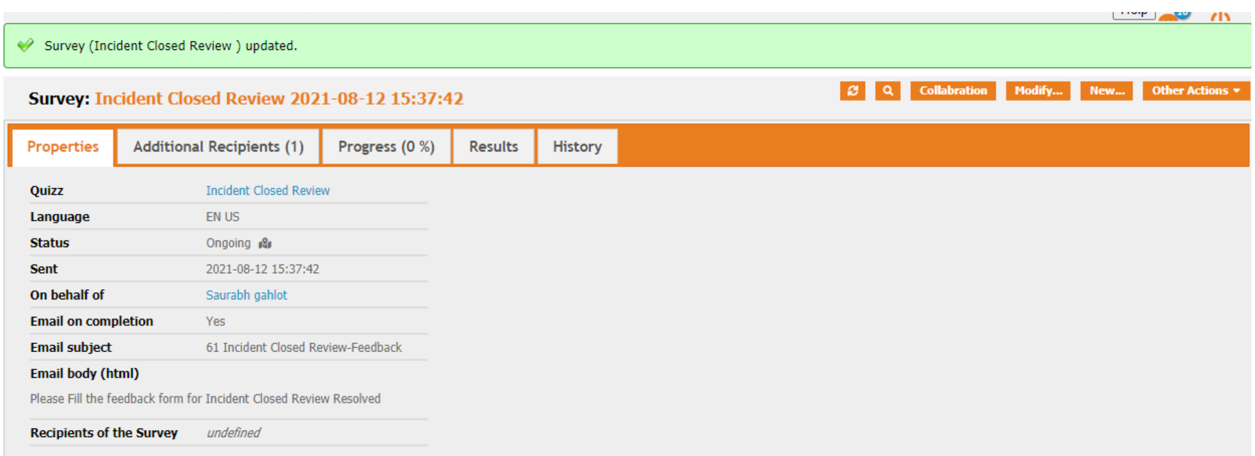
It reminds you the settings of the survey.



Confirm by clicking the *Start the survey* button.

The survey is now started:

- its name is made of the quiz name and the survey start date
- two new tabs have been added: *Progress* and *Results*



All target contacts will receive an email with a link to the quiz form.

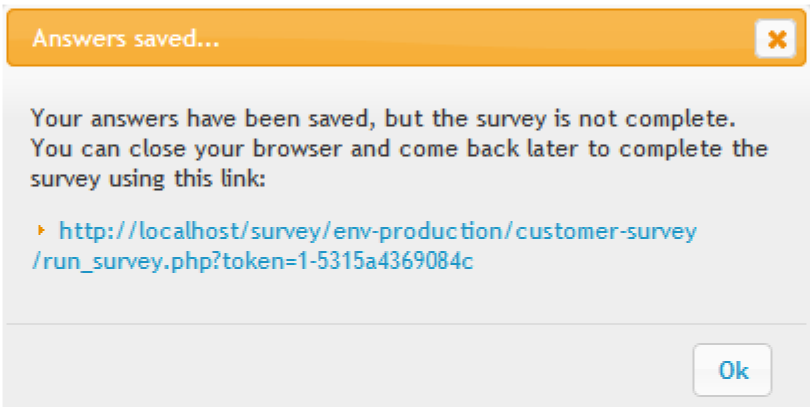
When the survey is running, it is still possible to add recipients: edit the survey and add contacts into the *Additional Recipients* tab. The newcomers will be notified as you save the changes on the survey.

A survey is not closed automatically: when 100% of the users have given their answer, it is still possible to add new contacts and wait for their replies.

Replying to the survey

Replying does not require to login to tbITSM.

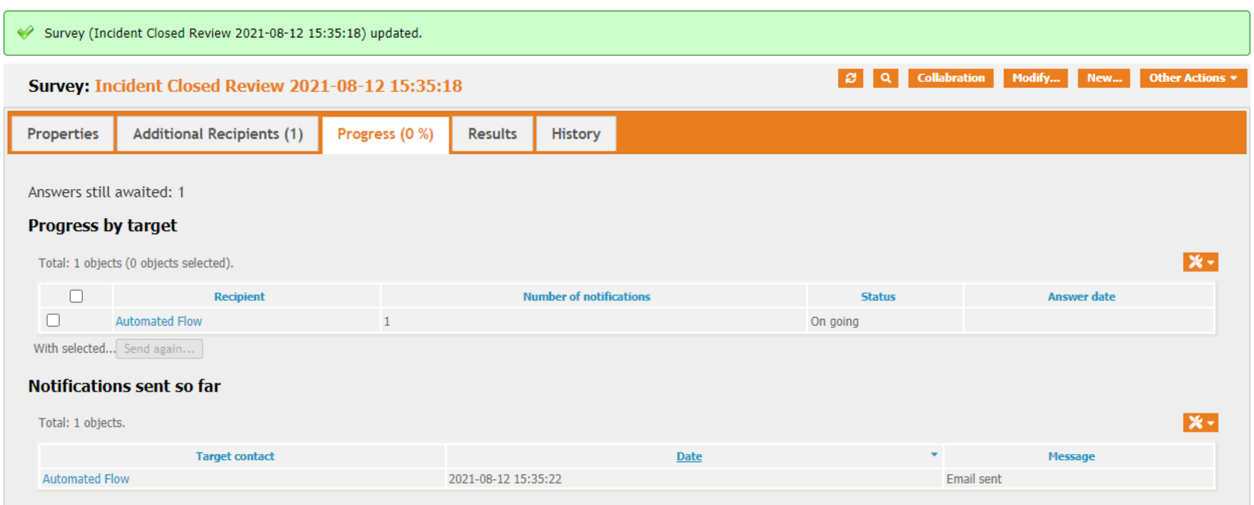
The customer has the capability to *suspend* the quiz at any time by clicking on the *suspend* button.



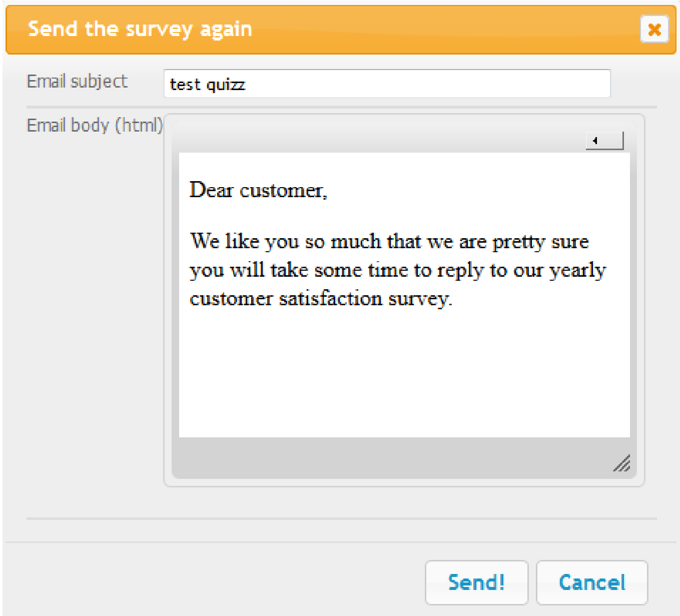
Resuming can be done either by using the URL provided in the above popup, or by using again the URL provided in the received email.

Monitoring the progress

The tab *Progress* gives an overview of who has received the quiz:



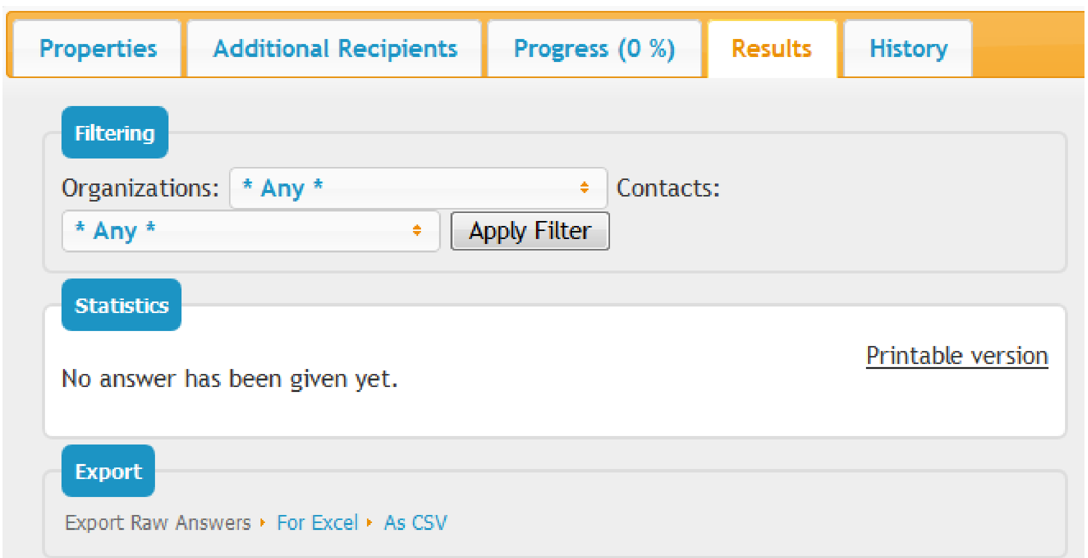
Select a contact and click on *Send again* to resend the email:



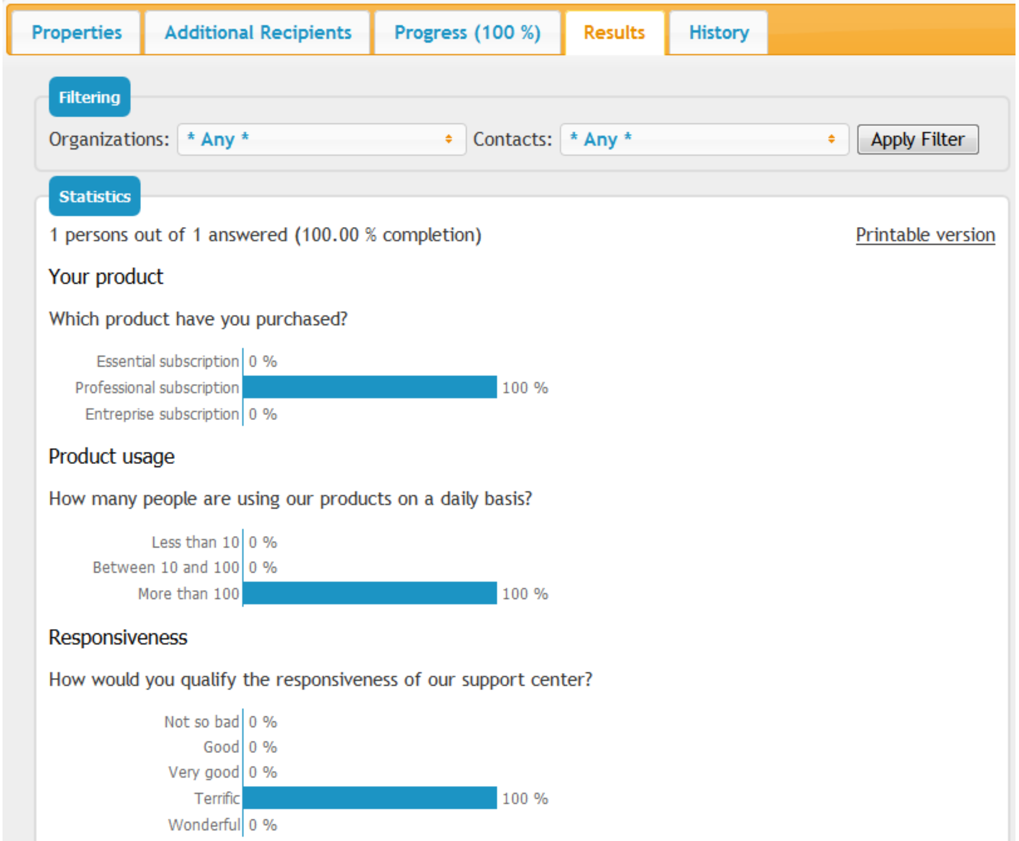
Email subject and body are prepopulated with the original values. Adjust them and confirm by clicking on the button *Send.*

Exploiting the results

Just after the survey has been started, the result tab will look like this:



With data, the results look like this:


Properties
Additional Recipients
Progress (100 %)
Results
History
Filtering
Organizations: * Any *
Contacts: * Any *
Apply Filter
Statistics
1 persons out of 1 answered (100.00 % completion)
Printable version
Your product
Which product have you purchased?
Essential subscription 0 %
Professional subscription 100 %
Entreprise subscription 0 %
Product usage
How many people are using our products on a daily basis?
Less than 10 0 %
Between 10 and 100 0 %
More than 100 100 %
Responsiveness
How would you qualify the responsiveness of our support center?
Not so bad 0 %
Good 0 %
Very good 0 %
Terrific 100 %
Wonderful 0 %

16. Data Archiver Simple

tbITSM is getting slow due to a high volume of tickets? You would like to get rid of them, but… what if a customer claims for an old ticket? Install this extension and you will gain performance, without losing data.

- Allow administrators to bulk flag outdated Tickets as Archived
- Hide Archived Tickets to all users most of the time, as if they were deleted.
- But still enable them to retrieve Archived Tickets if needed.

16.1. Features

- Improve performance when searching for a Ticket, if your outdated Tickets represent much more than the active ones, but cannot be deleted.
- Allow administrators to bulk flag outdated Tickets as Archived
- Hide Archived Tickets to all users most of the time, as if they were deleted.
- But still enable them to retrieve Archived Tickets if needed.

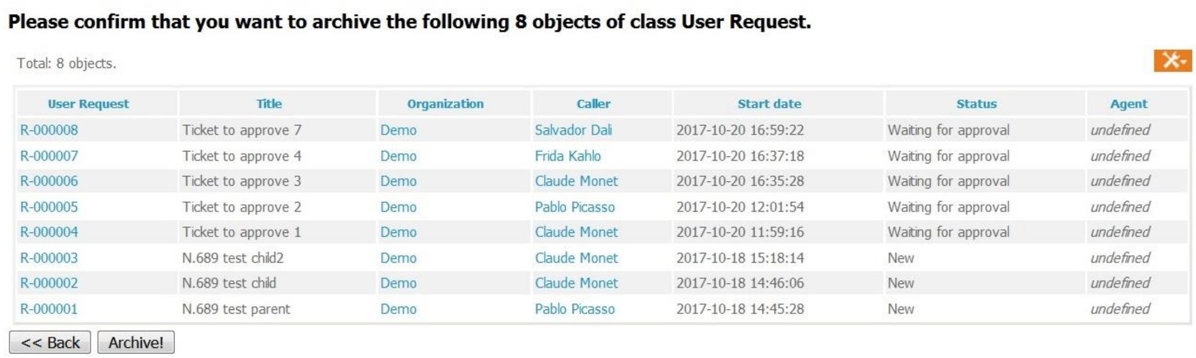
16.2. Administrator experience

When you have deployed this extension, then Tickets objects can be massively archived (and unarchived) by administrators only.

Search for the Tickets that you want to Archive and from that list, open the **Actions** menu. Below two possibilities, in standard mode (only mass archive action available) and in archive mode (both mass archive and unarchive actions available)

.

Mass archive or unarchive actions sends to a confirmation screen:

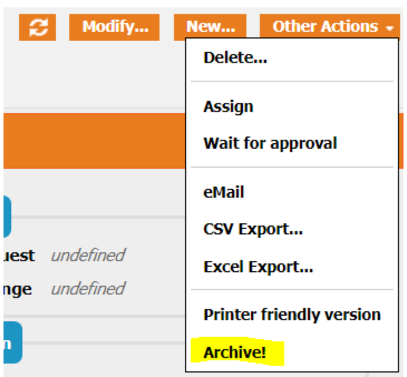


Please confirm that you want to archive the following 8 objects of class User Request.

Total: 8 objects.

User Request	Title	Organization	Caller	Start date	Status	Agent
R-000008	Ticket to approve 7	Demo	Salvador Dali	2017-10-20 16:59:22	Waiting for approval	*undefined*
R-000007	Ticket to approve 4	Demo	Frida Kahlo	2017-10-20 16:37:18	Waiting for approval	*undefined*
R-000006	Ticket to approve 3	Demo	Claude Monet	2017-10-20 16:35:28	Waiting for approval	*undefined*
R-000005	Ticket to approve 2	Demo	Pablo Picasso	2017-10-20 12:01:54	Waiting for approval	*undefined*
R-000004	Ticket to approve 1	Demo	Claude Monet	2017-10-20 11:59:16	Waiting for approval	*undefined*
R-000003	N.689 test child2	Demo	Claude Monet	2017-10-18 15:18:14	New	*undefined*
R-000002	N.689 test child	Demo	Claude Monet	2017-10-18 14:46:06	New	*undefined*
R-000001	N.689 test parent	Demo	Pablo Picasso	2017-10-18 14:45:28	New	*undefined*

You can also do it on a single ticket details screen, using the **Actions** menu:



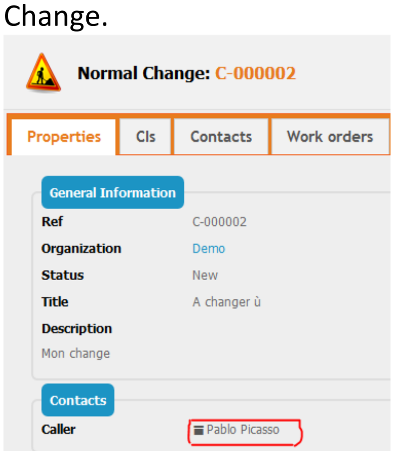
Both single object archive and unarchive actions are done without confirmation.

16.3. End User experience

In standard mode

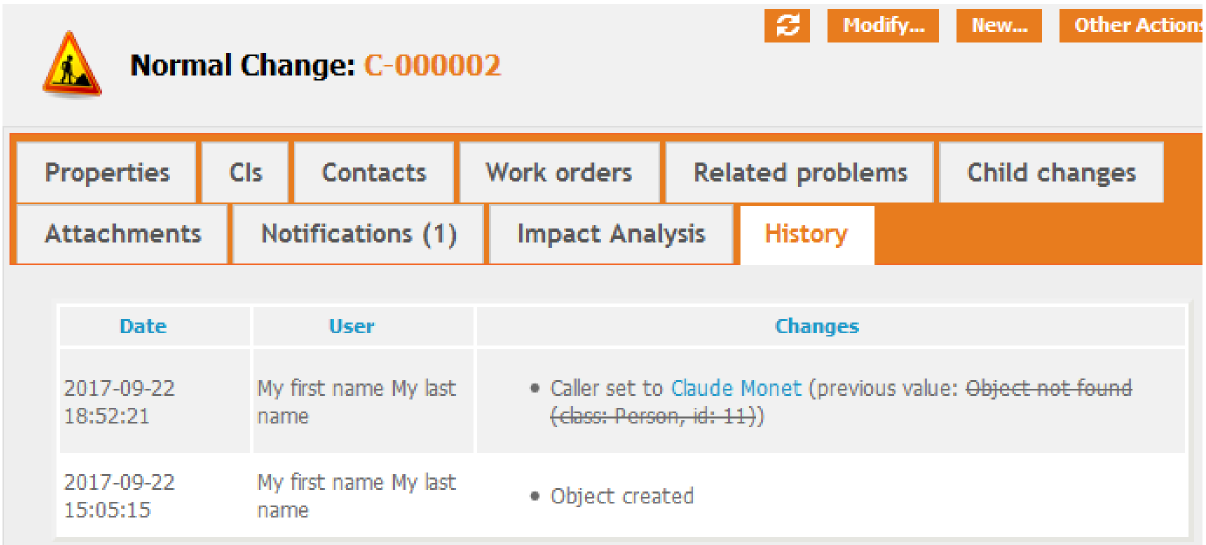
All archived objects are hidden, for all users including admins, like if they were deleted.

A reference to their friendly name can be found in other objects pointing to them. Example on an archived contact which is the caller of a non-archived Change.

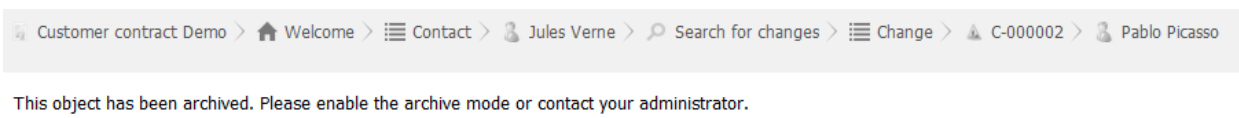


As you can see the link is inactive, you can't open the caller details.

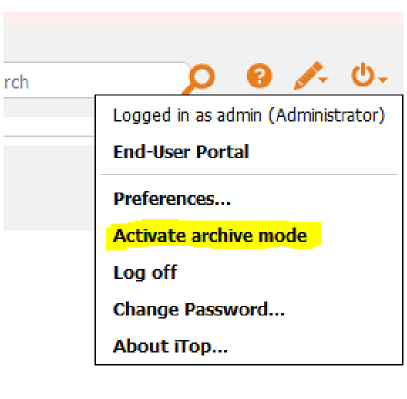
In the history of other objects to which they are or were linked, you just get the id of that archived object:



Then you will get a message like this one:



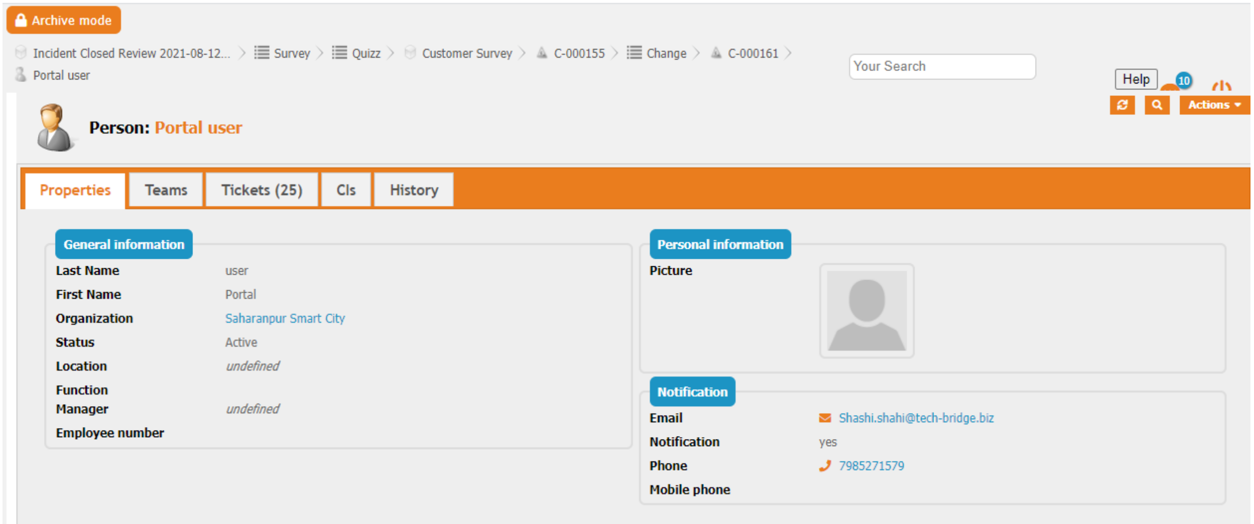
When an object is archived, all its n: n linkages to other objects are archived as well, meaning that they aren't visible anymore in Archive mode



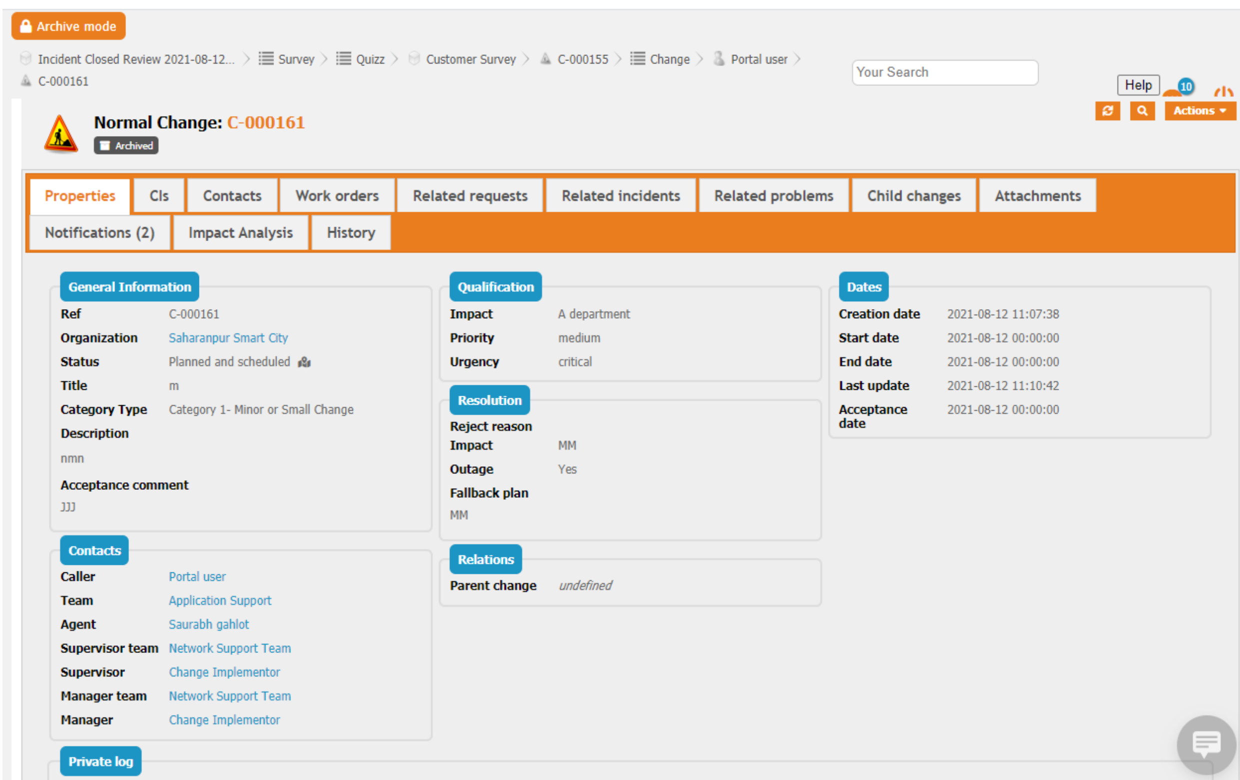
Any user can toggle the archive mode:

- In archived mode, you have an orange tag to remind you that you have activated it.

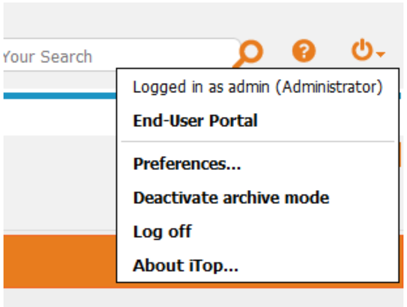
- All objects are **read-only**
- Archived objects are **visible** and tagged as archived



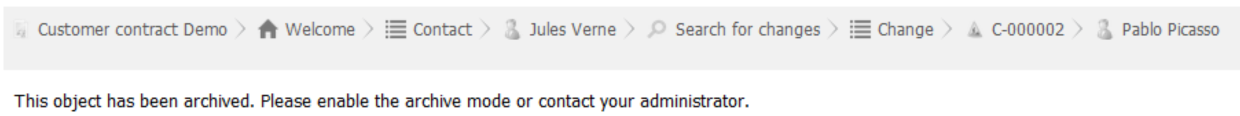
An attribute referencing an archived object is clickable:



If you deactivate archive mode while you are on an archived object:

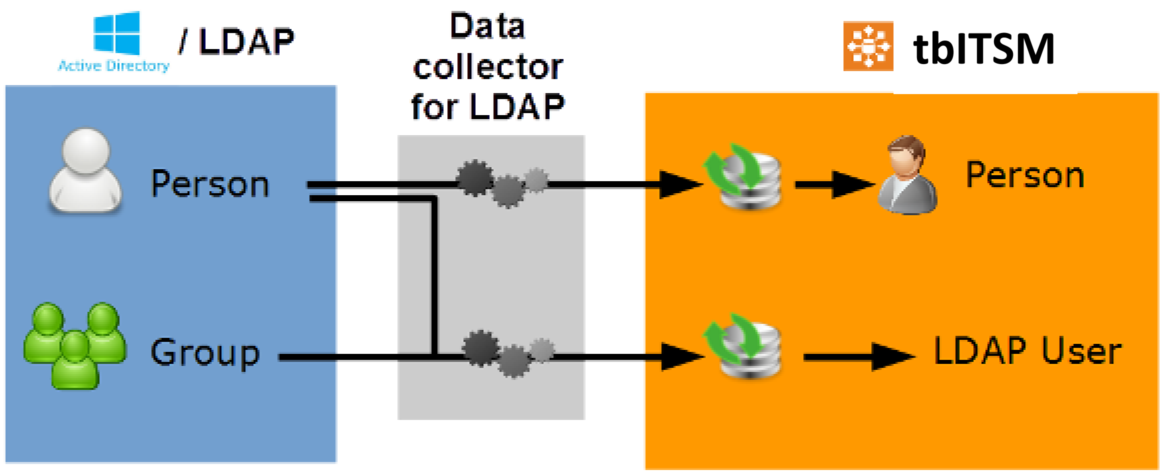


you get again the feedback message



17. Data Collector for LDAP

This stand-alone application collects information from a **single** LDAP Directory in order to automatically synchronize the persons and the users in tbITSM.



17.1. Features

Main functions:

- Automatic creation and update of Persons and Users in tbITSM based on LDAP data.
- Automatic assignment of Profiles to Users based on LDAP groups (this is optional).

Technical aspects:

- The collector can reside on any system with web access to tbITSM and LDAP access to the LDAP Directory
- The collector is compatible with Windows Active Directory
- The definition of the mapping between LDAP fields and tbITSM fields is fully configurable.
- The creation of the Synchronization Data Sources in tbITSM is fully automated.

This collector makes use of tbITSM' s built-in Data Synchronization mechanism. For more information about how the data synchronization works, refer to Data Synchronization Overview and relies on Data collector Base mechanism

17.2. Configuration

The file params.distrib.xml contains the default values for the parameters. Both files (params.distrib.xml and params.local.xml) use exactly the same format. But params.distrib.xml is considered as the reference and should remain unmodified. Should you need to change the value of a parameter, copy and modify its definition in params.local.xml. The values in params.local.xml have precedence over the ones in params.distrib.xml

The default values for the configuration of the data collection are defined in the file collectors/params.distrib.xml. This configuration defines which LDAP queries are executed on the LDAP server to retrieve the data, how to map the LDAP fields with the tbITSM fields and some default values for the tbITSM fields.

Parameter	Meaning	Default value
ldappersonfilter	The LDAP query used to retrieve the persons in LDAP/AD	(objectClass=person)
person_fields	The list of tbITSM fields of the Person object to populate from the LDAP data, and for each tbITSM field its mapping to the corresponding LDAP field	<person_fields> <primary_key>samaccountname</primary_key> <name>sn</name> <first_name>givenname</first_name> <email>mail</email> <phone>telephonenumber</phone> <mobile_phone>mobile</mobile_phone> <function>title</function> <employee_number>employeenumber</employee_number></person_fields>
person_defaults	The default values for some tbITSM fields for a Person. Used either when the LDAP query returns an empty value or if no mapping is defined for the field	<person_defaults> <org_id>Demo</org_id> <status>active</status> </person_defaults>
collect_person_only	Whether or not to synchronize users from LDAP (yes/no)	No
ldapuserfilter	The LDAP query to use to retrive user information in LDAP. *Note: the ampersand character & is a special character in XML and must be written as &*	(&(objectClass=person) (mail=*))
synchronize_profiles	Flag to activate or not synchronization of the user profiles based on defined LDAP groups. If set to yes, the synchronization of the profiles is using the tbITSM_group_pattern to identify corresponding group. If set to no make sure that you specify a default profile, since users cannot be created	No

	without at least one profile.	
tbITSM_group_pattern	Regular expression pattern to retrieve list of LDAP group to map with tbITSM profiles. The first capturing group (i.e., parentheses) must return the name of an existing tbITSM profile. The default regular expression looks for groups named tbITSM-<tbITSM Profile Name>	/^CN=tbITSM- (. *), OU=. */
user_fields	The list of tbITSM fields for the LDAPUser object, to populate from the LDAP data, and for each tbITSM field its mapping to the corresponding LDAP field	<user_fields> <primary_key>samaccountname</primary_key> <login>samaccountname</login> <contactid>mail</contactid> </user_fields>
user_defaults	The default values for some tbITSM fields for a UserLDAP. Used either when the LDAP query returns an empty value or if no mapping is defined for the field.	<user_defaults> <profile>Portal user</profile> <language>EN US</language> </user_defaults>

Those parameters can be redefined in the file conf/params.local.xml in order to take into account your specific needs. (For instance, the mapping between tbITSM and LDAP attributes)

The expected value for person_defaults/org_id is an organization name, not an id
The expected value for user_fields/login can be UID, samaccountname, mail, but the field must contain unique values

The expected value for user_fields/contactid is a field containing an **email address**
user_defaults/profile is a shortcut to fill the LDAP User field named profile list with one unique
profile.
If you want to assign several profiles to the LDAP Users, use the tag profile list with this format:

<user_defaults>
<profile_list>profileid->name:name_of_profile1|profileid->name:name_of_profile2</profile_list>

Other optional parameters

The following parameters can be redefined to alter the default behavior of the collector:

Parameter	Meaning	Default value
max_chunk_size	Maximum number of elements to process in one iteration (for upload and synchro in tbITSM). If there are more elements than this number, the process will automatically iterate.	1000
tbITSM_synchro_timeout	Timeout for waiting for the execution of one data synchro task (in seconds)- requires php_cURL	600
stop_on_synchro_error	Whether or not to stop when an error occurs during a synchronization (yes or no).	no
console_log_level	Level of ouput to the console. From -1 (none) to 9 (debug).	6 (info)
console_log_dateformat	Logger timestamp format	[Y-m-d H: i: s]
cURL_options	When using cURL to connect to the tbITSM Webservices the cURL options can be specified in this section. The syntax is <CURLOPT_NAME_OF_THE_OPTION1>VALUE 1</CURLOPT_NAME_OF_THE_OPTION1> where VALUE_x is either: The numeric value of the option, or the string representation of the corresponding PHP "define" (case sensitive). It is possible to define several php_cURL options like in the example below	
data_path	The path where to store the temporary files generated by the collector. You can use the special placeholder %APPROOT% to specify a path relative to the root folder of the collector.	%APPROOT%/data

17.3. Truncated data

If the collector does not retrieve all the expected records, this may be due to the server enforcing a limit to the length of the search results. This limit can be overcome by setting the parameter page_size (in the configuration file params.local.xml) to activate the pagination. Set a value less than the maximum number of search results allowed by the LDAP server to ensure that the data will not be truncated.

The support of pagination requires PHP 7.3 (or newer) to run the collector.

17.4. Usage

To launch the data collection and synchronization with tbITSM, run the following command (from the root directory where the application is installed):
php exec.php

The following (optional) command line options are available:

Option	Meaning	default value
--config_file	Specify the full path to the configuration file. The file conf/params.local.xml is used by default if this parameter is omitted.	empty
--console_log_level=<level>	Level of ouput to the console. From -1 (none) to 9 (debug).	6 (info)
--collect_only	Run only the data collection, but do not synchronize the data with tbITSM	false
--synchro_only	Synchronizes the data previously collected (stored in the data directory) with tbITSM. Do not run the collection.	false
--configure_only	Check (and update if necessary) the synchronization data sources in tbITSM and exit. Do NOT run the collection or the synchronization	
--max_chunk_size=<size>	Maximum number of items to process in one pass, for preserving the memory of the system. If there are more items to process, the application will iterate.	1000
--help	Usage mode to display exec.php help.	

Dates in source data must use this format **YYYY-MM-DD (*hh: mm: ss*)**

17.5. Scheduling

Once you've run the data collector interactively, the next step is to schedule its execution so that the collection and import occurs automatically at regular intervals.

The data collector does not provide any specific scheduling mechanism, but the simple command line php exec.php can be scheduled with either cron (on Linux systems) or using the Task Scheduler on Windows.

For optimal results, don't forget to adjust the configuration parameter full_load_interval in the (json_placeholders' section) to make it consistent with the frequency of the scheduling.

The execution of the command line will:

- Connect to tbITSM to create the Synchronization Data Sources (or check their definition if they already exist, updating them if needed)
- Connect to the LDAP server to collect the information about the Persons and the Users
- Upload the collected data into tbITSM

- Synchronize the collected data with the existing tbITSM Person and Users.

When the collector is run, two Synchro Data Sources are created and used for synchronizing Person and LDAP User objects in tbITSM:

Total: 4 objects.

Synchro Data Source	Target class	Status	Type of account	User	Full load interval
Idap1-Synchro LDAP Person	Person	Production		*undefined*	0s
Idap1-Synchro User LDAP	LDAP user	Production		*undefined*	0s
Idap2-Synchro LDAP Person	Person	Production		*undefined*	0s
Idap2-Synchro User LDAP	LDAP user	Production		*undefined*	0s

18. Data Synchronization Dashboard

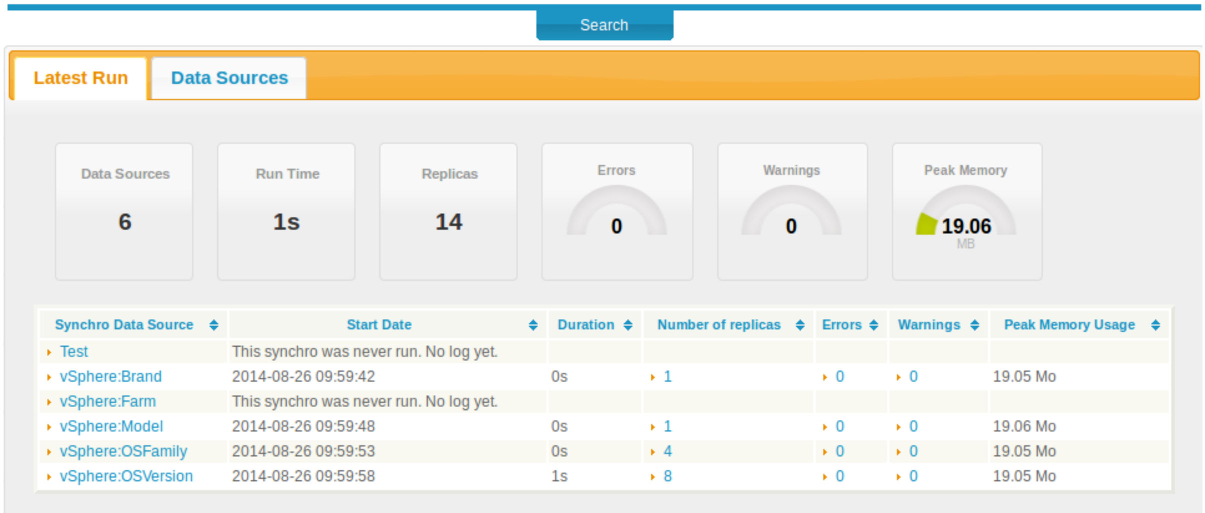
This extension provides an extra menu (under the Admin tools section) linking to a page that provides an overview of all Data Synchro Sources (last run, number of replicas, number of errors…)

18.1. Features

A quick overview to check the health of the running data synchro sources.

18.2. Usage

Click on the menu "Admin Tools / Synchronization Dashboard" to display the following page:



The first tab shows the list of all the Synchronization Data Sources with 6 overall indicators:

Data Sources	The total number of data sources
Run Time	The cumulated run time of the last time each data source was run
Replicas	The total number of replicas for all the data sources
Errors	The cumulated number of errors for all data sources, for their latest run
Warnings	The cumulated number of warnings for all data sources, for their latest run
Peak memory	The highest amount of PHP memory used by a data source during its latest run

The list under the indicators shows the same metrics, for each data source.
The second tab is simply the list of data sources.

Using the "search" form at the top of the page it is possible to filter the list of data sources. The filter also applies to the overall statistics.

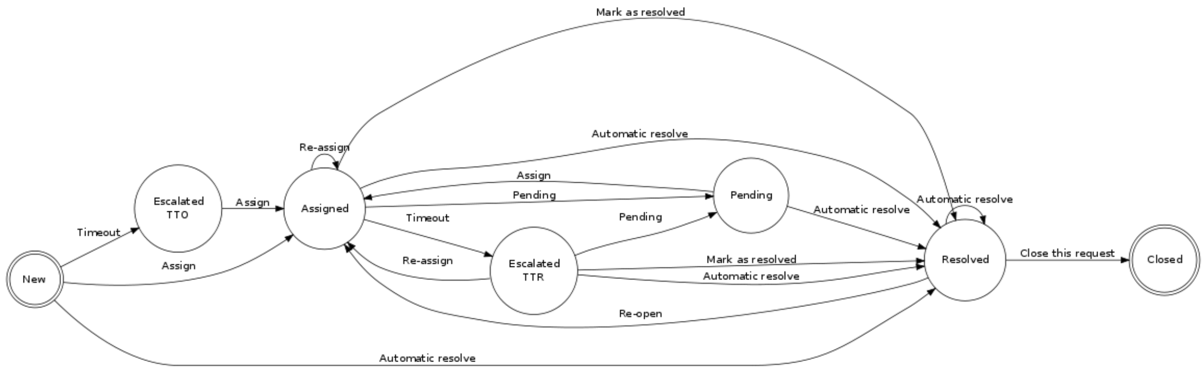
19. Dispatch Incident to a Team

The standard life cycle for Incident tickets in tbITSM does not allow to assign a ticket to a Team without assigning it to a specific Person inside this Team. By creating a new state *dispatched*, this extension allows to *dispatch* an Incident ticket to a Team, without assigning it to particular Person. From the *dispatched* state the Incident can then be assigned to a Person.

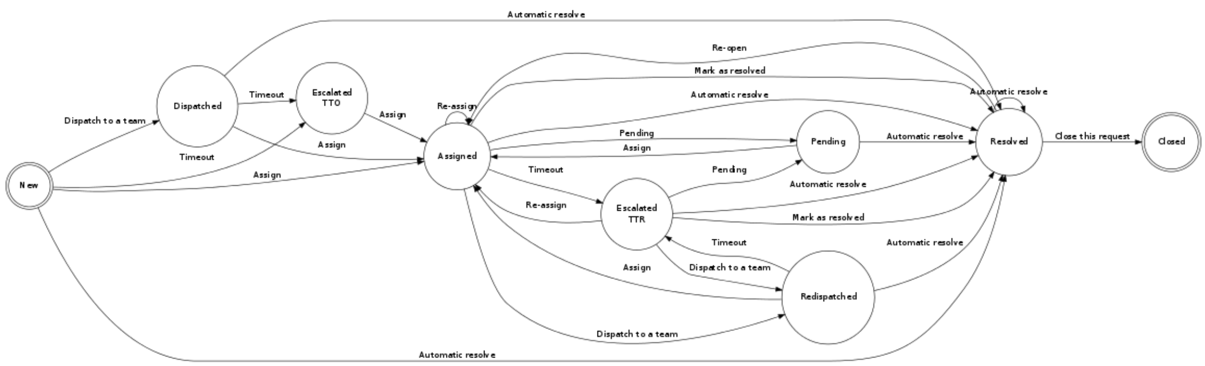
19.1. Features

Dispatch an Incident to a Team before assigning it to a Person.

The standard Incident life-cycle is the following:



Once the extension has been installed, the Incident life-cycle becomes as follows:

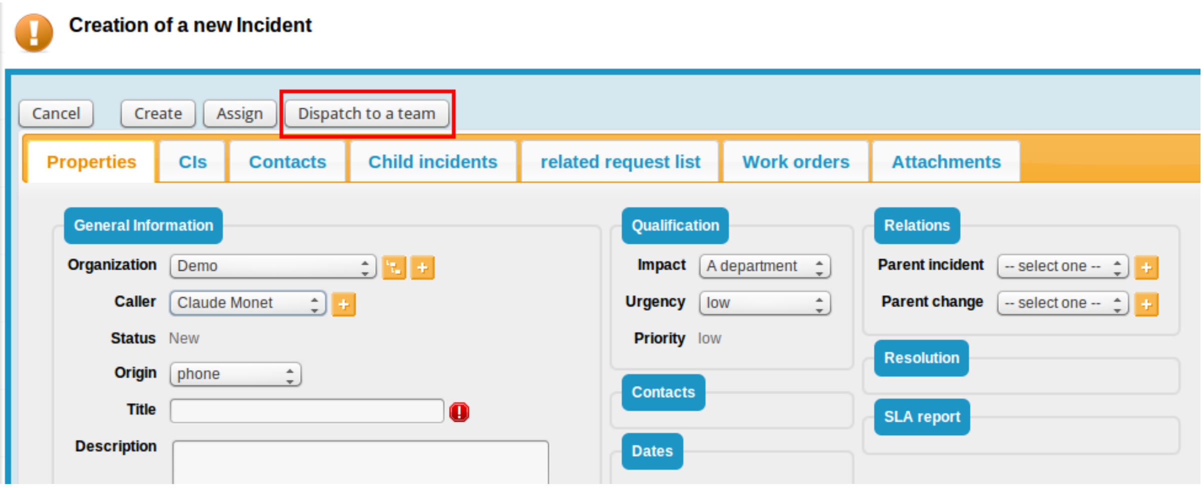


The definition of the TTO (Time To Own) metric is also modified to take into account the “dispatched” state (The Time To Own stops when the ticket is actually assigned to a Person, not only dispatched to a Team).

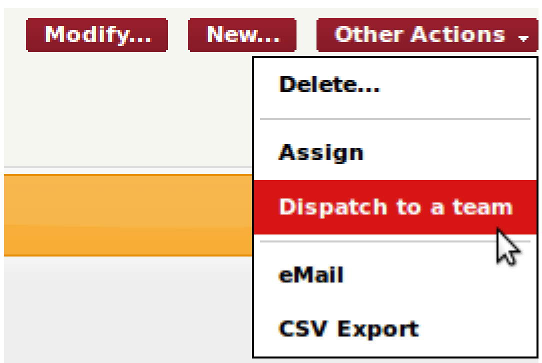
Actually, two new states are added to the life-cycle: *dispatched* and *redispatches*, to take into account the first assignment of a ticket. When a ticket which was assigned to a Person is dispatched again to a Team, the agent to which the ticket was assigned is cleared (even if the ticket is dispatched again to the same team).

19.2. Usage

The new *dispatched* state is fully integrated with the life-cycle of the ticket. For example, when creating a new Incident, an extra button “Dispatch to a team” appears:



On an Incident in state *new*, the action "Dispatch to a team." is available in the drop-down list of actions.



A user must have either the profile "Service Desk Agent" or "Support Agent" (or Administrator) to be allowed to dispatch a ticket.

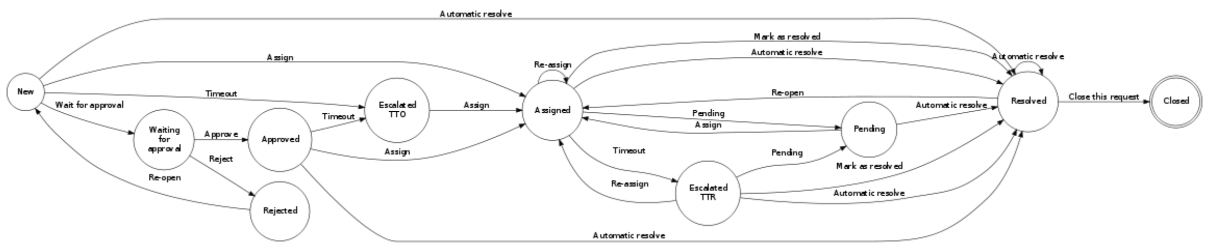
20. Dispatch User Requests to a Team

The standard life cycle for User Requests tickets in tbITSM does not allow to assign a ticket to a Team without assigning it to a specific Person inside this Team. By creating a new state *dispatched*, this extension allows to *dispatch* a User Request ticket to a Team, without assigning it to particular Person. From the *dispatched* state the User Request can then be assigned to a Person.

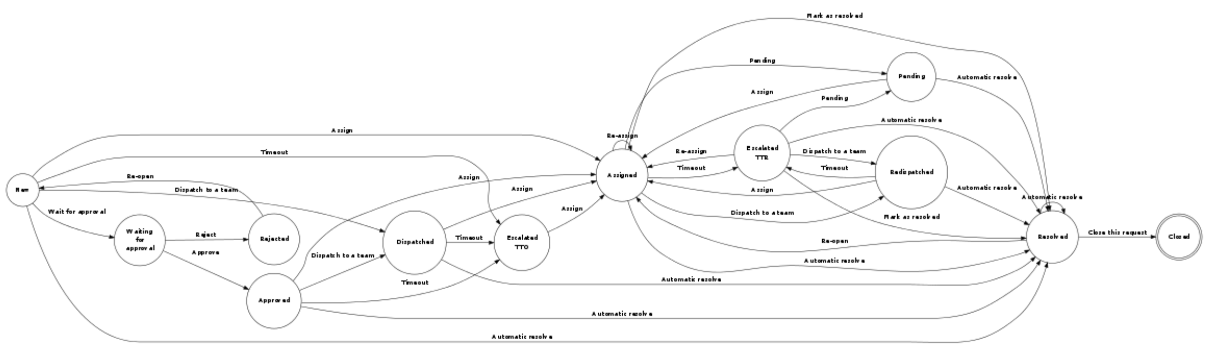
20.1. Features

Dispatch a User Request to a Team before assigning it to a Person.

The standard User Request life-cycle is the following:



Once the extension has been installed, the User Request life-cycle becomes as follows:

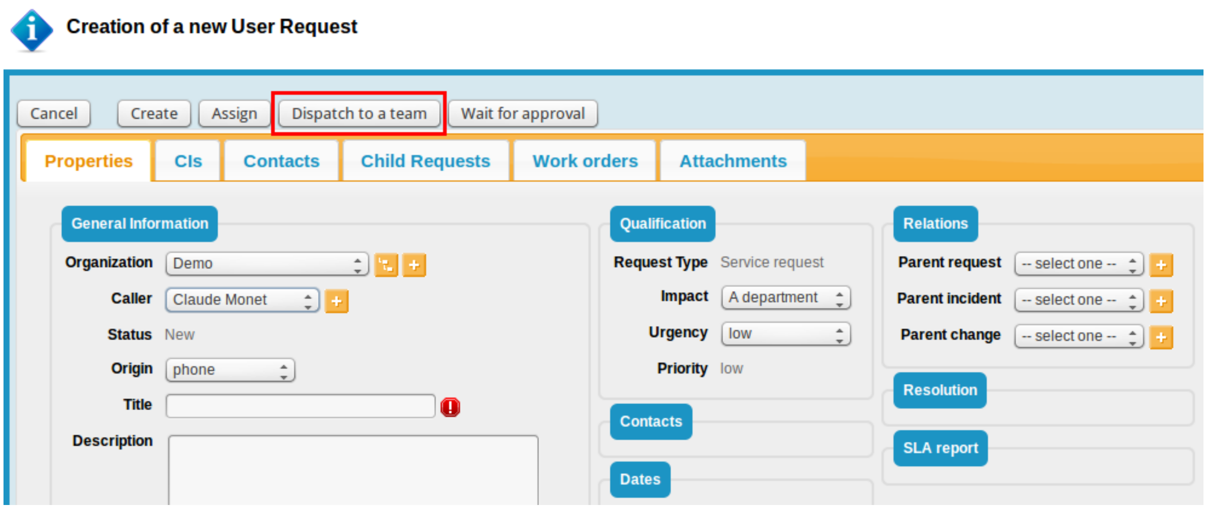


The definition of the TTO (Time To Own) metric is also modified to take into account the “dispatched” state (The Time To Own stops when the ticket is actually assigned to a Person, not only dispatched to a Team).

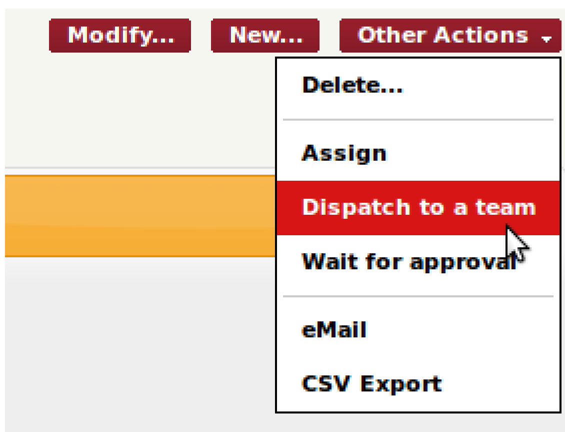
Actually, two new states are added to the life cycle: *dispatched* and *redispatches*, to take into account the first assignment of a ticket. When a ticket which was assigned to a Person is dispatched again to a Team, the agent to which the ticket was assigned is cleared (even if the ticket is dispatched again to the same team).

20.2. Usage

The new *dispatched* state is fully integrated with the life-cycle of the ticket. For example, when creating a new User Request, an extra button “Dispatch to a team” appears:



On a User Request in state *new*, the action “Dispatch to a team.” is available in the drop-down list of actions.



A user must have either the profile “Service Desk Agent” or “Support Agent” (or Administrator) to be allowed to dispatch a ticket.

21. Mail to Ticket Automation

This runs in the background to scan the defined mail inbox(es) and either create or update tickets based on the content of the incoming emails.

On Ticket creation, it fills the **description** of the ticket with the **content** of the email, set the caller, the customer, copy attachments, add contacts and many other fields.

On Ticket update, it extracts as best as possible, the **last reply part** of the email to update the **Public Log** of the ticket, copy attachment, change the status of the ticket and add contacts.

21.1. Features

- Determine if the sender is an existing Person (found by its email), then based on configuration it can reject the email in error or create a new Person.
- Determines whether a Ticket must be created or updated based either on the custom headers added by tbITSM in the email (in case of replies) or based on a configurable pattern in the title
- Connect to any mailbox using either the POP3 or IMAP protocol
- Interactive configuration of the mail inboxes
- Processes the incoming emails in both HTML or plain text format
- Support both Incidents and User Request tickets
- Ticket's attachments are automatically turned into attachments of the Ticket (“dangerous” types of attachments can be excluded)
- Automatic detection of duplicate attachments (like signature images)
- Keeps the messages in the mailbox until the corresponding Ticket is either closed or deleted
- Manual retry in case an error occurs when processing an email
- Images embedded inside an HTML mail are displayed inline in tbITSM as well.
- Images “too small” (below configurable dimensions) are **not** imported as attachments (to exclude signatures)

- Images bigger than configurable dimensions can be resized automatically before uploading into tbITSM

- "Inline images" are displayed at a reduced (configurable) size and can be "zoomed-in" by clicking on them.

- Automatically reject "Autoreply" emails, based on a set of configurable patterns to be tested against the subject of the email

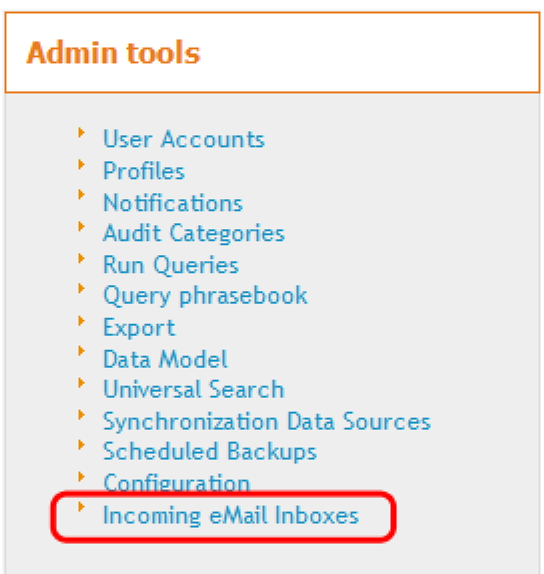
- Add the other recipients of the email (To: and CC:) as additional contacts on the ticket (configurable). The contact must already exist with that exact email, it won't be created, just linked to the Ticket.

- Apply a stimulus (configurable, depending on the state of the ticket) to change the state of the ticket upon reception of an email.

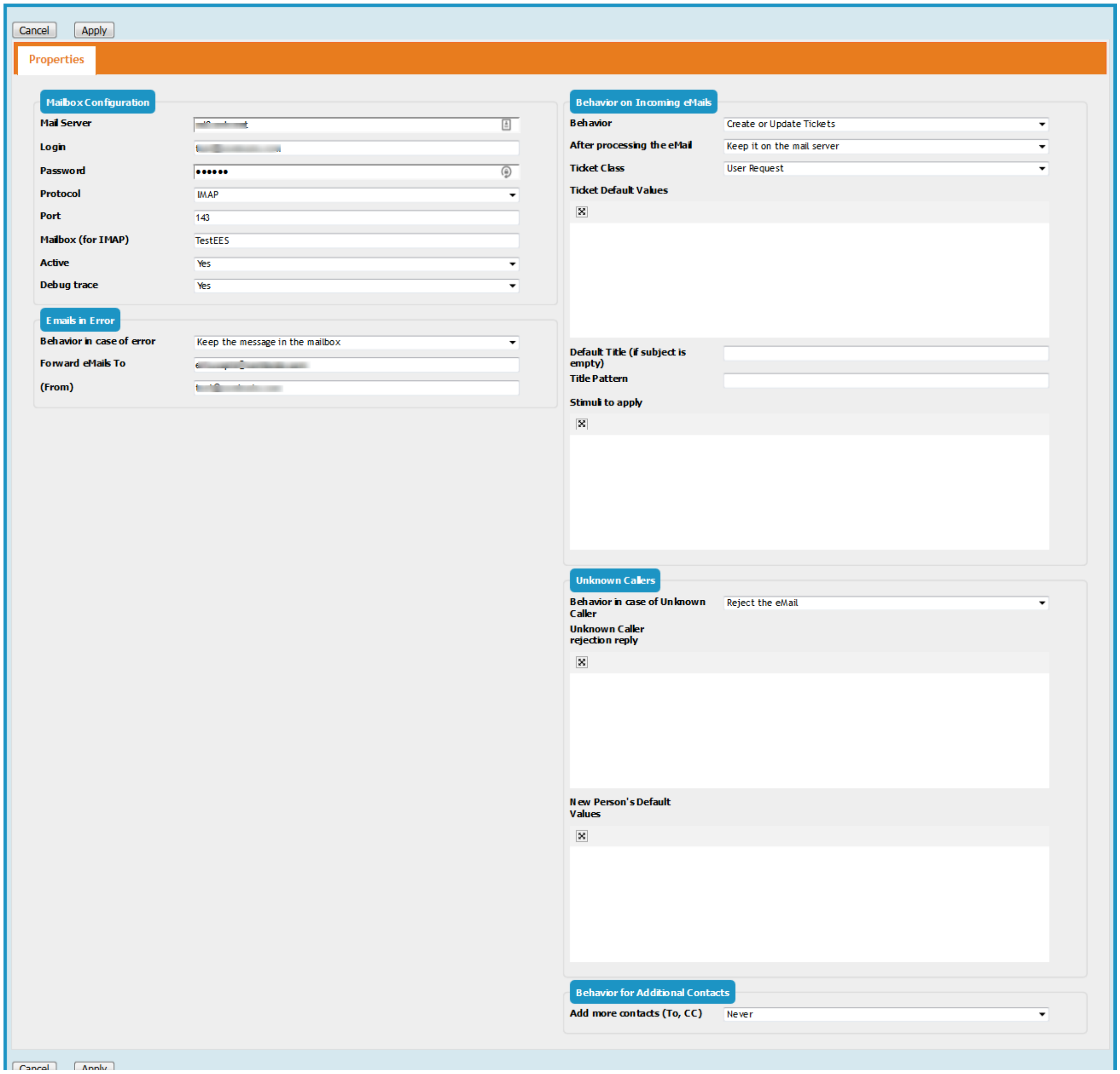
- Add a new type of Trigger: **Trigger (when updated by mail)** which allow to notify when a Ticket is updated through a received email.

21.2. Configuration

The behavior for each mailbox (which messages to process and whether to create or update a Ticket) is managed via the menu "Incoming email Inboxes" in the "Admin tools" section:



Click, "Create a new Mail Inbox" to create a new configuration for a mail inbox. This displays the following form:



21.2.1. Mailbox Configuration

The “Mailbox configuration” defines how the application connects to the mail inbox:

Field	Meaning	Sample Value
Mail Server	The IP address or fully qualified hostname of the mail server	10.153.20.142 or pop3.mycompany.com
Login	The name of the mail account used for connecting to the mailbox	test@mycompany.com
Password	The password for the above-mentioned account	
Protocol	The protocol to connect to the mail server: either POP3 or IMAP. If you need to use IMAP with SSL or TLS, refer to the	POP3

	imap_options configuration parameter below.	
Port	The TCP port to connect to the server. The standard values are 110 (secured: 995) for POP3 and 143 (secured: 993) for IMAP	110
Mailbox (for IMAP)	The IMAP mailbox (folder) to scan for incoming messages. If omitted the default (root) mailbox will be scanned. This option is ignored when using the POP3 protocol.	INBOX.Folder. Subfolder or INBOX/Folder/Subfolder
Active	If set to "Yes", the inbox will be polled. Otherwise, no.	Yes
Debug trace	Use this setting for tracing all the background operations related to this inbox for debugging and troubleshooting purposes. It is not recommended to activate this option for long periods on production since it tends to generate a lot of output which slows down the server	No

21.2.2. Emails in Error

This section defines the behavior when an incoming email cannot be processed properly. The email can be either kept in the mailbox (and remembered as an "Error" and no longer processed) or deleted immediately from the mailbox. Furthermore, the original message can be forwarded to an administrator (as an attachment) along with some explanation about the cause of the error.

Field	Meaning	Sample Value
Behavior	Whether or not the emails processed with an error should be kept in the mailbox. If so, the message will be flagged as "Error" and no longer processed, but still available for reading from the mailbox.	Keep the message in the mailbox
Forward emails to	The email address to which to forward the email when an error occurs. The forwarded message contains some explanation about the error and the original email as an attachment. If this address is left empty, the incoming emails which cannot be processed will simply be deleted from the inbox without further notice.	tbITSMadmin @mycompany.com
(From)	The IP address to be used as the "sender" of the error notification. For security reasons, many mail servers do not relay messages if the sender address is not a known address.	tbITSMadmin @mycompany.com

21.2.3. Behavior on Incoming emails

This section defines the behavior of the application when processing incoming emails.

Field	Meaning	Sample Value
Behavior	The behavior when a new message arrives in the inbox. The possible values are: **Create or Update**: create a new Ticket or Update an existing one if a matching Ticket is found **Create new ticket**: each new message creates a new Ticket **Update existing tickets**: all incoming messages which do **not** match an existing Ticket are treated as errors.	Create or Update
After processing the email	The action to be taken after successfully processing an incoming email: either keep the email on the mail server (until the associated ticket is closed or deleted), delete the email immediately or move it to another folder (on IMAP server only).	Keep the email on the mail server
Target folder	Is mandatory in case of move after processing the mail. Define the target folder.	INBOX.Folder. Subfolder or INBOX/Folder/Subfolder
Ticket Class	The class of Tickets to create or update when receiving an email. Make sure that you select a valid class for your tbITSM configuration.	User Request
Ticket default Values	The syntax for "Ticket Default Values" and "New Person's Default Values" are: - one field to initialize per line - <field_code>:<constant_value>	service_id: Networking
Default Title (if subject is empty)	The value to be used as the title of the Ticket, if the subject of the incoming email is empty. If this field is left empty the system will supply a default value ("No Subject")	Empty subject
Title pattern	Each notification sent by the application contains a reference to the "source" Ticket in the MessageID field of the email. Email client applications generally store this identifier in the "in-reply-to" or "references" header of the reply email. This is the primary mean of identifying that an email message is related to a ticket. If this header is not present in the incoming message, the application can parse the "subject" field to look for a given match. This pattern determines how to parse the subject. The pattern specified here must follow the PCRE syntax.	/R-([0-9]{6})/
Stimuli to apply	A list of state_code: stimulus_code (one per line) to define the stimulus to apply (after updating a ticket), for the given state of the ticket. This is useful for example to automatically reassign a ticket which is in the state "pending".	pending: ev_assign

21.2.4. Unknown Callers

This section determines the behavior of the application when the sender of an email (From:) does not correspond to a known email address in the application. There are two possibilities:

- **Reject the email:** the incoming email is treated as an error and thus either forwarded to an administrator or deleted.

- **Create a new person:** a new Person will be created based on the email of the sender and constant values defined below.

Field	Meaning	Sample Value
Behavior in case of Unknown Caller	What to do when the sender of the incoming email message does not correspond to any Person recorded in the application	Create a new person
Unknown Caller rejection reply	Optional reply to sender when the unknown caller behavior is set to “Reject the email” (no message is sent when left empty)	empty
New Person's Default Values	Default values for initializing the new Person. The application automatically fills the email field with the email address of the sender of the message. All other mandatory fields must be initialized with constant values provided here, otherwise the creation of the new Person will fail.	first_name: Unknown name: Caller org_id: Demo

21.2.5. Behavior for Additional Contacts

This section determines the behavior of the application regarding the additional recipients of the incoming email (persons in To: and CC: of the message). It is possible to specify if/when the email addresses which correspond to a valid contact in tbITSM are added to the ticket (via the Contacts tab). Email addresses which do not correspond to a valid contact in tbITSM are *always* ignored.

Field	Meaning	Sample Value
Add more contacts (To, CC)	Whether or not to add the To: and CC: email addresses as additional contacts to the ticket. The possible values are: * Never: no additional contact will be added * Always: additional contacts will be added when creating and updating a ticket * When creating a ticket: additional contacts will be added only when creating a new ticket * When updating a ticket: additional contacts will be added only when updating an existing ticket	Never

21.2.6. Other configuration parameters

In addition to the configuration performed by creating a Mail Inbox object using the user interface of the application, a few parameters are available in the configuration file to fine tune the behavior of the application.

The parameters listed below apply to **all** the Mail Inboxes

Parameter	Meaning	Default Value
debug	Set to true to turn on debug output	false
periodicity	Interval (in seconds) at which to check for incoming messages	30
body_parts_order	Comma separated, ordered, list of MIME types, determining the preferred part of the message to retrieve for populating the description or public_log of the Ticket. In order to import as inline images, the images embedded in HTML, the HTML part of the email must be processed in priority over the text part. Therefore, the recommended configuration is text/html, text/plain.	text/html, text/plain
pop3_auth_options	POP3 authentication options. Possible values are: 'CRAM-MD5', 'APOP', 'PLAIN', 'LOGIN', 'USER'	USER
imap_options	Additional IMAP options. Possible values are listed here: [IMAP flags](). For example to use SSL you could specify : array(0 ⇒ 'imap', 2 ⇒ 'ssl') **Warning**: Do NOT use the pop3 flag to connect to a POP3 mailbox using the PHP IMAP extension. Due to a limitation of the IMAP extension this will not work! If you want to connect to a POP3 server, use the POP3 protocol instead.	array('imap')
exclude_attachment_types	Array of MIME types to exclude when retrieving attachments	array('application/attachment')
html-tags-to-remove	**new in 3.0.0** Used for computing the "new part" of HTML messages by removing the specified tags. The syntax is an array of tag_name => array of CSS class names.	*see above*
maximum_email_size	If an incoming email is bigger than the specified size, the message will be saved to the big_files_dir if it is configured and deleted from the inbox. A notification message will be sent to the administrator as for other "errors" happening when processing the mail inbox. The size can be specified using the 'short' notation: 100K, 3M, 2G… If set to zero, no limit will be enforced… with the risk of a PHP crash if there is not enough memory to decode an incoming email.	10M
introductory-patterns	**only for plain text emails** When computing the "new part" of a	*see above*

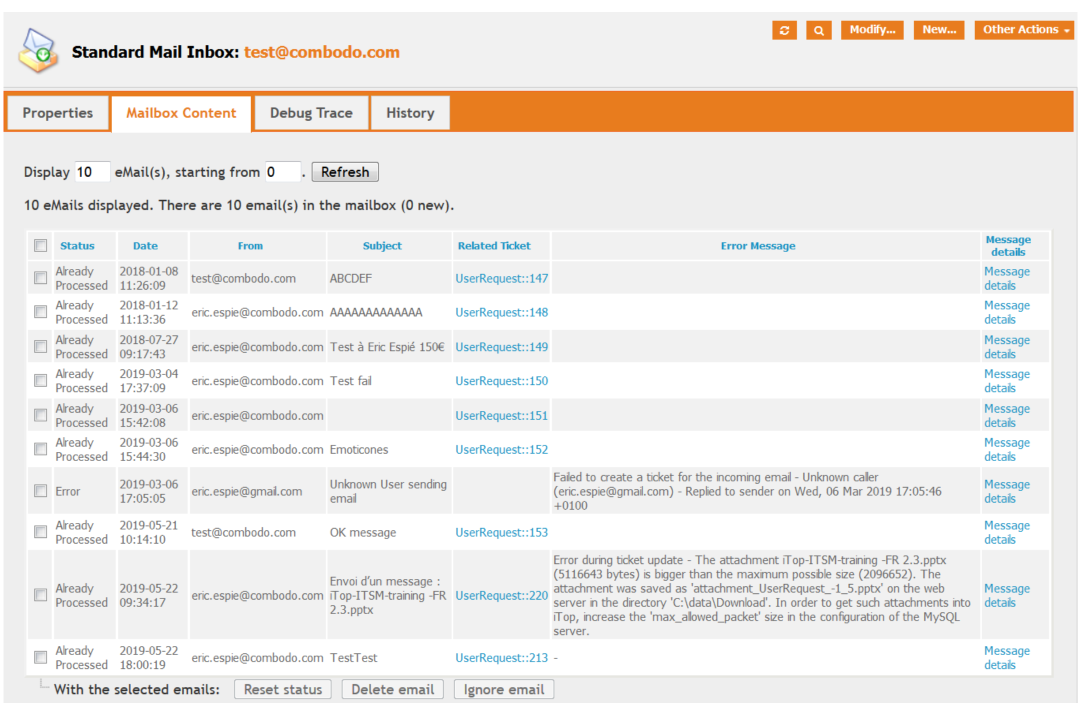
	message, lines matching this pattern and preceding what looks like an "old part" of the message, are removed. Adapt this list to your localization… and favorite email client dialect. The pattern specified here must follow the PCRE syntax.	
multiline-delimiter-patterns	**only for plain text emails** and **only for mail for ticket update** Multi-line regular expression patterns used for computing the "new part" of a message. each of these patterns determine the beginning of the "old part" of a message, when a message is a "Reply" to another message. Everything that matches this pattern (and all the text that follows this match) will be removed for the "new part". All patterns are tested successively. The pattern that provides a match closer to the beginning of the text will be used. Adapt this list to your localization and to the dialect of your favorite email clients… Patterns must follow the PCRE syntax.	*see above*
delimiter-patterns	**only for plain text emails** and **only for mail for ticket update** this regular expression patterns are used for detecting the lines beginning "old part" of a message, in case none of the multiline-delimiter-patterns did match. Patterns must follow the PCRE syntax.	*see above*
big_files_dir	The path to a directory where to store emails bigger than maximum_email_size. If this directory is not configured, the emails are simply deleted before notifying the administrator.	
use_message_id_as_uid	Boolean. For IMAP connections only. Whether or not to use the identifier from the message (MessageID) instead of the Mailbox' unique identifier (UID) to uniquely identify the already processed messages. This can be useful to work around problems when the UID of the messages on the server changes between sessions (like with Gmail). If you toggle this value, make sure that you first empty the mailbox (and stop the cron job), since all messages present in the mailbox when the setting is changed will be considered as new and processed again.	false
images_minimum_size	Minimum dimensions for importing images. Images smaller than the given dimensions will be ignored and not imported as attachments. The dimensions are expressed as a string *widthxheight* (where *width* and *height* are integer numbers, in pixels).	100×20
images_maximum_size	Images bigger than these dimensions (for example 1000×1000) will be resized to fit in the given dimensions. The dimensions are expressed as a string *widthxheight* (where *width* and *height* are integer	

	numbers, in pixels). **Note** this feature is available only if PHP GD is installed. If no dimensions are given, the images are never resized.	
undesired-subject-patterns	An array of regular expression patterns (as PHP text strings) that will be used to test the subject of the incoming email. If any of these pattern matches, the email will be considered as “undesirable” and rejected (the same processing as for any other error case will then be applied). The patterns specified here must follow the PCRE syntax.	array ()
undesired-purge-delay	Delay in days to remove automatically the undesired messages (0 means that the messages are removed immediately)	7
recommended_max_allowed_packet	Display a warning on the 'Mailbox Content' screen if the database parameter 'max_allowed_packet' is less than the one configured	10*1024*1024
retention_period	Define the number of hours we keep the replicas after the disappear of the source message. This parameter can protect replicas when connection with source is lost.	1

When specifying PCRE patterns inside the configuration file, make sure that you double the backslash characters, since backslashes must be escaped inside PHP literal text strings.

21.2.7. Checking the connection

Once the Mail Inbox object has been created, you can use the tab “Mailbox Content” in the details of the object to check that the application can properly connect to the mail server and retrieve messages from it.



To deeply inspect the content of the mailbox it is always better to use a real mail client application. The view provided in this tab is just use to help troubleshooting connection problems.

This view also allows to perform two different kinds of actions on a set of messages:

- “Reset Status”: for messages which are either flagged as “Error” or “Already processed”, this status will be reset and the email will be considered again as “New” and thus candidate for processing the next time cron.php runs.
- “Delete email”: deletes the message(s) from the mailbox. No confirmation will be asked!!
- “Ignore email”: mark new messages as “ignored” to avoid processing them.
- Debugging

22. Notify on Expiration

This extension allows to trigger a **notification** when a date is reached:

- It can be used for a **Contract**, a **License** or any object with a date.
- The notification delay is configurable for each class of object. You can set a 3 months' notice on Contracts and only 30 days on Licenses for example.
- Multiple notifications with different delays can be defined for a given class. Therefore, you can have a first notification 45 days before contract expiration, another one 15 days later and maybe a last notification 5 days before the deadline.

22.1. Features

This extension allows to trigger notification when an expiration date is about to be reached.

- You can define various Expiration rules, for different usage. Targeted for **Contract & License** expiration, it can be used with any class having a date attribute.

- You can set multiple rules for the same class with different term of notice, for example one '3 months ahead' and another 'one month before' the deadline. You can even create a third one '5 days after' the deadline if you want.

- Notification for a particular object reaching Expiration, unless configured differently, will occur only once.

- Notification message is configurable as any standard trigger/action.

- For advanced need, you can define the objects in scope of the notification with an OQL.

22.2. Usage

Example: Setup a notification 20 days before a License expires

- Create an Expiration Rule
- Create a Trigger
- Create a Notification

22.3. Expiration Rule

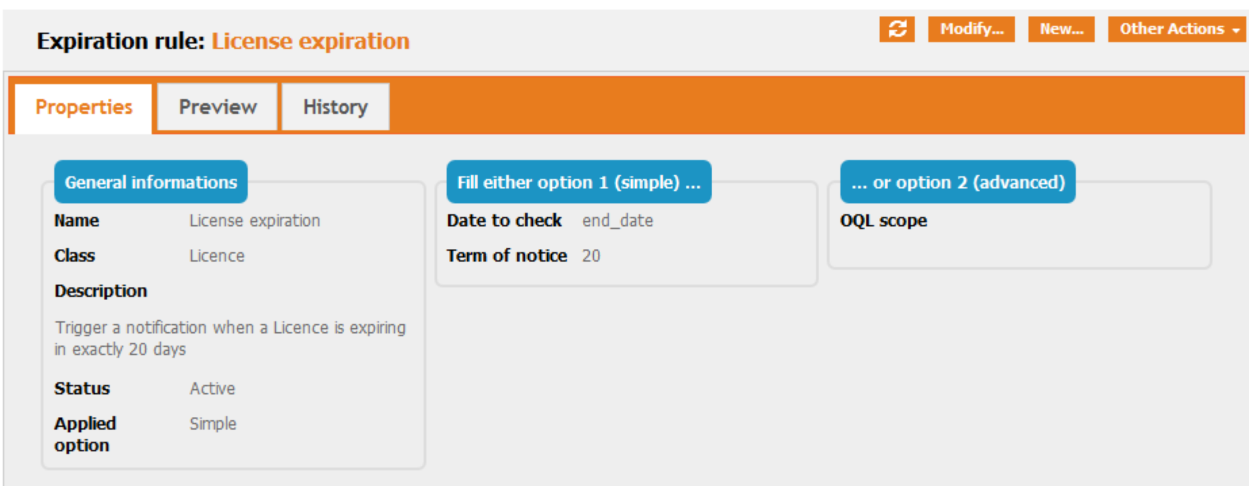
The extension brings a new Menu entry Expiration rule in the "Service Management" category

To create a new one:

- Give it any **Name** you like,
- Define the **Class** on which the rule will be applied, a Class code is expected, for e.g., Customer Contract
- Give it a **Status**: Active to enable the cron to execute this Expiration rule in the Background task.
- Choose an **Applied option** by filling the fields either in **…option 1 (simple)** or in **…option 2 (advanced)** field set.

22.3.1. Simple

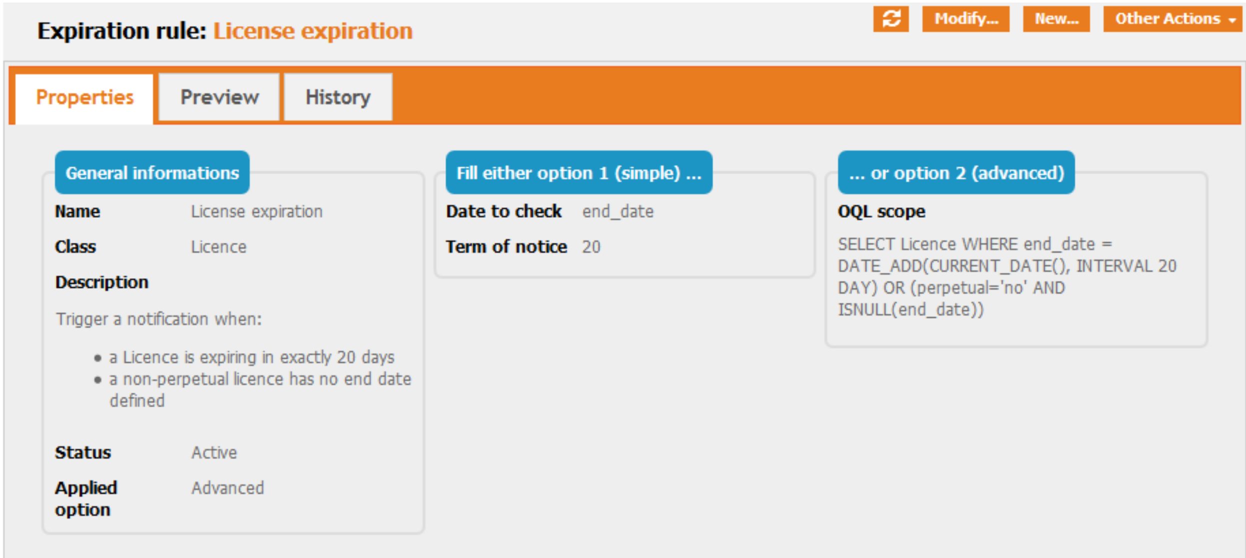
- Define the **Date to check**, which must be the code of a date attribute, *not the label*.
- **Term of notice** is configurable and says when to trigger the notification, by specifying a number of days **before** the deadline.



22.3.2. Advanced

- Define the **OQL scope** with an OQL query returning the objects on which to apply the trigger. *As soon as an OQL is entered, the advanced option is used, even if the simple one is also documented.*

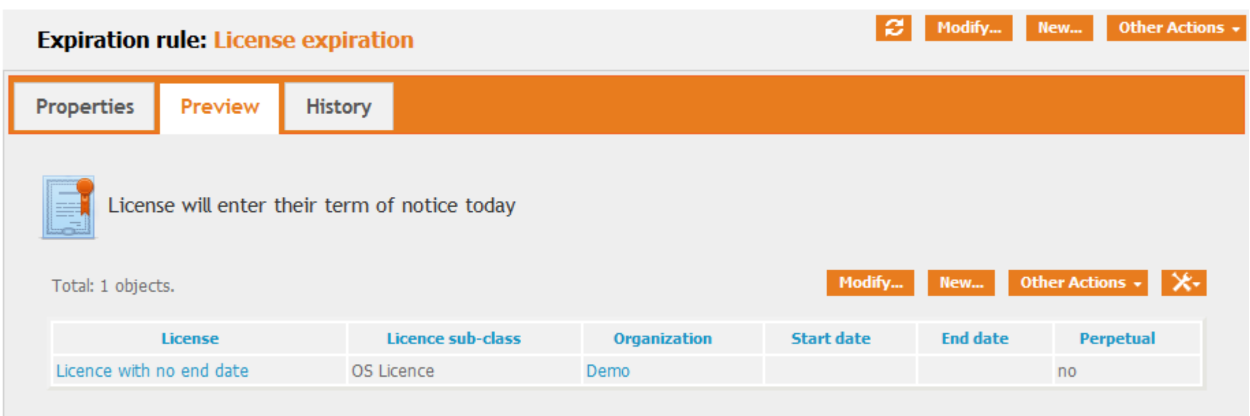
- With the advanced mode, you can trigger notification after a date, which is not possible in simple mode.



You can create as many Expirations Rule as you want, for License, Customer Contract and even multiple for the same class but with a different term of notice

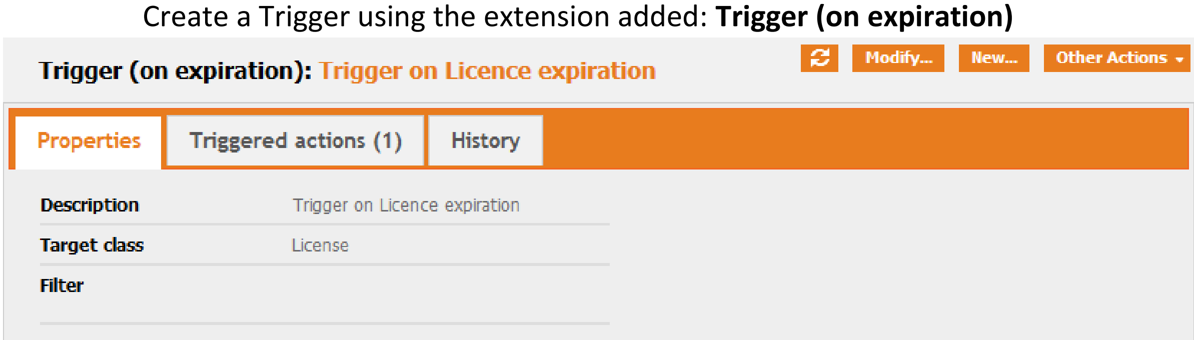
22.3.3. Preview

You can check anytime which objects would be trigger if the Expiration Rule is run.



22.3.4. Trigger

Create a Trigger using the extension added: **Trigger (on expiration)**

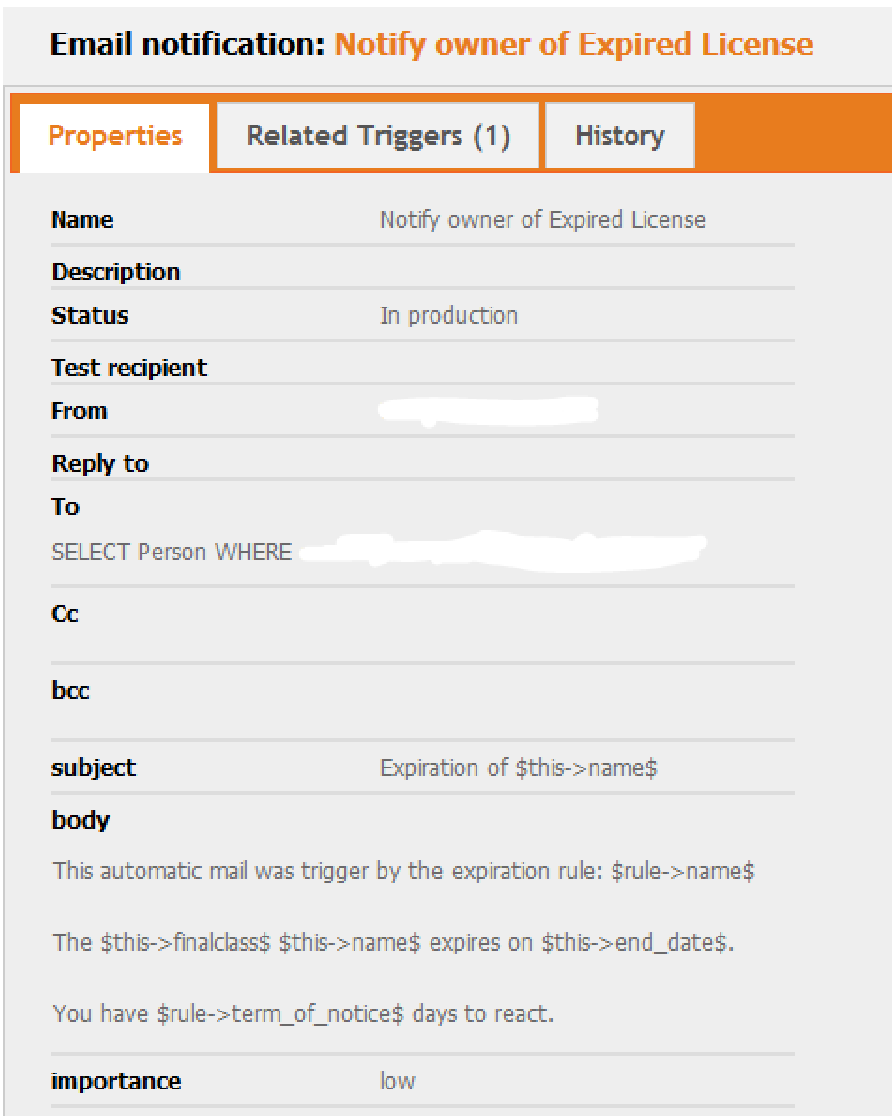


22.3.5. Notification

Create a Notification, defining who should receive it and the body of the message

On top of the standard placeholders, this extension brings also:

Placeholder	Purpose
$rule->name$	The name of the **Expiration rule** which has trigger this notification
$rule->description$	The **Description** *of the Expiration rule which has trigger this notification*
$rule->term_of_notice$	The **Term of notice** in days *defined in the Expiration rule which has trigger this notification*



Link the Notification to the above created Trigger.

23. Send Updates by Email

This extension allows to send email notifications when a ticket case log is updated in the console. Support agents can communicate via email with the callers directly by just updating the log. Used in conjunction with "Mail to Ticket Automation" extension, users can reply directly to the email, otherwise they have to use the portal to update the ticket.

23.1. Features

This extension allows to send an email when updating the case log (either the public log or private log) of a Ticket. Email reply works with any type of Ticket. Attachments added while editing the ticket are automatically sent as attachments to the email.

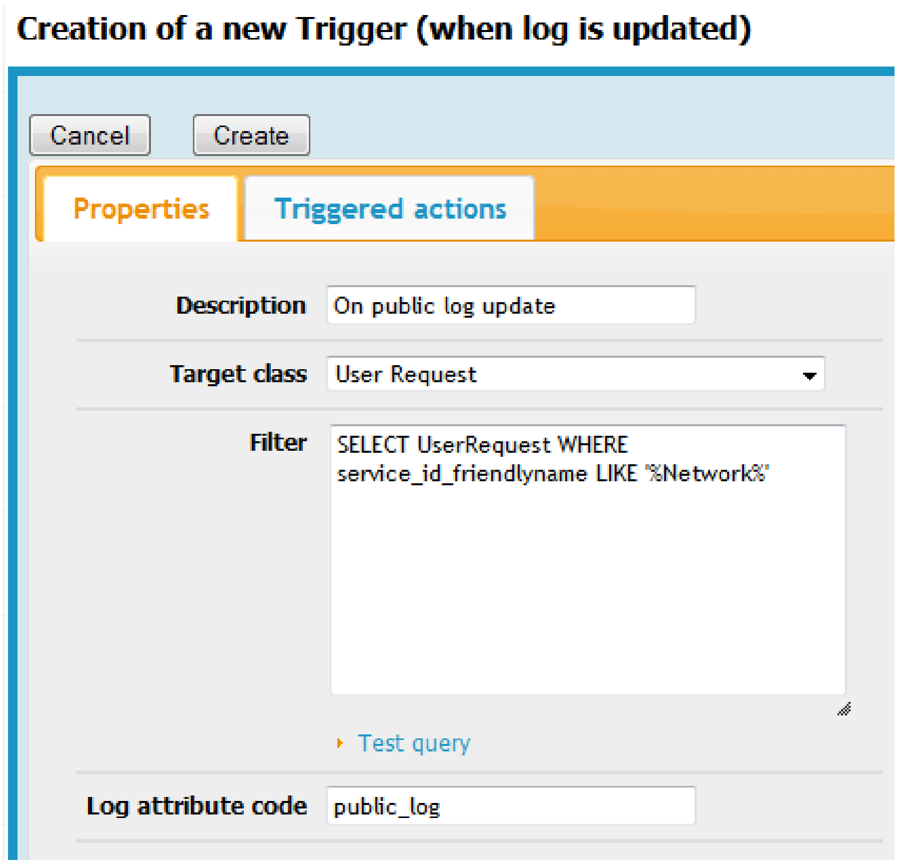
It allows an administrator to define notifications in order to inform contacts (for instance the caller, or the agent) when a case log is updated on a Ticket. Thus, allowing support agents to communicate with the callers directly by updating either the public log or the private log of the ticket.

The configuration relies on a specific type of Trigger ("on log update") and the usual Email Actions. The definition of the Trigger object determines which field of the ticket (public_log, private_log…) is used for the feature.

The user can select the attachments to be sent with the message. Newly added attachments are automatically selected but can be manually deselected if needed.

23.2. Configuration

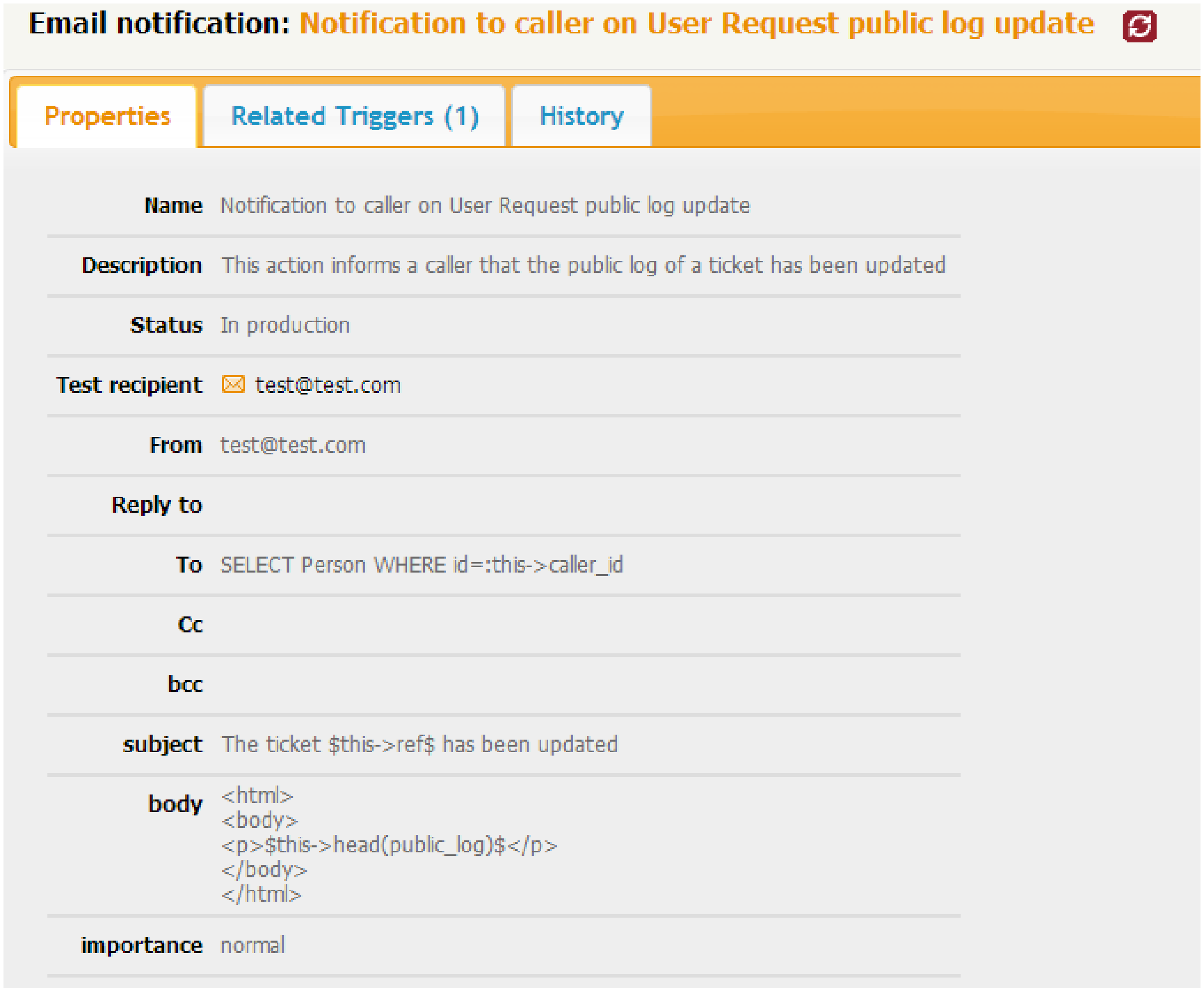
From the *Admin tools/Notifications* menu, create a trigger "when log is updated", and select the expected object class and case log attribute code:



The optional argument 'Filter' is used to specialize the trigger. The filter is an OQL specifying which objects will be taken into account by the trigger. In the given example, the created trigger will be activated only for User Request having a Service named like “…Network…”.

Create an email action for this trigger. This email action can be configured as follows:

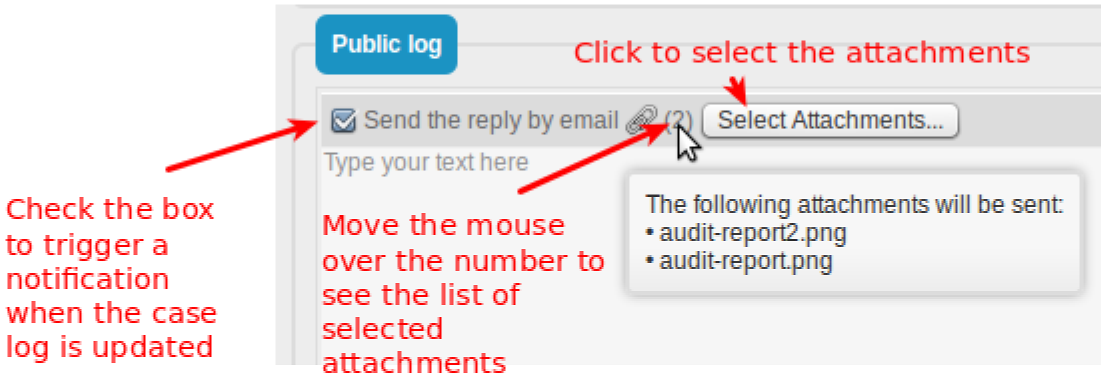
- Status = any active state (e.g., 'In production' or 'Being tested')
- TO = SELECT Person WHERE id =: this->caller_id
- Body = Dear $this->caller_id_friendlyname$, … $this->head(public_log) $
- plus, other fields required to send notifications



23.3. Usage

Once this extension is installed and configured, you just have to modify the case log of a ticket to use it.

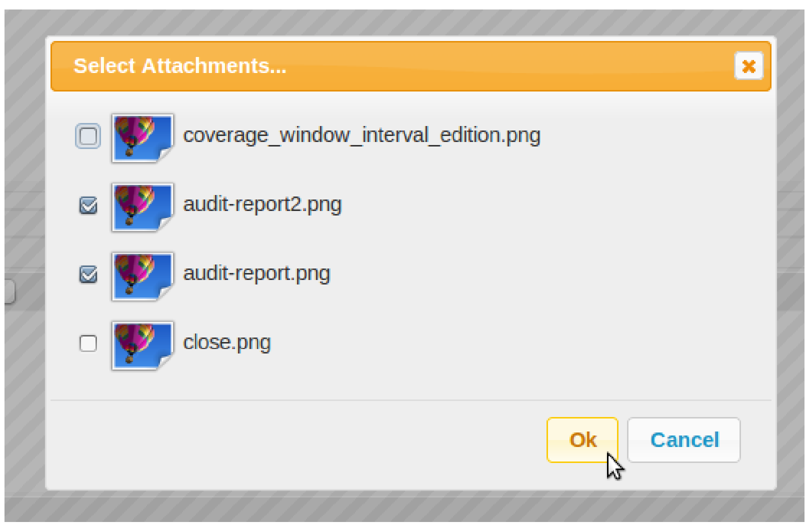
For instance, if you modify a User Request, a small check box and paper clip icon appear on top of the public log to define if the public log update should trigger the notification or not:



The default behavior of the check box is configured in the tbITSM configuration file. If you uncheck it, no notification is sent.

All new attachments added during this modification of the Ticket will be sent along with the notification. As soon as a new attachment is added to the Ticket (while it is being edited) the number next to the paper clip increases. Moving the mouse over the paper clip icon displays a small tooltip with the list of files that will be sent as attachments with the notification.

The user can select the attachments to be sent along with the reply, by pressing the "Select Attachments..." button. This displays the following dialog box:

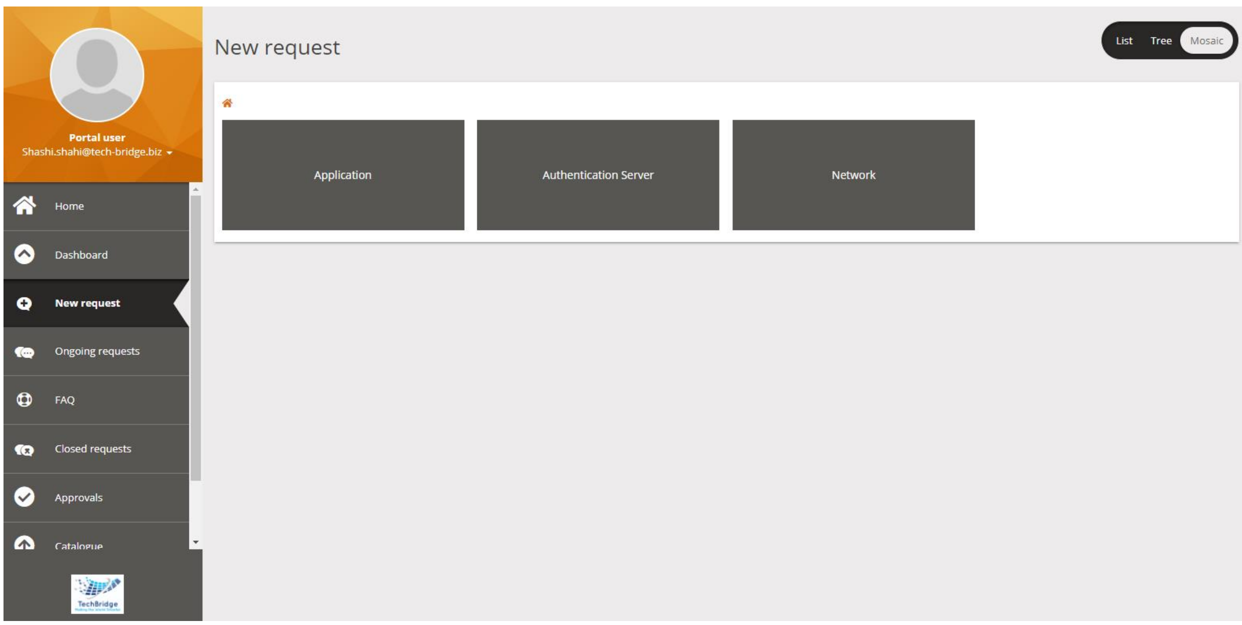


Click on the check-box in front of each desired attachment then click "Ok" to validate the choice.

24. Service Catalog Mosaic View

This extension changes the default view mode of the service catalog within the portal, to mosaic.

- This mode is much more efficient and user friendly if you have a small number of services and service subcategories.
- In this case, you can display icons, which if well chosen, are understood more rapidly than a text.
- Navigation on small devices is also easier with this mode.



25. SLA considering Business Hours

With this extension, computation of SLA deadlines (TTO and TTR) takes into account time periods during which you committed to deliver a given service to a given customer. It takes into account as well Bank Holidays for your company. Without this extension, SLA computation is based on 24×7 delivered services and no holidays.

Coverage Windows & Holidays

- You can define the *holidays* which apply to your company (days which are closed for business, e.g., January 1st, May 1st…).
- You can define *coverage windows* applying to your services (i.e., working hours within each day of the week, maybe excluding a lunch break: 8-12h and 14-18h on Monday, Tuesday, Wednesday, Thursday and Friday).
- You can choose by customer and service, which coverage window applies (e.g., Business Hours vs 24*7).
- Time To Own (*TTO*) and Time To Resolved (*TTR*) metrics are computed according to *coverage windows* excluding *holidays*.
- This extension can be used even if you have customized the Service Management Data Model, assuming you have a mean to determine on each Ticket the applicable Service Level Target (*SLT*).

25.1. Features

By default, SLA computation assumes that a given duration does not depend on the time it starts (i.e., computations are based on a 24×7 service).

If you would like to compute SLA deadlines (TTO and TTR) depending on the customer/service and the times at which your support teams are working, then you will have to implement Coverage windows.

This extension allows you to define:

- Coverage windows (i.e., open hours within each day of the week), possibly as several intervals with the day (i.e., integrate a lunch break)
- Holidays (complete days which are closed for business, e.g., January 1st, May 1st…)
- How the Coverage Windows and Holidays are related to each Ticket.

Once the Holidays and Coverage Windows are created in the system, the TTO and TTR deadlines are computed according to the corresponding coverage windows and exclude the holidays.

The conversion from previous format(s) of Coverage Windows is automatic, during the installation of the module.

25.2. Configuration

Once the new module has been installed, the default configuration states that: The Coverage Window applicable to a Ticket is defined, for each Service, on the Customer Contract.

There is just one global list of Holidays. Any Holiday defined is applicable to any Ticket.

You can adjust the configuration parameters if you want to use different rules.

Parameter name	Description	Default Value
coverage_oql	The OQL to retrieve the Coverage Window object to apply to a given ticket. If the query returns several Coverage Window objects, **only the first one** will be used.	SELECT CoverageWindow AS cw JOIN lnkCustomerContractToService AS l1 ON l1.coveragewindow_id = cw.id JOIN CustomerContract AS cc ON l1.customercontract_id = cc.id WHERE cc.org_id=: this->org_id AND l1.service_id =: this->service_id
holidays_oql	The OQL to retrieve the Holiday objects to apply to a given ticket. **All** the Holidays returned by the query will be taken into account for the computation.	SELECT Holiday

The two OQL queries listed above are relative to a Ticket (the current ticket being processed). This means that all the placeholders: this->xxx refer to the corresponding fields of the Ticket.

You can also change the display format of the TTO and TTR deadlines by modifying the configuration parameter **deadline_format** in config-tbITSM.php with the following value:

```
'deadline_format' => '$date$ ($difference$)',
```

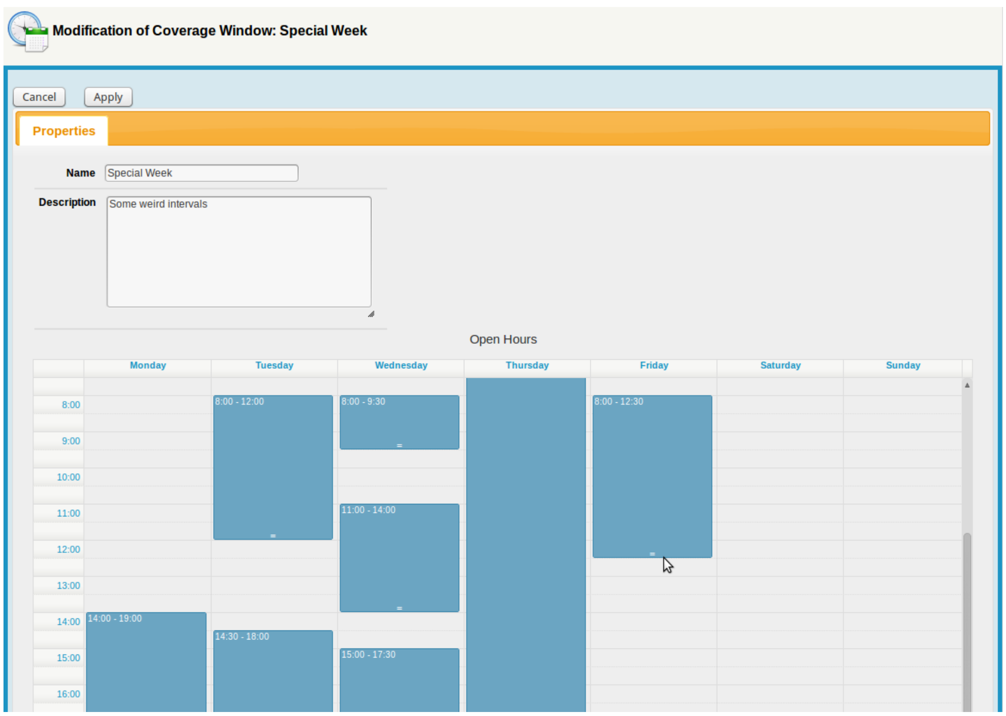
This will display the deadline date and the delay from now to the deadline.

25.3. Usage

25.3.1. Managing Coverage Windows

The menu “Coverage windows” in the module “Service management” displays all coverage windows defined in tbITSM. If none has been defined click on “Create a new coverage window”, otherwise click on “Create” to create a new one.

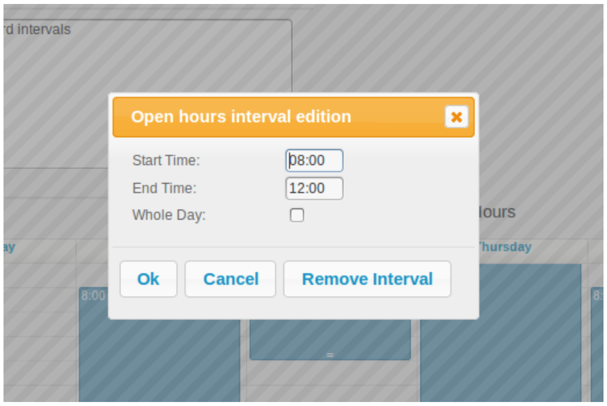
The open hours for a given “Coverage Window” object are represented as a set of intervals, for each day of the week. The area outside of the blue intervals is considered as “Off hours”.



The intervals are edited directly in the calendar at the bottom of the page:

- Click and drag over an empty area to create a new interval
- Drag the handle at the bottom of an interval to adjust its duration
- Drag the interval around to change its starting time
- Click inside an interval to open a dialog box in order to edit the interval's properties

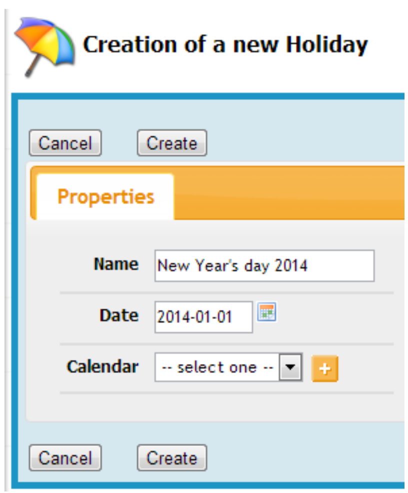
When clicking on an interval, the following dialog is displayed:



- Click on “Remove Interval” to completely remove an interval from the coverage window.
- All the modifications to the intervals are applied only when clicking on the “Apply” button to submit the modifications to the Coverage Window object as a whole.

25.3.2. Managing Holidays

The menu “Holidays” in the module “Service management” displays all holidays defined in tbITSM. If none has been defined click on “Create a new holiday”, otherwise click on “Create” to create a new one.

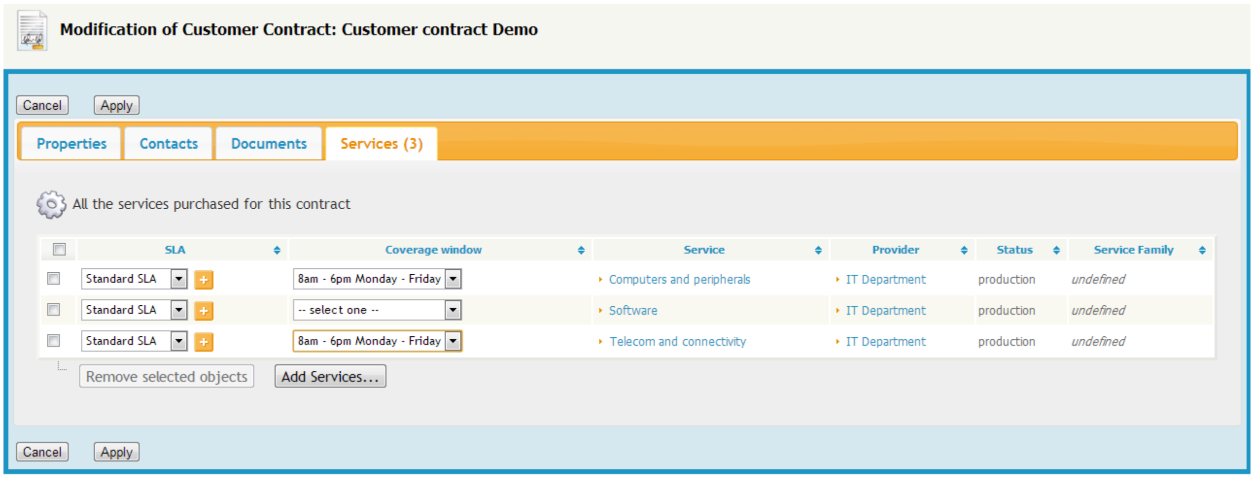


The Holiday Calendar allows you to group holidays (for example based on the country of your customers). This is a convenient way to simplify the queries required by a complex (multi-customer / multi-country) environment.

The OQL queries configured by default when installing the module do not use Holiday Calendars but Holidays.

25.3.3. Select coverage windows for a customer

To change the coverage window of a customer contract, simply modify the contract and in the tab “Services” select the appropriate coverage window required for each service:



By default, if no coverage window is selected, the computation of deadlines will be done using a 24h*7 coverage window.

26. User Actions Configurator

This extension allows an administrator to define new user-actions to simplify and automate processes. Each user-action adds a menu to create a new object, prefilled with information from the currently displayed object.

Among the possible usages, you can provide to your users:

- a mean to quickly clone a CI (clone all the attributes but still force the user to enter a new name).
- a shortcut to create a ticket from a Contact that will be the caller of the ticket.
- a shortcut to create a parent ticket (and record the information into the child ticket).
- a shortcut to create a ticket from a CI (and link the CI to the ticket).
- a shortcut to create a change ticket from a user request (and record the information into the user request).

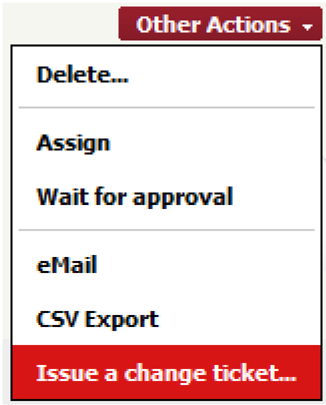
This extension aims at improving the end-user's productivity. It adds a menu to create an item prefilled with information coming from an existing item.

Amongst the possible usages, you can:

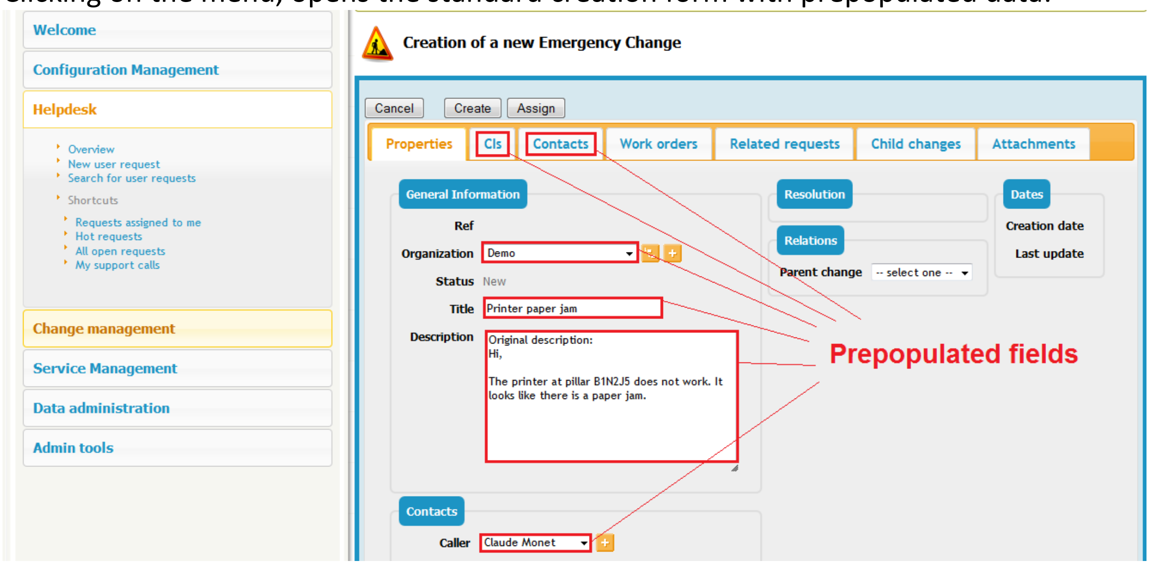
- Provide a mean to quickly clone CIs (clone all the attributes but still force the user to enter a new name)
- Provide a shortcut to create a ticket from a Contact that will be the caller of the ticket
- Provide a shortcut to create a parent ticket (and record the information into the child ticket)
- Provide a shortcut to create a ticket from a CI (and record the information into the child ticket)
- Provide a shortcut to create a change ticket from a user request (and record the information into the user request)

Let's use the last example to illustrate the possibilities of the module:

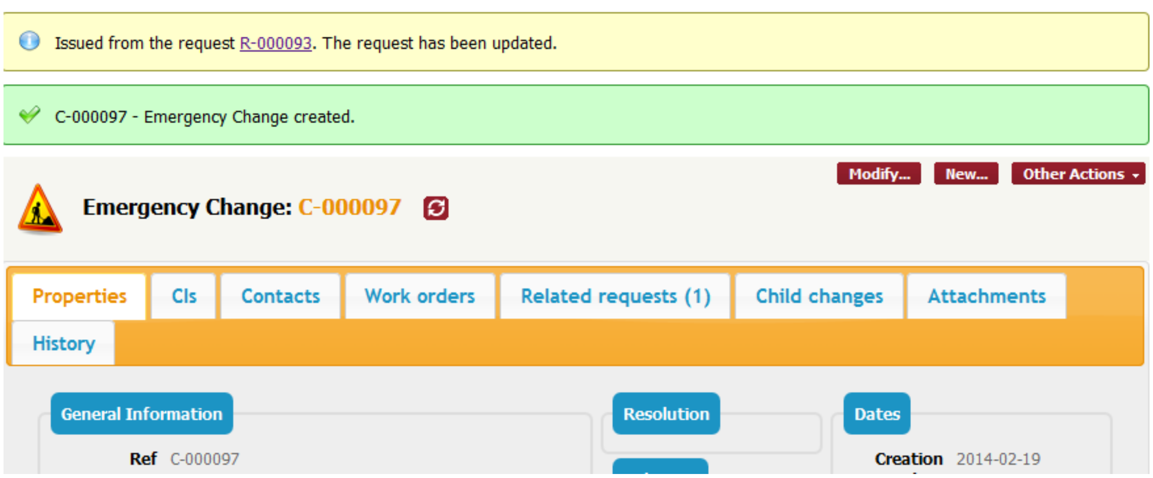
Users have a menu in the **Other Actions** drop-down menu of a *User Request*:



Clicking on the menu, opens the standard creation form with prepopulated data:



The user can adjust the values, then create the item. In addition to the standard report, a message indicates that something has been retrofitted back to the *source* item.



26.1. Principles

A shortcut is governed by a *rule definition*.

The menu is visible if a series of conditions are met:

- The item being visited is in the scope of the rule (given by an OQL). For instance, you may want to propose the shortcut only for user request being assigned or pending: SELECT UserRequest WHERE status IN ('assigned', 'pending')
- The current user has a profile allowed for the rule (can be anybody)
- Several rules can coexist for a single *source object*: there will be as many menus as there are rules that match the conditions.

A rule defines which kind of object to create:

- Either a given class (can be an abstract class, the user will have to select a subclass)
- Or the exact same class as the *source object*
- A rule specifies what has to be done to *preset* or *prepopulate* the form. This is done by the mean of *actions*. The following kind of actions are available:
- Set the value of an attribute depending on the value of one or several attributes from the *source object*.
- Copy N-N links (e.g., Documents on a Ticket, while cloning a Ticket. Documents are not duplicated, just the links are)
- Copy 1: n link by duplicating the 1-N linked objects (e.g., Server Interfaces, Work Orders on a Change)
- Add an N-N link to the *source object* or to an external key of the *source object*.
- Additionally, the rule specifies what has to be done to *retrofit* some information from the *created object* to the *source object*.

26.2. Configuration

26.2.1. Fields

Setting name	Description	Example
source_scope	The OQL to define the *source objects*. The only parameter available is *current_contact_id*	“SELECT UserRequest WHERE status = 'assigned'”
allowed_profiles	CSV list of profiles allowing the shortcut. The user must have at least one profile to have the shortcut available.	“Administrator, Support Agent”

	Wrong profiles names are ignored. Set as an empty string to allow the shortcut to anybody.	
menu_label	Optional: Label or dictionary entry for the new menu entry. It is optional and defaults to “Clone…”	
form_label	Optional: Label or dictionary entry for the form header. It is optional and defaults to “Cloning %1$s”	
report_label	Optional: Label or dictionary entry for the report once the object has been created. It is optional and defaults to “Cloned from %1$s”	
dest_class	Class of the object to create. If empty, it defaults to the class of the source object	“Change”
Preset	Array of *actions* to preset the object in the creation form. More information below.	array ()
Retrofit	Array of *actions* to retrofit some information from the created object to the source object. More information below.	array ()

menu_label, form_label and report_label can be localized without the burden of create dictionaries. To do so, create the settings menu_label/<language_code> (e.g., “menu_label/FR FR”) for each supported language. The setting menu_label will be the default value.

26.2.2. Actions

An action is specified as a string formatted as verb (arg1[, arg2[…]]).

The same action can be requested multiple times.

The following *verbs* are available:

Verb	Parameters	Description
clone_scalars	<none>	Copy all the scalar attributes
Clone	attcode1,attcode2, …	Copy the given attributes
Reset	Attcode	Reset the attribute to its default value
Nullify	Attcode	Reset the attribute to its null value (will appear to be *undefined* from the end user’s perspective)
Copy	att_to_read, att_to_write	Copy from att_to_read to att_to_write
copy_head	att_to_read, att_to_write	Copy last entry in the att_to_read case log attribute to att_to_write. An exception (non-blocking) is thrown if att_to_read is not a case log
Append	attcode, string	Append the string to the attribute. The string can contain

		placeholder like $this->attcode$ (or $current_contact_id$, $current_contact_friendlyname$, $current_date$, $current_time$). Commas must be escaped with a backslash. Newlines (\n) are allowed. The character set must be utf-8.
Set	attcode, value	Set a value. If the value is a string, it can then contain placeholder like $this->attcode$ (or $current_contact_id$, $current_contact_friendlyname$, $current_date$, $current_time$). Commas must be escaped with a backslash. Newlines (\n) are allowed. The character set must be utf-8.
add_to_list	attRead, attWrite, attLink, value	attRead is an external key on the read object, attWrite is a N-N link set (AttributeLinkedSetIndirect) on the written object, attLink is an attribute on the link class that will be set to <value>
apply_stimulus	stimulus code	Applies the given stimulus (saves the object). To be used in **retrofit ONLY.** Best practice: It is strongly recommended to set transition mandatory fields as well, otherwise they will stay empty and could break reporting.
call_method	function name	Calls the provided method on the written object. Its prototype must be "public function xxxx($oSource)". The function can send exceptions in case of failure. In such a case, the error message gets displayed in the log/error.log file
clone_attachments	<none>	Copy all the attachments from *source* to *destination*

You can use a string instead of a number, for identifying an object copier rule.
Internally the menu UID will be build this way: **object_copier_** + *the rule id you gave it*.

So UID of the above menus, we will be: object_copier_0 and object_copier_1
You can also display the hyperlink as a button (next to the "Other actions…" menu) by specifying in the configuration parameter 'shortcut actions' its UID

Example with call_method
Here we want to copy a n: n relation, which would be of a similar nature but from 2 different Classes of links, for example you want to create a Change Ticket from a FunctionalCI and automatically retrieve Contacts associated to CI and link them to the Change.

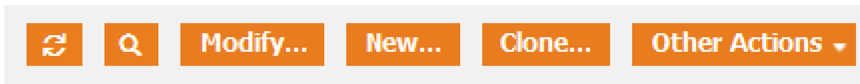
26.2.3. Display menu as button

If you want to display a menu as a separate button, outside of the Other actions menu, use the shortcut_actions parameter in the Configuration file.

Example: to display the Clone... menu (*rule id=1*) in the toolbar after the New... and Modify. buttons, set this:

'shortcut_actions' => 'UI: Menu: Modify, UI: Menu: New, object_copier_1',

The result will look like this:



If the specified menu is not available for a given class, it's silently not displayed

27. Workflow Graphical View

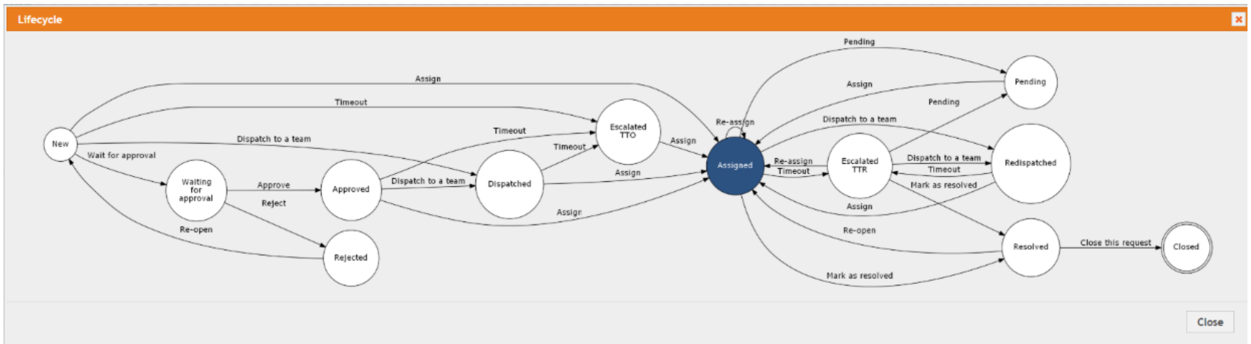
With this extension, get a quick overview of the object's workflow.

This allows to know what the current state is and what the next steps are.

Indeed, the extension adds a small map icon near the state attribute. By clicking on it, you will be able to see the states and transitions of the object's current workflow. The current state is highlighted in blue.

27.1. Features

Easily have a look at the object workflow by hovering its state.



27.2. Configuration

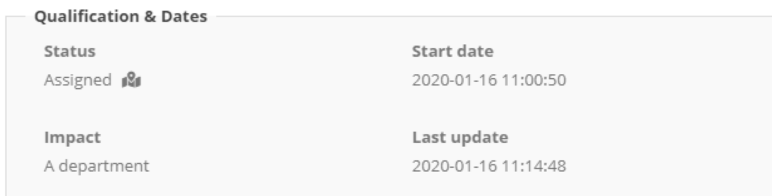
The extension has 5 parameters that can be set in the module settings part of the configuration file:

- **hide_internal_stimuli**: Define if the internal stimuli (not accessible by the user in the application) should be hidden in the graph. Values can true false, default is false.
- **stimuli_to_hide**: An array of classes / stimuli to hide from the graph. For each classes, the value must be a string of stimuli separated by a comma.
- **disabled_classes**: An array of classes for which the extension will be disabled. Default value is an empty array.
- **disabled_portals**: An array of "portals" (see the /tbITSM_design/portals tag of the XML Data Model) for which the extension will be disabled. For the admin. console use back office, for an end-user's portal instance use its ID (e.g., tbITSM-portal for the default one). Default value is an empty array.
- **show_button_css_classes**: A list of CSS classes for the button that show the workflow. Can be either classes from FontAwesome or your own.

27.3. Usage

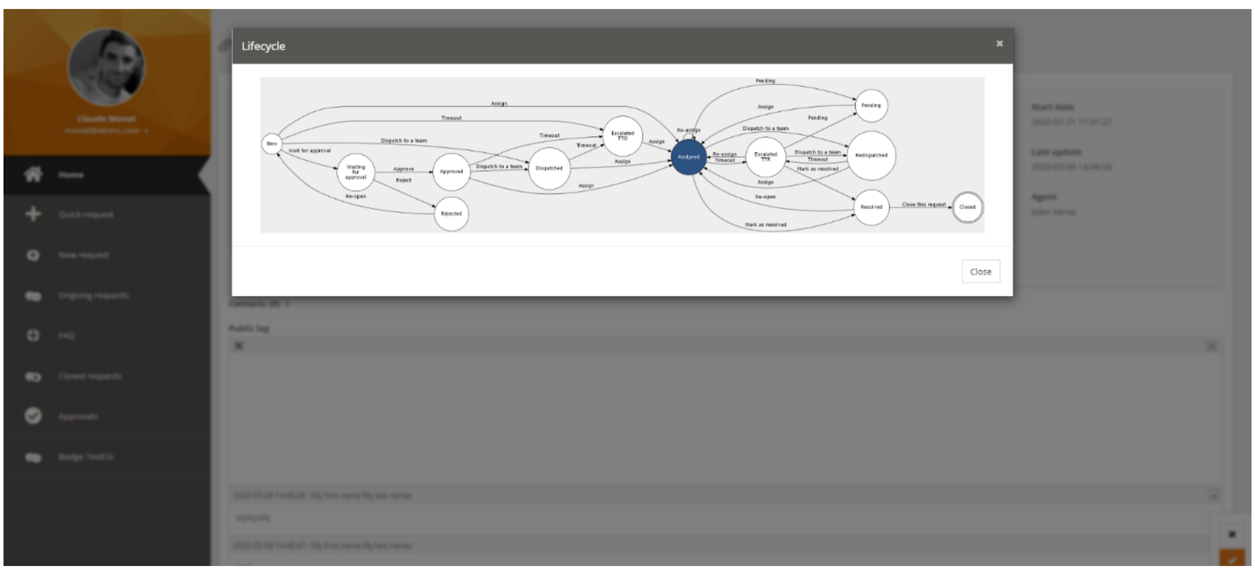
27.3.1. End-users' portal

In an object detail, if it has a workflow, a small map icon appears near the state
attribute.



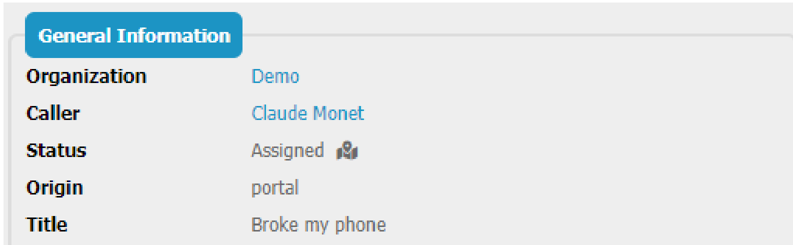
Click on it the show the workflow with its states and transitions. Also, the current state will be highlighted in blue to easily understand the next steps.

Note that dashed lines are for internal stimuli.



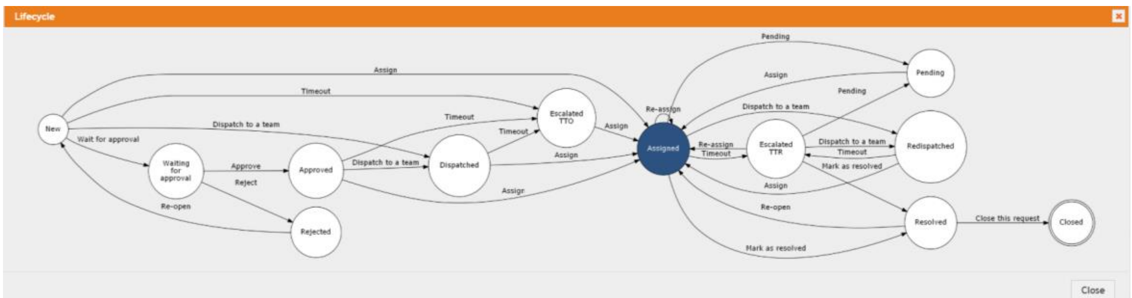
27.3.2. Backoffice

Same as in the portal, in an object detail, if it has a workflow, a small map icon appears near the state attribute.



Click on it the show the workflow with its states and transitions. Also, the current state will be highlighted in blue to easily understand the next steps.

Note that dashed lines are for internal stimuli.



28.1. Intended Audience

This document is targeted towards system administrators in charge of configuring and operating the tbITSM application. For the basic usage of the application, from the end-user point of view, refer to the tbITSM User Manual.

28.2. Prerequisites

This document assumes that you have at least a basic knowledge about MySQL or the **SQL language** in general.

28.3. New tbITSM

A few pages which can be useful:

- **Managing Organizations:** During Implementation, you have defined the tree structure of organizations for your tbITSM. Later you may need to add or modify some organizations.

- **tbITSM logs:** where are the log files, what to be found in it, how to configure logging

- **Scheduling background tasks:** This should have been set during tbITSM installation

- **Tuning tbITSM Performance:** if you encounter such issue, that could help.

- **Object Query Language:** for retrieving objects matching some criterion (*SQL like language*)

- **HTML formatting limitations:** what HTML syntax is supported within tbITSM Notes documents and Template HTML fields, Ticket's Description and AttributeCaseLog fields

- **Managing Personal Data:** some tips to help you complying to GDPR.

28.4. Administrator Menus

This part explains all the actions available on the Administrator menus:

Those menus which used to be limited to users with Administrator profile only, can now be delegated to other users.

28.5. Administration

- **User Accounts & Profiles:** What are the possible profiles? What do they mean? How to create a user account? … User forgets its password, Password policy.

- **Auditing the CMDB:** Define audit rules, using OQL, to control the completeness of the data stored in tbITSM.

- **Run Object Queries:** Get a set of objects compliant with some criteria.

- **Query Phrasebook:** Prepare your OQL queries and saved them for re-use.

- **Export:** Command line interface which can be used within a script. There is an interactive version and a CLI legacy.

- **Universal Search:** Allow to search for any tbITSM objects even if no menu exist for them.

- **Browsing the Data Model:** Not sure about the internal name of an object field, while writing an OQL, check here.

28.6. Configuration

- **Configuration Editor:** allow to change any configuration parameter on the fly. See the details of all configuration parameters and their usage.

- **Examples of possible actions based on parameters:** Setting tbITSM in read-only, Preventing concurrent modifications or adding attachment capability to objects which don't have it yet.

- **Synchronize Data Sources:** Allow to align tbITSM objects to an external data source. If you are not familiar with this, check the Overview first.

- **Notifications:** Based on trigger, emails are sent automatically. Define those triggers and the email templates with placeholders.

28.7. System

- **Database integrity:** Tools to identify database integrity issues and tips on possible fixes.

- **Data Backup:** Backup your database in a zip file, on demand or on schedule (requires Background tasks to be set).

- **Application upgrade:** allow to upgrade your tbITSM application to another version.

29. Managing User Accounts

tbITSM 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 tbITSM (viewing, creating/modifying or deleting which objects).

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

29.1. Viewing Profiles

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

Profiles

Total: 14 objects.

Actions

Pages: 1 2 10 objects per page

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
Communication Manager	Person responsible for managing communications to the customers
Configuration Manager	Person in charge of the documentation of the managed CIs
Document author	Any person who could contribute to documentation
Portal power user	Users having this profile will have the rights to see all the tickets for a customer in the portal. Must be used in conjunction with other profiles (e.g. Portal User).
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

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

Class	Read	Bulk Read (Export)	Modify	Bulk Modify	Delete	Bulk Delete	Stimuli
Closing rule	Yes	Yes	No	No	No	No	
Mail Inbox	Yes	Yes	No	No	No	No	
Attachment	Yes	Yes	No	No	No	No	
Organization	Yes	Yes	Yes	Yes	Yes	No	
Location	Yes	Yes	Yes	Yes	Yes	No	
Contact	Yes	Yes	Yes	Yes	Yes	No	
Person	Yes	Yes	Yes	Yes	Yes	No	
Team	Yes	Yes	Yes	Yes	Yes	No	
Document	Yes	Yes	Yes	Yes	Yes	No	
Document File	Yes	Yes	Yes	Yes	Yes	No	
Document Note	Yes	Yes	Yes	Yes	Yes	No	
Document Web	Yes	Yes	Yes	Yes	Yes	No	
Functional CI	Yes	Yes	Yes	Yes	Yes	No	

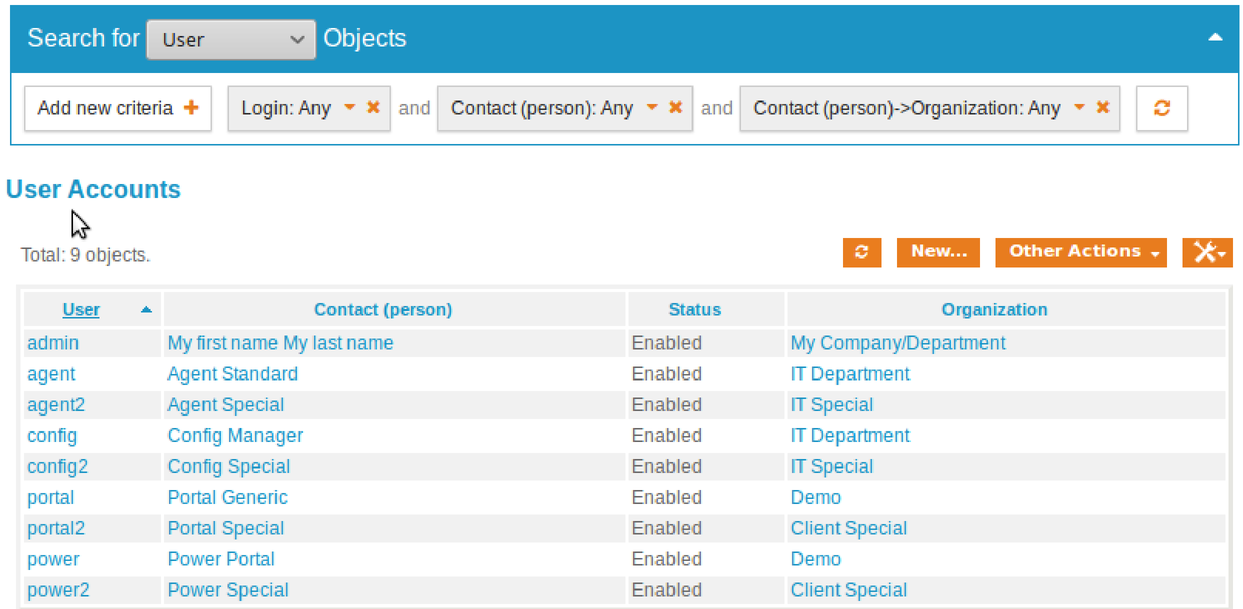
The tab "Users", lists all users having this profile.
The tab "Grant matrix" displays, for each class of objects, all the actions allowed for this profile.

29.2. 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.
Portal power user	Users having this profile will have the rights to see all the tickets for a customer in the portal. Must be used in conjunction with other profiles (e.g., Portal User).
Problem Manager	Person analyzing and solving the current problems.
REST Services User	User account with access to the REST Web Services. If the configuration setting secure_rest_services is set to true (which is the default), then only the user accounts having this profile are allowed to use the REST web services.
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.

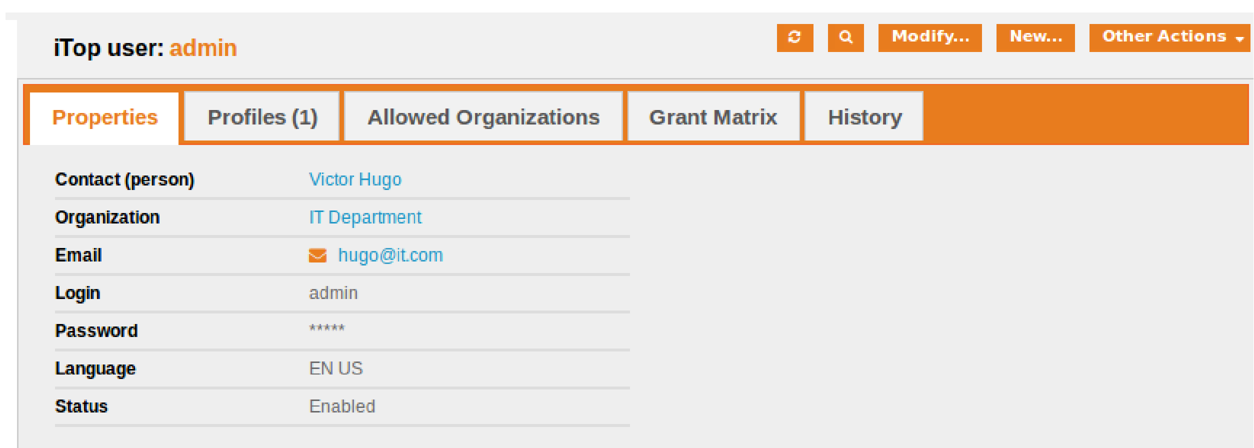
29.3. Viewing User Accounts

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



User	Contact (person)	Status	Organization
admin	My first name My last name	Enabled	My Company/Department
agent	Agent Standard	Enabled	IT Department
agent2	Agent Special	Enabled	IT Special
config	Config Manager	Enabled	IT Department
config2	Config Special	Enabled	IT Special
portal	Portal Generic	Enabled	Demo
portal2	Portal Special	Enabled	Client Special
power	Power Portal	Enabled	Demo
power2	Power Special	Enabled	Client Special

When clicking on a user you get the following details:



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.

29.4. 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:

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

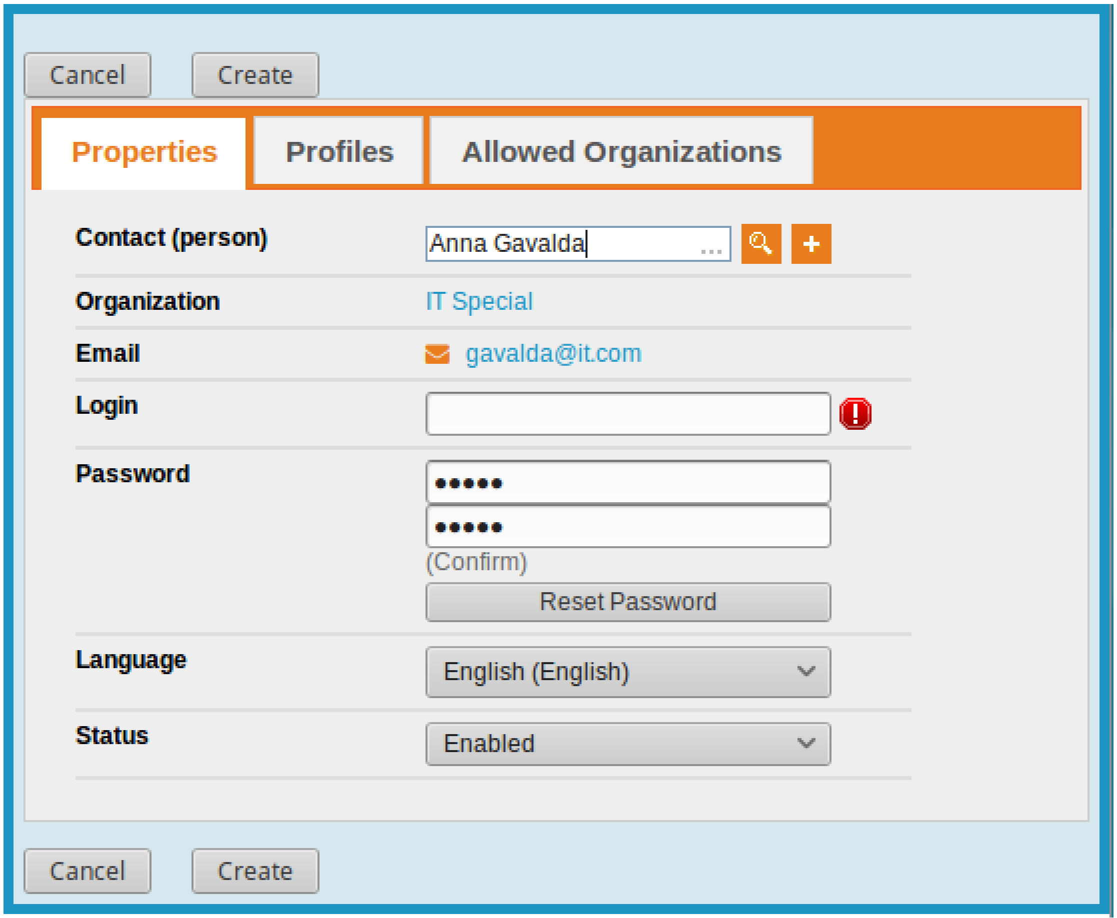
- tbITSM user accounts are internal to tbITSM. Their passwords are stored (encrypted) within the database of tbITSM. This type of account is useful for administrative users, for scripts and integration with other applications.

- 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 tbITSM

- If you decide to create an tbITSM 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.



If you have password policies, the password will need to follow them

A user record defines:

- The favorite language of this user, that will be used for displaying the tbITSM 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 tbITSM user account must have at least one profile.

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

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

29.5. 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.

An object is considered as belonging to an organization, if it has a field named exactly **org_id** which is an AttributeExternalKey or an AttributeExternalField on an AttributeExternalKey on class Organization.

Object without any org_id field are always visible to all users.

Object with an org_id field which would be empty (=0) are **never** visible to users with allowed organizations.

An Attachment object has an org_id field, fed with the organization of the object it is linked to. If that object has no org_id field, then it is empty, then it is not visible to users with allowed organizations.

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 tbITSM portal!

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

29.6. 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 tbITSM database, and therefore cannot be reconstructed from the content of the database.

29.7. I forgot my password

Users having an tbITSM user type of account can reset their password on their own: there will be no need for the administrator to do anything.

29.8. Deactivating an account

Starting with **tbITSM**, a new field "Status" has been added on the User Accounts. The "Status" has two possible values: "Enabled" or "Disabled". When set to "Disabled" the account is deactivated and the user can no longer connect to tbITSM. By default, the value for the field is Enabled.

30. Managing Personal Data

30.1. Inventory of personal data in tbITSM

30.1.1. Personal data in the Person object

Most of the personal information in tbITSM is stored in the Person object. In the standard tbITSM data model, these data consist of:

Field code	Label	Description of the field
Email	Email	Email address of the person
employee_number	Employee number	Employee number or any identifier used within the company
first_name	First Name	First name of the person
Function	Function	Function / job title of the person
location_id	Location	Location of the person (link to a Location object)
manager_id	Manager	Manager of the person (link to a Person object)
mobile_phone	Mobile Phone	Mobile phone number
Phone	Phone	Fixed phone number
Picture	Picture	Picture of the person

Due to the relational nature of tbITSM, the information stored in the Person object is not supposed to be duplicated elsewhere. The few exceptions to this principle are listed below:

30.1.2. Personal data in the History

All modification to the Person object is recorded in the history of this Person object, thus storing some personal information. You can list all the changes made to a given Person object using the following OQL query:

SELECT CMDBChangeOp WHERE objclass='Person' AND objkey=: person_id

Where: person_*id* is the identifier of the Person object. All the modifications performed by the user account associated with the contact also keep track of the name (i.e., friendly name) of the contact who made the changes.

SELECT CMDBChange WHERE userinfo=: person_name
Where: person_*name* is the complete name (i.e., friendlyname) of the Person, for example "Claude Monet".

30.1.3. Personal data in the case logs

Each Case Log field in tbITSM stores information about the person who created each entry in the case log (in the current version of tbITSM it is not possible to modify a case log entry). The information recorded in the case log (for each entry) contains:

- The date and time of the modification
- The full name (at the time of the modification) of the Person who made the modification
- The identifier of the Person object who made the modification

30.1.4. Personal data in Email notifications

Whenever a notification is sent by tbITSM as an email, the information is recorded in the EventNotificationEmail object. This object records the email sent to the mail server. This object contains the email addresses of the person it was addressed (in TO, CC or BCC) and the content of the message sent. The EmailNotification object also stores the date at which it was emitted.

30.2. Data flows

One of the important topics when dealing with personal data is to document the flow of personal data within your organization. This document (and tbITSM itself) cannot document this flow for you since each use case is specific and depends on the processes and integrations put in place when operating tbITSM. However, tbITSM provides several features which can help you document and control this flow of information.

log of REST/JSON web services

You can audit the operations performed via the REST/JSON services by setting the configuration parameter 'log_rest_service' to true in the tbITSM configuration file. Once this parameter is set, every REST/JSON operation will create an EventRestService object in the tbITSM database. This object records:

- The date/time of the operation
- The user account used to perform the operation
- The JSON input parameters
- The output of the operation (code, message and – a trimmed version of the - JSON output)

Using all the information you can determine the flow of personal information via REST/JSON webservices (if any).

Enabling 'log_rest_service' can have a significant impact on the performance of Web services and increase the tbITSM database space use. Therefore, it is recommended to turn on this feature for auditing purposes only during a limited period of time.

Starting with tbitsm, only the user accounts with the profile “REST Services User” can execute REST/JSON operations.

30.3. List of Synchronization Data Sources

You can easily list all Synchro Data Sources which import of update Persons by search searching for Synchronization Data Sources which “Target Class” is “Person”. This search can be performed through the “Admin tools / Synchronization Data Sources” or by running the following OQL query:
SELECT SynchroDataSource WHERE scope_class='Person'

Each data source keeps track of a complete log of its operations, including the date/time it was run and the summary of the operations executed for each run (creations/updates/deletions).

Note that if a “User” is specified for a Synchronization Data Source, then only the specified user account can execute this synchronization.

30.4. Restriction of Bulk Exports (and Bulk Imports)

Only the user accounts having the capability “Bulk Read” on Persons can perform exports of a list of persons.

Similarly, only the user accounts with the capability “Bulk Modify” on Persons can perform a CSV import of Persons.

These capabilities can be viewed using the “Grant matrix” tab on the details of each user account.

30.5. Cleanup of personal data

To completely cleanup the data related to a person, one has to:

- Delete the Person object corresponding to this person, this will also remove the history information about this Person.
- Delete the case log entries created by this person

- Delete the EmailNotification objects corresponding to emails sent to this person

But there are limitations to those actions:

Depending on the relations between the targeted Person object and other objects in the tbITSM Database, the deletion of the Person object may not be possible without specifying a replacement Person on some other objects in order to respect the database integrity constraints.

Deleting the case log entries is currently not implemented in tbITSM, and is non-trivial to achieve with SQL queries since the case log text must be kept consistent with another field called the case log index.

Updating the CMDBChange objects to replace the references to the contact's name can be done in one single SQL query, since there is only one field 'userinfo' to anonymize in the table. Since this field is only displayed to the end-user and not user for any reference in tbITSM, there is no integrity constraint and any value can be put in the field.

Since one email notifications may be addressed to several persons (in TO, CC and BCC), deleting one such notification will affect other users than just the targeted person.

Due to the limitations mentioned above the "pseudonymization" approach seem more realistic to implement than a complete wipe of the personal data.

30.6. Pseudonymization of personal data

Anonymize the Person object by replacing the values in all its mandatory fields by anonymous values (constant or random)

Empty all non-mandatory fields (like Location, Manager, Phone number, n: n relations)

Pseudonymization of the case log entries, by removing all references to the person, without altering the length of the text, to preserve the consistency with the case log "index" field.

Deletion of the Person's history (CMDB Change / CMDB ChangeOp) to delete the "previous values" of the fields, which would help recover the values.

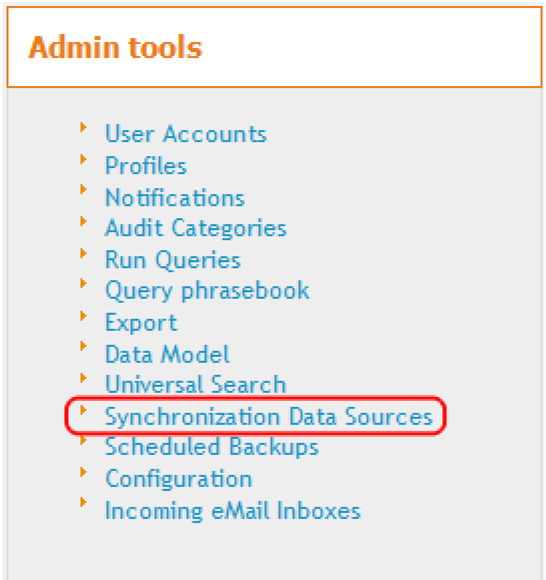
Anonymization of the CMDB Change "userinfo" fields containing the name of the Person.

Deletion of all EmailNotification objects past a (short) retention period

31. Data Synchronization References

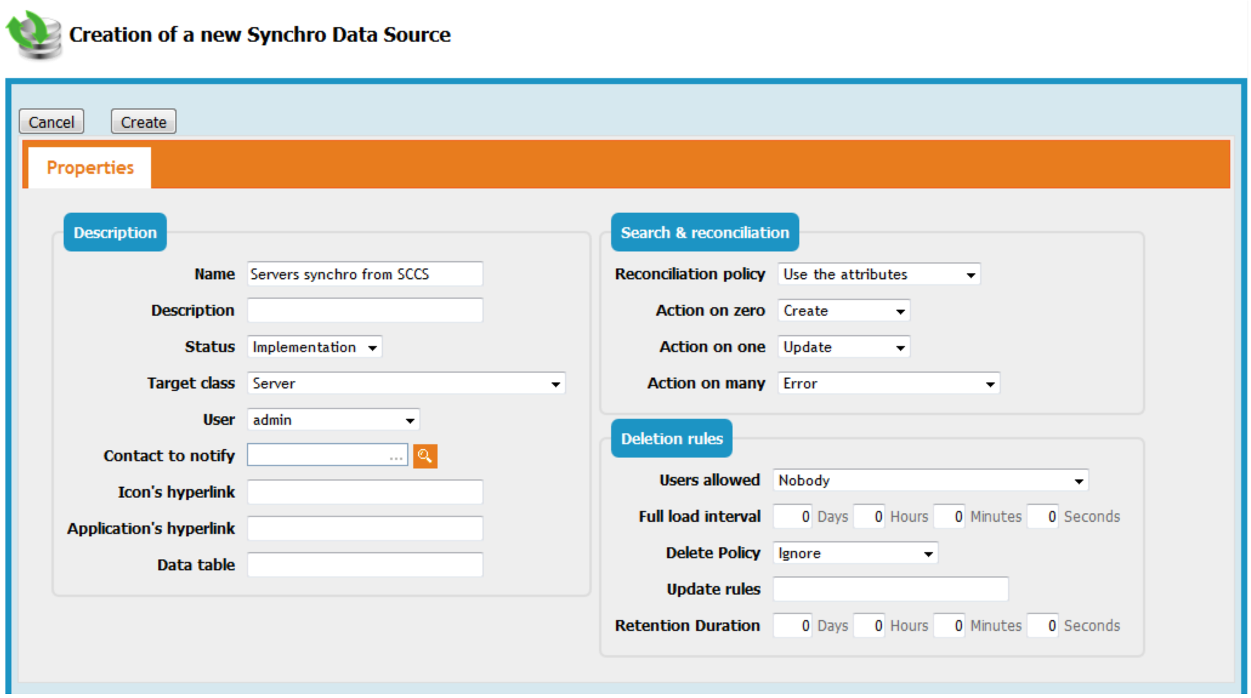
31.1. Data Source Definition

The **Data Sources** are defined using the menu **Admin Tools/Synchronization Data Sources**.



Each data source defines:

- The **target** class of the objects to be synchronized
- The **list of attributes/fields** of the objects to synchronize
- How to **search/reconcile the objects** with the objects already existing in tbITSM
- **The rules of synchronizing/updating** and possibly deleting objects in tbITSM
- How the **synchronized objects behave** for the tbITSM end-users (which fields are read-only, are users allowed to delete the objects…)
- An **optional hyperlink and icon** to refer tbITSM users to the corresponding object in the external application



31.1.1. Description

This section of the *Properties* defines some general parameters for this **Synchro Data Source**

Property	Mandatory?	Meaning	Sample value
Name	Mandatory	The name which refers to this data source	Servers' synchro from SCCS
Description	Optional	An additional description, for information only	
Status	Mandatory	implementation, production or obsolete: for filtering/classifying the data sources	production
Target Class	Mandatory	The class of objects to synchronize	Server
User	Mandatory	The user allowed to execute this synchro	admin
Contact to notify	Optional	The contact to notify (by email) of the results when the synchronization is executed and **contains** at least one **error**	
Icon's hyperlink	Optional	The hyperlink (URL) to an image representing this data source. This icon is shown instead of the default icon in the tooltip appearing on the "Lock" symbol next to the title of the synchronized object in tbITSM	
Application's hyperlink	Optional	The hyperlink (URL) to the remote application from which the synchronized data comes from. The hyperlink can contain placeholders such as $this-	https://myapp. acme.com/displ ay/server/$repli

		>attcode$ (which will be replaced by the value of the attribute of the current object) or $replica->column_name$ (which will be replaced by the value of the named column from the replica)	ca->primary_key$
Data table	Optional	The name of the MySQL table in which the replica information will be stored. If omitted, the system will generate a unique name for you.	synchro_data_servers_from_sccs

31.2. Search & reconciliation

This section of the properties defines

- How the information contained in the replica table is matched against the content of the tbITSM database
- What is the behavior of this synchro data source concerning the creation of new tbITSM objects?

Property	Meaning
Reconciliation policy	Use the attributes: when a new replica is created, the system will search for a matching object in tbITSM based on the list of *Synchro attributes* with the flag Reconciliation set to Yes. Use primary_key: the column primary_key of the replica is expected to contain the identifier of the object in tbITSM.
Action on zero	What to do when the reconciliation did not find any object in tbITSM. Create instructs the system to create a new object of the Target Class. Error will cause the system to report this case as an error (i.e., this synchro data source is supposed to only update existing objects).
Action on one	What to do when the reconciliation found exactly one matching object in tbITSM. Update instructs the system to update this object. Error will cause the system to report this case as an error (i.e., this synchro data source is supposed to create objects and never update existing objects).
Action on many	What to do when the reconciliation found several matching objects in tbITSM. Create instructs the system to create another object. Error will cause the system to report this case as an error. Take the first one instructs the system to pick the first object in the list and to update it.

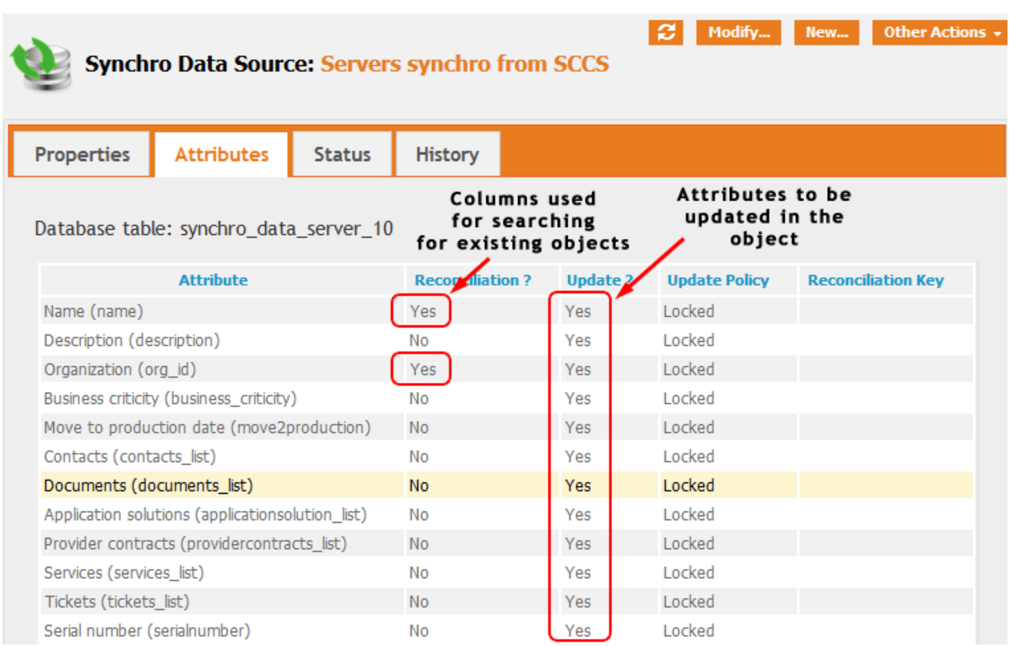
31.3. Deletion rules

This section defines how the deletion of the synchronized objects are handled:

- by the end-users in tbITSM,
- by the synchronization system itself.

Property	Meaning
Users Allowed	Which end-users are allowed to delete synchronized objects in tbITSM. Nobody when an object is synchronized, nobody is allowed to delete it in tbITSM? only the synchronization system can delete it. Administrators only only administrators are allowed to delete a synchronized object. Everybody allowed to delete such objects the normal access rules apply to such synchronized objects.
Full load interval	The duration between two executions of the synchronization which are considered as having loaded (touched) all replicas. When a replica has not been *touched* for a duration greater than this value, it is considered as obsolete.
Delete Policy	What to do when a replica becomes obsolete. Ignore do nothing, the associated object remains as is in tbITSM. Delete: Delete the associated object in tbITSM (and the replica in the data table). Update: Update the associated object as specified by the Update rules (see below). Update then Delete Update the associated object as specified by the Update rules (see below), then after the Retention Duration, delete the object (and the replica).
Update rules	A list of values separated by semi-colons, of the form attribute_code:value to specify which attribute of the associated object to set and to which value. Example: status: obsolete; description: no longer synchronized.
Retention duration	Only used when the Delete Policy is set to Update then delete. In that case, an object deleted from the Source is updated immediately and deleted from tbITSM only after that Retention duration.

31.4. Attributes



The Attributes tab

The “Attributes” tab defines:

- How to determine whether a new object must be created in tbITSM or an existing object must be associated with the replica (if the reconciliation rule is “Use Attributes” in the properties)
- Which attribute(s) of the associated object must be updated, and how?

When a new record appears in the synchro_data_xxx table, the synchronization mechanism will try to associate this record with an existing tbITSM object. If one or more Attribute is marked as “Yes” in the “Reconciliation?” column, these columns/attributes will be used to search for an tbITSM object that matches all these values. If one object is found, it will become associated with this record. If no object is found, a new tbITSM object will be created.

The Update column

The **Update column** defines if the attribute must be updated or not.

- Checked = Yes: the attribute will be set/updated
- Unchecked = No: the external source information for this attribute will be ignored, the existing value on the tbITSM object will be kept as is.

The Update Policy column

The column **Update Policy** defines the behavior of the updated attribute. The possible values are:

- **Locked:** the attribute will be completely driven by the data synchronization and will appear as read-only (with a small lock icon next to it) in the tbITSM user interface,

- **Unlocked:** the attribute will be updated by the synchronization (whenever the source data changes) but end-users can still modify the attribute in parallel inside tbITSM,

- **Initialize if empty:** the synchronization will only set a value of this attribute if the value is empty in tbITSM. The attribute remains modifiable inside tbITSM.

The Reconciliation key column

Finally, the **Reconciliation key column** defines how to process attributes which are references to another tbITSM object (i.e., Foreign keys). The supplied data can contain either the actual value of the foreign key (id) or another attribute such as the name of the object; which must uniquely identify the object.

For foreign keys, tbITSM does not support the reconciliation on a combination of the multiple attributes (like name AND organization). If such a need arises, then you must use an ETL or a script to perform the desired reconciliation before providing the id to tbITSM.

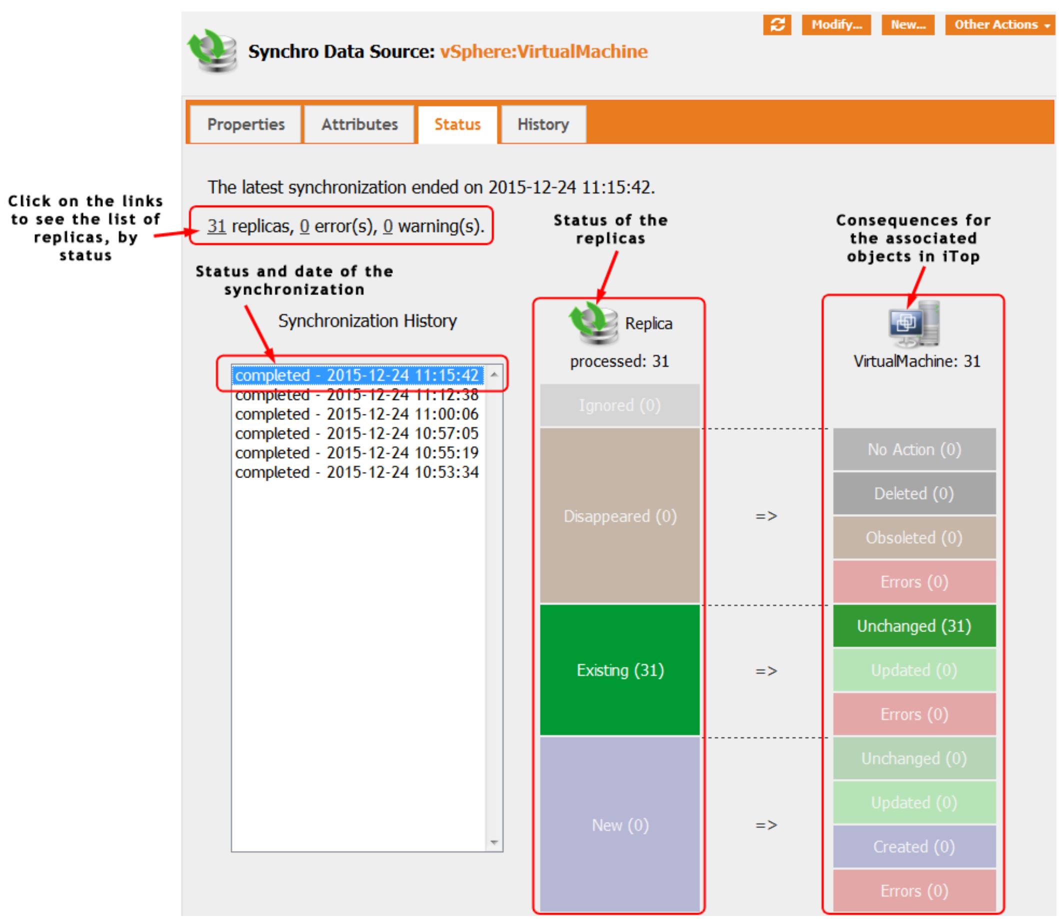
31.5. Status

The status tabs

The **Status tab** displays information about the execution of the synchronization. If the synchronization has never been run, the tab displays the following message:

This synchro was never run. No log yet.

Otherwise, the tab displays a schema similar to the one below:



31.6. Synchronization history

On the left of the screen, each execution of the synchro is displayed in an historical list, showing the overall status, the start date and time of the execution.
Click on an element of this list to show the corresponding status in detail, displayed in the two columns on the right.
The column at the left displays the status for the records in the synchro_data_xxx table:

- New records are records which were added in the table since the previous execution of the synchro,
- Existing records are records which were existing before and have been loaded again since the previous execution of the synchro,
- Disappeared records are records which were not loaded since more than the full_load_interval duration.
- Ignored records corresponds to objects no longer existing in tbITSM.

The column at the right displays the consequences of the synchro, in terms of modifications on the associated objects in tbITSM.

For each New record, the consequences can be either:

- The creation of the new object in tbITSM (Created),
- The update of an already existing object in tbITSM (Updated),
- The simple association with an object in tbITSM, which is already up-to-date (Unchanged),
- Or an Error when trying to either create or update an object in tbITSM (most probably because of inconsistent data).

For each Existing record, the consequences can be either:

- The update of the associated object (Updated),
- No change at all (Unchanged)
- Or an error when trying to update the associated object (Errors)

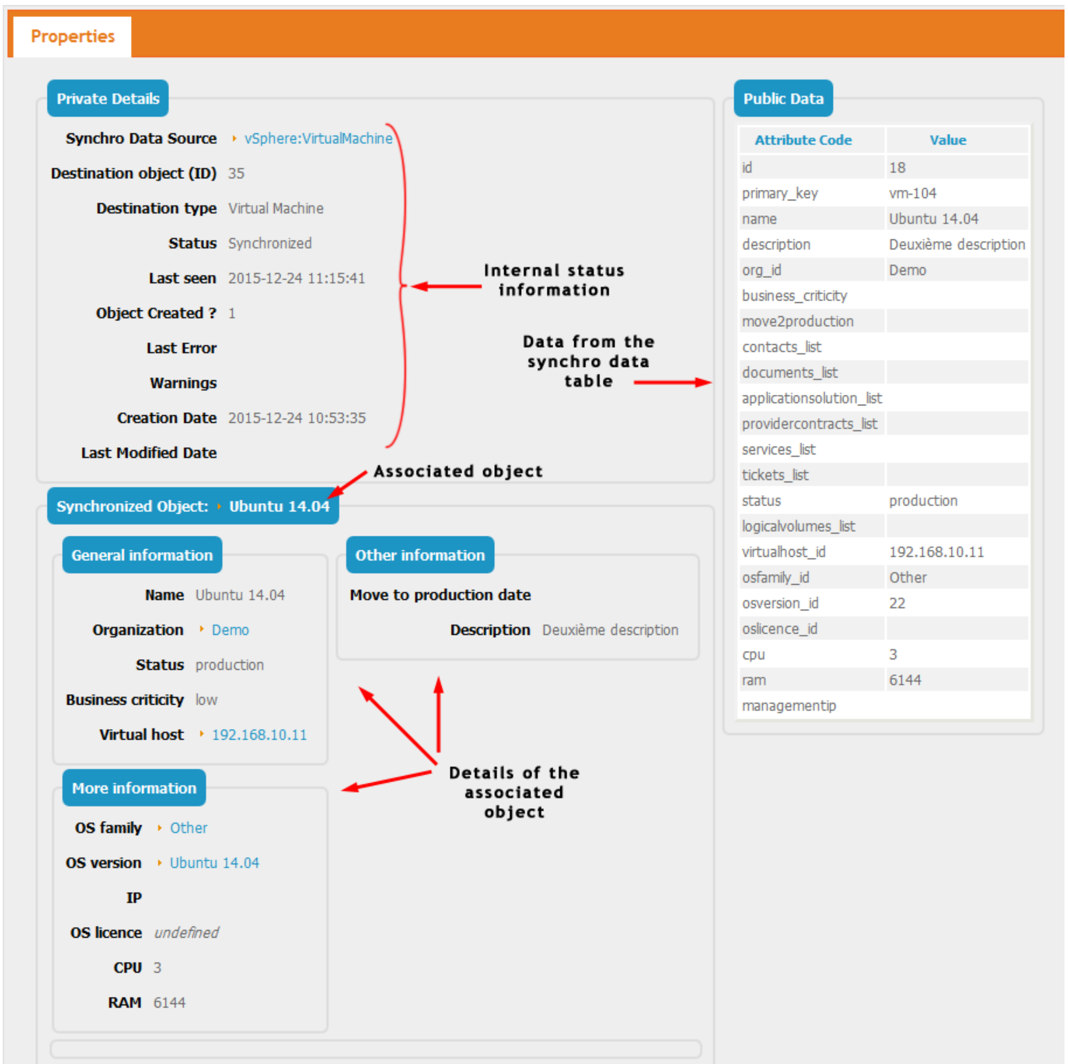
For each Disappeared record, the consequences (depending on the "Deletion rules") can be either:

- The associated object in tbITSM has been deleted (Deleted),
- The associated object in tbITSM has been updated (Updated),
- The associated object in tbITSM was left unchanged (No Action),
- Or an error occurred when trying to update or delete the associated object in tbITSM (Errors).

31.7. Replicas

In tbITSM, the *Replicas* are used to store the status information and the relation between the raw data loaded in the synchro_data_xxx table and the associated tbITSM object.

The Replicas for a given data synchronization source can be displayed by clicking on the links at the top of the status tab. A Replica is displayed as shown below:



The content of the replicas is useful to understand the behavior of the data synchronization mechanism (and to get more information about the errors), but its data are maintained by the synchronization and should never be modified manually.

31.8. Using MySQL statements to populate the synchro_data_xxx table

If you populate the table synchro_data_xxx using SQL commands (either with an ETL or a homemade script), you **must** take care of never creating duplicate records. Remember that each record in the synchro_data_xxx table corresponds to exactly one object in tbITSM.

In order to avoid such duplicates, you **must** first query for an existing record before inserting a new record into the table. To speedup this query you can either construct a value that uniquely identifies the “source” object and store this value in the “primary_key” column of the table (which has a unique index), or use any combination of the record's column and alter the definition of the synchro_data_xxx table to add your own indexes.

32. Notification

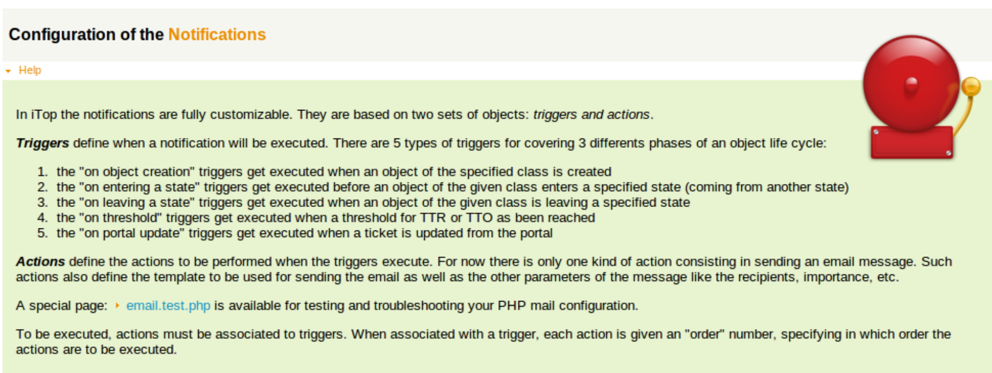
tbITSM integrates a notification system linked to the life cycle of the objects. This allows administrators to define e-mail notification rules when an object of a given class enters or leaves a specified state, when a new object is created, when an update occurs from the portal or when certain thresholds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

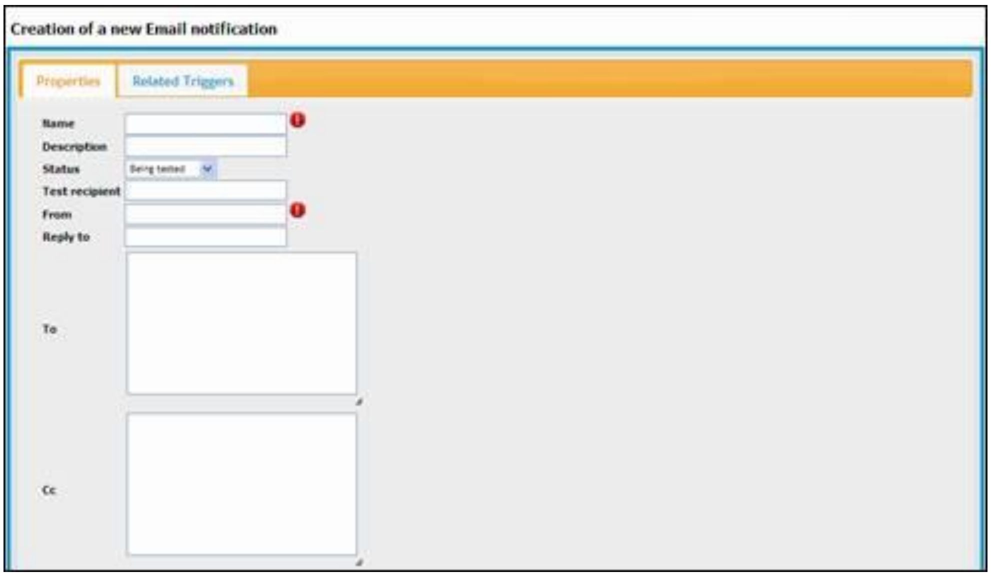


- The “Triggers” tab displays all created triggers.
- The “Actions” tab displays all Actions

32.1. 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:



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 tbITSM 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 i sthe “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”.

32.2. 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:

Placholder	Meaning
$this->name()$	The name of the current object
$this->hyperlink()$	The URL to access the current object in tbITSM
$this->hyperlink(portal)$	The URL to access the current object in the tbITSM portal
$this->html(attribute)$	The HTML representation of the value of the attribute

Comments specific to some types of attributes:

- Case Log attributes: the syntax $this->head(attribute)$ returns the text of the latest entry in the case log.
- $this->head_html(attribute)$ returns an HTML formatted representation of the latest entry, whereas $this>html(attribute)$ returns an HTML formatted representation of the whole case log (you can apply your own CSS styling to make it beautiful).
- Link set attributes: the very standard syntax $this->attribute$ returns a list of names (separated by a new line character, see the note below).

32.3. Testing notifications

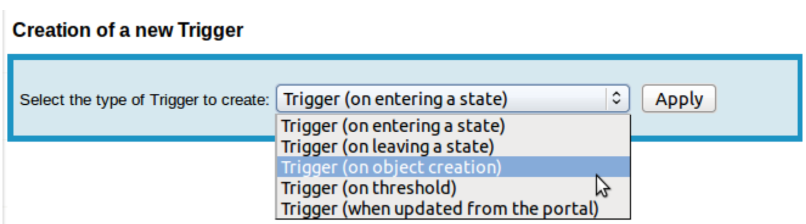
To test a new action, you can use the status “Being tested” and fill “Test recipient” with a test address. In that case, the notification will be sent to this latter address. Once the notification have been tested, change its status to “In Production” to have notifications flow to their actual recipients.

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

32.4. 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 tbITSM 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:

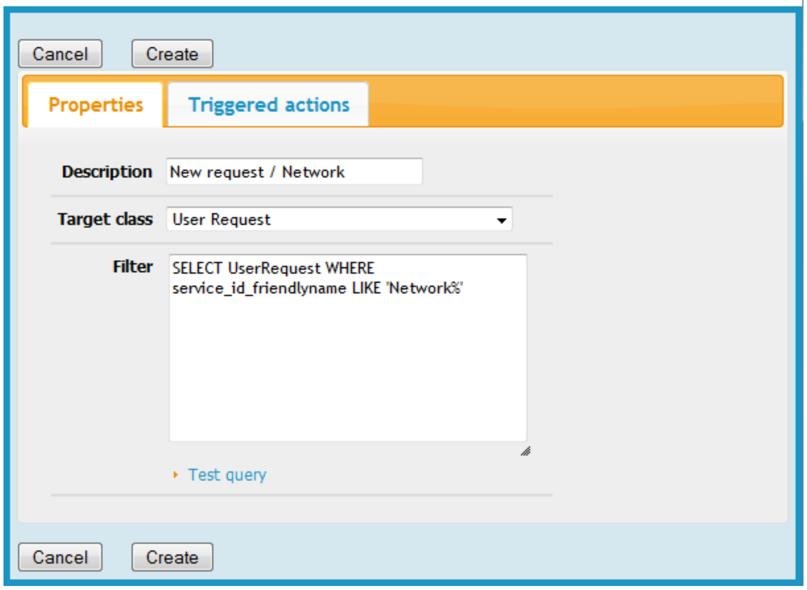


You have to select which type of trigger you want to create:

- 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:

Creation of a new Trigger (on object creation)



Any type of trigger requires you to specify three parameters:

- *Description* is left to you to further identify the purpose of this trigger.
- *Class* defines the class of object for which this trigger is applicable.
- *Filter* restrict the objects to which the trigger applies. It is an OQL query returning all the objects that would activate the trigger. Leaving it blank means: all the objects of the expected class.

Depending on the type of trigger, you will have to define additional parameters:

- *On entering/leaving a state* both require the state. The value to be entered for the "state" is 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.
- *On threshold* requires a stop watch and a threshold. The expected value for the stop watch is an attribute code. User Requests and Incident tickets come with two stop watches: **tto** and **ttr**. The threshold is a percentage of the goal of the stop watch. With the standard data model you can use 75 or 100.

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 tbITSM 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 SSMTP as a proxy to the actual mail server.

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

SMTP = <smtp server>
smtp_port = 25

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.

32.5. Email Configuration

tbITSM supports two methods for sending emails: the built-in mail function of PHP or SMTP. The configuration parameter email_transport determines which method is used for sending eMails from tbITSM. 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 the SMTP transport, the following parameters can be set in the tbITSM 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)	

Though it seems easier to use the default transport (PHP mail), the drawback is the lack of reporting when it fails. For instance, you may get the error message "*No valid recipient for this message.*" for a configuration issue.

32.6. Notifications and application responsiveness

Sending emails is a relatively slow operation. Depending on your mail server, sending one email may take several seconds (establishing the connexion to the server, sending the data, etc…). When a Ticket is created or updated in tbITSM, several emails may be emitted, depending on the notifications configured. This can take a few seconds to complete. To improve the responsiveness of the application, the notifications can be sent asynchronously by a process running in the background on the web server. To activate the asynchronous sending of notifications, set 'email_asynchronous' ⇒ true, in the configuration file and make sure that the background process is up and running.

33. Release Management

Release and Deployment Management is one of the main processes under the Service Transition section of the IT Infrastructure Library (ITIL) framework. This process is often referred to in short-form as simply “release management”. ITIL defines release and deployment management as the process of managing planning and scheduling the rollout of IT services, updates and releases to the production environment. Release in this context refers to the development of a newer version of a service or component and deployment means the process of integrating it into the live production environment.

Release management plays an important role of bridging the gap between project activities and the things that project teams produce and the ongoing operations and users that will consume these things. It is very common for organizations to have multiple projects underway at the same time and release management provides a structured approach for bringing changes together, testing to make sure they work correctly and then safely introducing them into the live environments that business operations rely on. Release management also ensures that any applicable knowledge and resources are transferred from the teams developing the new features or components to the operations team that will be responsible for supporting them.

33.1. Release Management Process

It is important that each project team wishing to introduce changes to the production environment are aligned with each other and are aware of each other’s changes and resource usages. They must follow the same process, policies and guidelines for planning, building, testing and deploying a release. ITIL breaks release management down into six sub-processes that enable release management to be performed effectively, efficiently and safely to facilitate the flow of changes into the operations environment.

- **Release management support**: provides guidelines and support for the deployment of releases including the roles that are involved in other parts of the release and deployment management process.

- **Release planning**: defines the scope and content of releases according to release management policies, assigns authorized changes into release packages and defines a schedule for building, testing and deploying the release.

- **Release build:** deals with the actual development of all required release components including the issuance of all necessary work orders and purchase orders for components sourced from vendors and ensuring that all release components are ready for validation and testing.

- **Release deployment**: manages the deployment of release components into the live production environment and the transition of documentation and training to end-users and operating staff.

- **Early-life (post-release) support**: the initial period after the deployment of a new release when the release and deployment management team work with the incident management team to resolve operational issues and remove errors and deficiencies caused by the release.

- **Release closure**: formally closing release activities, verifying all documents and records are properly updated and reporting release outcomes and feedback to project teams.

33.2. Role of a Release Manager

The release manager plays a combined coordination and governance/oversight role – tasked with ensuring that the release is completed effectively and safely. Release managers are typically IT professionals with specialized skills and experience using standards, processes and tools to coordinate release activities. In IT contexts, release managers work with business leaders, IT project teams and operations staff to ensure a well-orchestrated release of technical features into the IT environment. In product management contexts, release managers work with business development, marketing, R&D and other teams to coordinate across the company in support of a planned product release.

Large releases may involve multiple staff members working as a release team. If this is the case, the release manager provides an over-arching management and leadership function, coordinating both the release team and the release itself. The release manager is responsible for governance and quality oversight of the release, determining what level of risk and complexity the release represents and ensuring that the right level of due-diligence is applied to ensure release objectives are achieved without compromising the ongoing operations of the company.

33.3. Why Release Management is needed?

The primary objective of release management is to plan, schedule and control the deployment of IT services and updates into the live environments. Companies evolve and as they do, their needs change, so the IT environment needs to change too. Release management provides a means for making changes effectively and safely. It does this by ensuring only sufficiently tested services and components can be released into the live environment that the business uses.

Some other benefits of release management include:

- Faster delivery of changes and new features to users
- Reduced risk of un-authorized releases breaking features that people are using
- Predictable schedule of deployments at times that minimize business impact
- Ensuring new or changed services can meet agreed service requirement
- Providing proper knowledge transfer to users and support staff