

# iTop Extensions

This section contains a list of extension modules for iTop

- [Ticket creation from eMails](#): automatic creation of User Request Tickets by scanning a given POP3 or IMAP mailbox.
- [TeemIP](#) IP Management module for iTop
- [Android Application](#) to Connect to iTop.

## Ticket Creation from eMails

This page shows the **version 2.1** of the 'Ticket Creation from eMails' module. This version supports connecting to either a POP3 or an IMAP server. It uses the PHP imap extension for connecting to IMAP server and falls back to the PEAR POP3 implementation for connecting to POP3 servers.

This page describes an extension to iTop 1.2.1 or 2.0 to automate the creation of User Request tickets from incoming eMails.

The modules listed in the page, turn incoming emails from a POP3 or IMAP mailbox into User Request tickets.

- The subject of the email becomes the title of the ticket.
- The body of the email becomes the description of the ticket (a plain text version of the body is produced if the body is in HTML)
- A contact with the same email address as the sender of the email **MUST ALREADY** exist in iTop.
- The organization of the ticket is the organization of the caller/email of the sender of the email.
- Unless otherwise specified, that attachments and inline-images are uploaded as attachments in iTop.

The incoming emails are kept in the mailbox until the corresponding ticket in iTop is either closed or deleted. Then, the email is deleted from the mailbox.

## Requirements

- iTop 1.2.1 or 2.0 (it *should* work with iTop 1.2 as well, but it's not guaranteed)
- PHP 5.2.1+ with the [IMAP extension](#) enabled if you want to connect to an IMAP server, or PEAR::NetSocket (iTop comes with its own copy of PEAR::POP3) if you want to connect to a POP3 server.

- A connection to a POP3 or IMAP server with a valid mailbox

## Installation

### Installation for iTop 2.0

1. Download the package: [EmailSynchro-v2.zip](#) and expand the two folders it contains “combodo-email-synchro” and “itop-simple-email-synchro” into the “**extensions**” directory of iTop.
2. For Connecting to an IMAP server, install the [PHP IMAP](#) extension or for a POP3 server install Pear::NetSocket as follows:
  1. On Windows: locate the directory where php is installed. Open command prompt AS AN ADMINISTRATOR and from the this directory launch 'go-pear.bat'. Accept the default answer to all questions.
  2. Then from this command prompt, type “pear install Net\_Socket”
  3. On Linux Ubuntu, as root, launch the following command: apt-get install php-net-socket
  4. On Linux RedHat/Fedora/CentOS, as root, launch the following command: yum install php-pear-Net-Socket
3. If you have already installed iTop make sure that the configuration file config-itop in conf/production is NOT read-only.
4. Point your web browser to `http(s)://<your_itop_root>/setup` and follow the wizard. Make sure that you select the option to “Upgrade an existing iTop instance”:
- 5.

**iTop** Install or Upgrade choice

What do you want to do?

☐ Install a new iTop

☒ Upgrade an existing iTop instance

Location on the disk:

**Database Server Connection**

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

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

Password:

✓ Database server connection Ok.

**Database**

Database Name:

Use a prefix for the tables:

☒ Backup the iTop database before upgrading

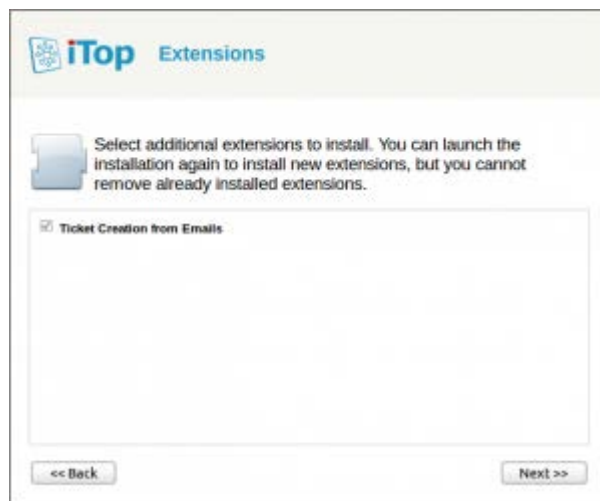
Save the backup to:

✓ mysqldump is present: mysqldump Ver 10.13 Distrib 5.1.63, for debian-linux-gnu (64bit) 23.08 Go free in /opt/dev/2.0/data

<< Back Next >>



Finally check the module “Ticket Creation from Emails” in the list of extensions at the end of the interactive wizard. Then complete the installation.

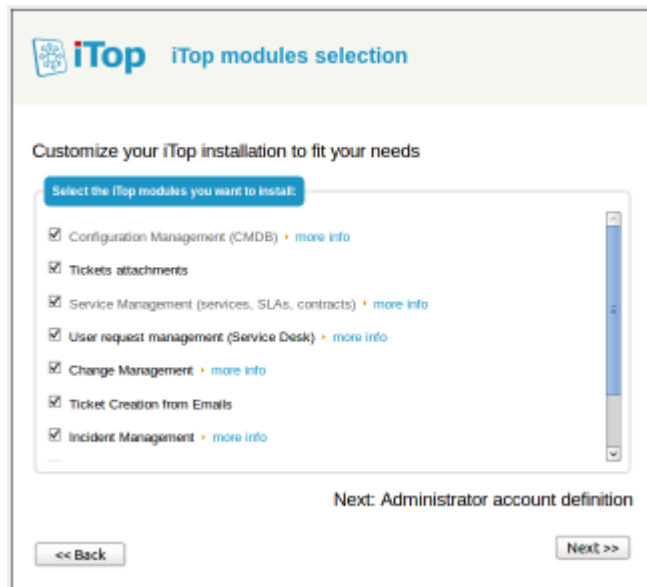


## Installation for iTop 1.2.1

This page describes an extension to iTop, so it assumes that you have already downloaded iTop (and maybe already installed it too)

1. Download the package: [EmailSynchro-v2.zip](#) and expand the two folders it contains “combodo-email-synchro” and “itop-simple-email-synchro” into the “**modules**” directory of iTop.
2. For connecting to an IMAP server, install the [PHP IMAP](#) extension or for a POP3 server install Pear::NetSocket as follows:
  1. On Windows: locate the directory where php is installed. Open command prompt AS AN ADMINISTRATOR and from the this directory launch 'go-pear.bat'. Accept the default answer to all questions.
  2. Then from this command prompt, type “pear install Net\_Socket”
  3. On Linux Ubuntu, as root, launch the following command: apt-get install php-net-socket
  4. On Linux RedHat/Fedora/CentOS, as root, launch the following command: yum install php-pear-Net-Socket

3. If you have already installed iTop, then make a backup of the iTop configuration file ("config-itop.php" at the root of the iTop installation) and make sure that the configuration file is NOT read-only.
4. Point your web browser to `http(s)://<your_itop_root>/setup` and follow the wizard. Make sure that you check the module "Ticket Creation from Emails". then complete the installation.



## Configuration

Once the new module has been installed, edit the configuration file `config-itop.php` and look for the two new sections:

```
'combodo-email-synchro' => array (
    'debug' => false,
    'notify_errors_to' => '',
    'notify_errors_from' => '',
    'introductory_patterns' => array (
        0 => '/^le .+ a écrit :$/i',
        1 => '/^on .+ wrote:$/i',
        2 => '|^[0-9]{4}/[0-9]{1,2}/[0-9]{1,2} .+:$',
    ),
),

'itop-simple-email-synchro' => array (
    'mail_server' => 'imap.demo.com',
    'protocol' => 'imap',
    'port' => 143,
```

```

    'login' => 'mailbox-name',
    'pwd' => 'your-password',
    'mailbox' => '',
    'pop3_auth_option' => 'USER',
    'imap_options' => array (
        0 => 'imap',
    ),
    'debug' => true,
    'exclude_attachment_types' => array (
        0 => 'application/exe',
    ),
    'default_values' => array (
    ),
    'parts_order' => 'text/plain,text/html',
),

```

You can re-arrange it to make it look a bit easier to understand:

```

    'combodo-email-synchro' => array (
        'debug' => false,
        'notify_errors_to' => '',
        'notify_errors_from' => '',
        'introductory_patterns' => array ('/^le .+ a écrit :$/i',
            '/^on .+ wrote:$/i', '^[0-9]{4}/[0-9]{1,2}/[0-9]{1,2} .+:$/i'),
    ),

    'itop-simple-email-synchro' => array (
        'protocol' => 'imap',
        'mail_server' => 'imap.demo.com',
        'port' => 143,
        'login' => 'mailbox-name',
        'pwd' => 'your-password',
        'mailbox' => '',
        'pop3_auth_option' => 'USER',
        'imap_options' => array ('imap'),
        'debug' => false,
        'exclude_attachment_types' => array
('application/exe'),
        'default_values' => array (),
        'parts_order' => 'text/plain,text/html',
    ),

```

First of all, make sure the both entries for 'debug' are set to true, so that you can see what's happening.

Then provide some valid values at least for 'mail\_server', 'login', 'pwd' and 'port'. If your server is an POP3 server, change the 'protocol' entry from 'imap' to 'pop3' and adjust the port number from 143 to 110 (or whatever makes sense for connecting to your mail server).

To test your settings:

1. Make sure there is at least one email in the mailbox you're going to poll
2. The source address from this email must be the email address of an existing contact in iTop
3. Open a command line prompt on the server and from the directory <itop\_root>/webservices execute the command:

```
php cron.php --auth_user=<user> --auth_pwd=<pwd>
```

Windows users, run:

```
php.exe cron.php --auth_user=<user> --auth_pwd=<pwd>
```

Where <user> and <pwd> are valid credentials for an iTop account (for example an administrator account).

You should see something like this:

```
Processing Message Source: imap.demo.com GetMessagesCount returned: 1
Searching EmailReplicas: 'SELECT EmailReplica WHERE uid1 IN (1)'
Dispatching new message: 1
iTop Simple Email Synchro: dispatch of the message 0 (1)
The message is new (unread).
iTop Simple Email Synchro: dispatch of the message 0 (1) returned 2
(PROCESS_MESSAGE)
Processing new message: 1
IMAPEmailSource: Fetching the message 0
iTop Simple Email Synchro: Processing message 0 (1)
Email body format: text/plain
ERROR:Object not following integrity rules: issues = Unexpected value for
attribute 'service_id': Null not allowed,
Unexpected value for attribute 'servicesubcategory_id': Null not allowed,
Unexpected value for attribute 'workgroup_id': Null not allowed, class
= UserRequest, id =
Exiting: 1343645038
```

At this point, the ticket is NOT created but this shows that iTop is able to connect to the given mailbox and read messages from it. You're almost done.

If you see an output like:

```
Fatal error: Uncaught exception 'ConfigException' with message 'Could not read configuration file (the file exists but cannot be read).
```

```
Do you have the rights to access this file?: file =  
/opt/test/email-synchro//config-itop.php' in  
/opt/test/email-synchro/core/config.class.inc.php on line 652
```

This means that the current user does not have the rights to read the iTop configuration file. Either adjust the rights on this file or launch the 'php' command from another account (or sudo it).

## 'default' settings

The ticket creation script populates the new ticket with the following information:

1. The 'caller' of the ticket is the contact with the same email address as the sender of the email.
2. The 'org\_id' of the ticket is filled with the organization of the caller.
3. The 'title' of the ticket is filled with the 'Subject' of the email (or with 'No subject' if the subject is empty)
4. The 'description' of the ticket is filled with the 'text body' of the email (or with 'No description.' if the body is empty)
5. Email attachments are turned into attachments to the ticket.

In order to create a valid iTop User Request ticket, each other mandatory field must be filled with some valid default value. This is the purpose of the 'default\_values' array in the configuration file.

For example, using the the standard iTop 1.2.1 data model (and the Demo data) the following configuration is enough to create tickets:

```
'itop-simple-email-synchro' => array (  
    'mail_server' => 'pop3.combodo.com',  
    'port' => 110,  
    'login' => 'test@combodo.com',  
    'pwd' => 'your_pwd',  
    'mailbox' => '',  
    'imap_options' => array ('pop3'),  
    'debug' => true,  
    'exclude_attachment_types' => array  
( 'application/exe' ),  
    'default_values' => array (
```

```

        'service_id' => 'Computers and peripherals',
        'servicesubcategory_id' => 'Repair',
        'workgroup_id' => 'Hardware support',
    ),
    'parts_order' => 'text/plain, text/html',
),

```

## combodo-email-synchro parameters

Parameter	Type	Description	Default Value
debug	boolean	Whether or not to produce some debug output. Set it to true when troubleshooting, reset it to false when in production	false
save_errors_to	string	The path to a folder where to save the errors (i.e. messages that failed to decode). If left empty the errors will not be saved on disk.	
notify_errors_to	string	The email address to send notifications to about email decoding errors. If left empty, no notification will be sent in case of error.	
notify_errors_from	string	The 'from' email address to use when sending notifications. Be aware that some mail servers may reject the sending of messages if the 'from' email address is not a valid one.	
introductory_patterns	array of string	Patterns used to guess what's the 'new' part of a message that comes in reply to another. Not used!	

## itop-simple-email-synchro parameters

Parameter	Type	Description	Default Value
protocol	string	The protocol to use for connecting to the mailserver. Either "pop3" or "imap" (non-case sensitive)	'imap'
mail_server	string	Either the fully qualified domain name, or the IP address of the POP3 mail server	'imap.demo.com'
port	number	The socket/port used by the server. The default port for POP3 is 110, for IMAP it is 143. But this may be different in your environment.	143
login	string	The login used to connect to the server. This is generally the name of the 'mailbox-name' mailbox or the email address.	



pwd	string	The password used to connect to the server/mailbox. 'your-password'
mailbox	string	The 'mailbox' parameter to the imap_open call. useful if one user has access to several mailboxes on the same system
pop3_auth_option	string	Specific to POP3. Ignored if the protocol is "imap". The type of POP3 authentication mechanism to use when connecting to the mailbox. One of 'CRAM-MD5', 'APOP' , 'PLAIN' , 'LOGIN' or 'USER' 'USER'
imap_options	array of strings	Specific to IMAP. Ignored if the protocol is "pop3". A list of options to pass to imap_open. For example 'imap', 'ssl', of etc... For the complete list of options and their meaning, refer to the <a href="#">PHP documentation of imap_open</a> . Note that you don't have to put the '/' in front of the option names. array('imap')
debug	boolean	Whether or not to produce some debug output. Set it to true when troubleshooting, reset it to false when in production true
exclude_attachment_types	array of strings	By default any attachment to the email is turned into an attachment to the ticket in iTop. This list (i.e PHP array) of array ( 0 ⇒ strings corresponds to the mimeTypes 'application/exe',) for which attachments will NOT be created.
default_values	array of values	Default values to populate the newly created tickets. For external keys, values indicate the name (non-case sensitive) of the target object. array()

## Testing your configuration

1. Make sure that you set the two 'debug' flags to 'true' in the configuration file.
2. Make sure that the current user has enough right to read the configuration file
3. From the command line, go to the directory 'webservices' and launch the following command:

```
php cron.php --auth_user=<admin_user> --auth_pwd=<admin_pwd>
--verbose=1
```

The output should look like the following:

```
Starting: 1317719369
Planned duration = 600 seconds
Processing asynchronous task: ExecAsyncTask
Returned: processed 0 tasks
Processing asynchronous task: ProcessSLAResponseTicket
Returned: No ticket to process
Processing asynchronous task: EmailBackgroundProcess
Processing Message Source: test@combodo.com GetMessageCount returned:
0
Returned: Message(s) read: 0, message(s) processed: 0, message(s) deleted:
0
Sleeping
...
```

By default the script will run to 10 minutes, but you can press CTRL-C to stop the script at any time.

If you see a message like

```
Failed to initialize the mailbox: <your_login>, the mailbox will not be
polled. Reason (Cannot connect to <your_pop3_server> on port <your_port>)
```

Then check the credentials and connection to the mail server.

If you're trying to connect to the server through a SLL connection, add 'ssl' in the list of imap\_options (and adjust the 'port' as well if needed). Then if you see the following error:

```
Warning: imap_open(): Couldn't open stream {imap.demo.com:993/imap/ssl}
in
/opt/test/email-synchro/modules/combodo-email-synchro/model.email-syn
chro.php on line 774
```

Call Stack:

```
0.0007      376268    1. {main} ()
/opt/test/email-synchro/webservices/cron.php:0
0.7834     33215716    2. CronExec ()
/opt/test/email-synchro/webservices/cron.php:206
0.8142     33250056    3. EmailBackgroundProcess->Process ()
/opt/test/email-synchro/webservices/cron.php:86
0.8142     33250772    4.
iTopSimpleEmailProcessor->ListEmailSources ()
```

```
/opt/test/email-synchro/modules/combodo-email-synchro/model.email-synchro.php:1055
```

```
0.8143 33251516 5. IMAPEmailSource->__construct()
```

```
/opt/test/email-synchro/modules/itop-simple-email-synchro/model.itop-simple-email-synchro.php:71
```

```
0.8143 33251976 6. imap_open()
```

```
/opt/test/email-synchro/modules/combodo-email-synchro/model.email-synchro.php:774
```

```
Array
```

```
(  
  [0] => Certificate failure for imap.demo.com: Server name does not  
  match certificate: /OU=Domain Control Validated/CN=ssl0.mycompany.net  
)
```

This means that the server is not sending back a certificate corresponding to its DNS name. You can bypass this verification by adding the 'novalidate-cert' option to the list of `imap_options`. An example of such a configuration is shown below:

```
'itop-simple-email-synchro' => array (  
    'mail_server' => 'pop3.combodo.com',  
    'port' => 993,  
    'login' => 'test@combodo.com',  
    'pwd' => 'combodo',  
    'mailbox' => '',  
    'imap_options' => array ('imap', 'ssl',  
'novalidate-cert'),  
    'debug' => true,  
    'exclude_attachment_types' => array  
( 'application/exe'),  
    'default_values' => array (  
        'service_id' => 'Computers and peripherals',  
        'servicesubcategory_id' => 'Repair',  
        'workgroup_id' => 'Hardware support',  
    ),  
    'parts_order' => 'text/plain, text/html',  
)
```

## Ticket's default values

In order to create a User Request ticket in iTop 1.2.1 the following fields are mandatory

- Organization (org\_id)
- Caller (caller\_id)
- Title (title)
- Description (description)
- Service (service\_id)
- Service element (servicesubcategory\_id)
- Workgroup (workgroup\_id)

Organization, Caller, Title and Description are retrieved from the email and automatically filled with the corresponding value. However Service, Service element and Workgroup, cannot be retrieved from a standard email. This is why a default value must be provided for each of these fields. Note that for the ticket to be valid, the values for this 3 fields must be VALID VALUES for ORGANIZATION of the CALLER.

The 'default\_values' settings can also be used to pre-fill any other field of the ticket.

## Scheduling the automatic creation of tickets

The creation of ticket is integrated with iTop's generic background process 'cron.php'.

For information about how to schedule the background processing in iTop, refer to [Background tasks](#)

It is considered as a good practice to redirect the output of cron.php to a log file, but be sure to turn off debugging before doing so (by setting 'debug' ⇒ false in both modules) otherwise the process may fill your disk very quickly.

## Customizing the email processing

If your company uses some email templates for specific requests or processes, you may want to enhance the iTop to process such pre-formatted emails.

All the processing that creates the tickets in iTop is located in the function ProcessMessage in the file 'model.itop-simple-email-synchro.php'. Feel free to adapt the code inside this function to suit your own needs.

The module 'combodo-email-synchro' provide the generic mechanism for processing an arbitrary number of email source, listing the messages,

deconding, them... and should probably not be touched by your customization.

## Upgrading from V1 to V2

The configuration of the V2 extension is slightly different from the configuration of the V1 extension, so the new version is not a 'drop-in' replacement for the version 1.

In order to upgrade:

1. Replace the two directories modules/combodo-email-synchro and modules/itop-simple-email-synchro by their new version
2. Make a backup copy of of your configuration files and make it writeable by the web server
3. Point your browser to `http(s)://<your_itop_server>/setup` - Make sure that you check the module 'Ticket Creation from Emails' if it's not checked already
4. Once the installation completes edit the configuration file as explained above.

Note: the internal data used to maintain the linkage between the incoming emails and the iTop tickets remain the same, so the upgrade should be transparent to the iTop users.

## Mapping between the V1 and V2 configuration options

V2 option	V1 option	Explanation / Remarks
mail_server	pop3_server	Same conventions apply. You can use a DNS name or an IP address.
port	pop3_port	Same conventions apply.
login	pop3_login	The login used to connect to the server. This is generally the name of the mailbox or the email address.
pwd	pop3_pwd	The password used to connect to the server/mailbox.
mailbox		This is a new setting specific to IMAP, ignored when connecting to a POP3 server
pop3_auth_option	?	POP3 Authentication protocol. Ignored if the protocol is 'imap'
imap_options		This is a new setting specific to imap_open. Ignored if the protocol is

		'pop3'
debug	debug	Same as in V1, no change.
exclude_attachment_types	exclude_attachment_types	Same as in V1, no change.
default_values	default_values	Same as in V1, no change.

# TeemIp 1.0

Extension name TeemIp

Version 1.0

Release date 2013-03-05

Download [TeemIp-Module-1.0.zip](#)

Compatibility iTop 2.0

Keywords IP Management, IPAM, IPv4

Website [TeemIP web site](#)

TeemIp provides a simple and powerful user interface which allows network administrators to manage their IP Plan, subnet space and IPs in accordance with best in class IP Management practices:

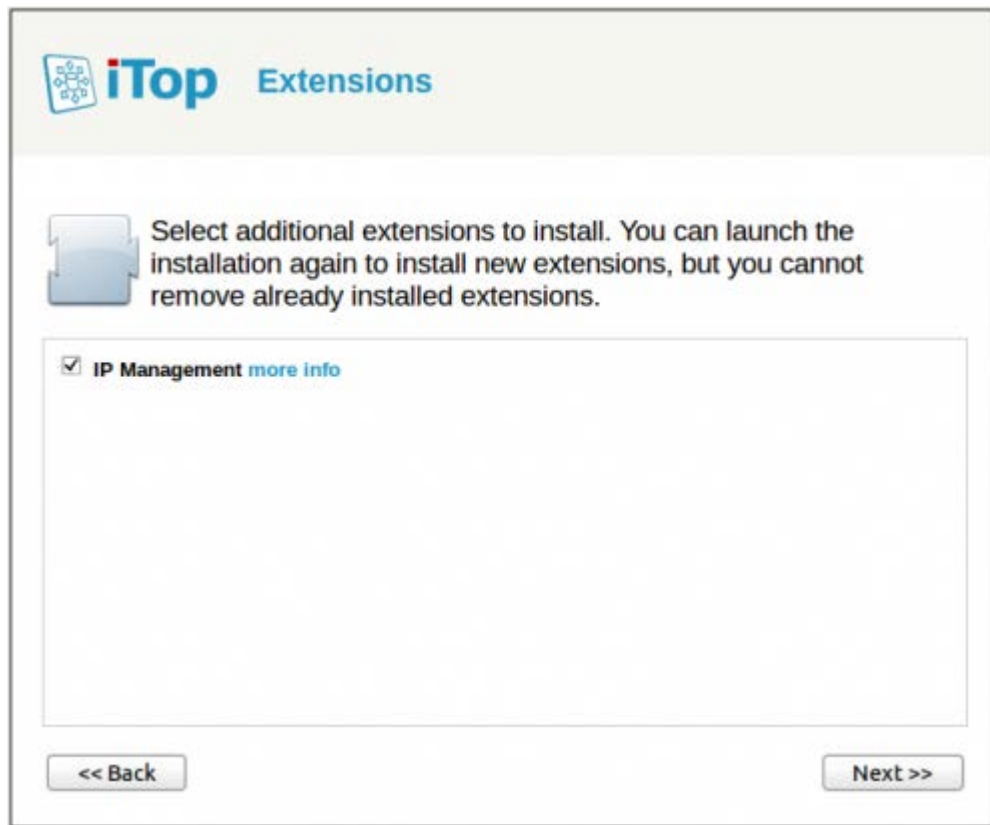
- Define your IP Plan through hierarchical Network Blocks
- Manage Subnets within predefined Network Blocks
- Attach IP Ranges to your Subnets
- Register IPs and get a clear view on the IP space consumption
- Proactively notify administrators on key events
- Synchronize your data with external tools
- and much more...

TeemIp is available either as a standalone application or as an extension to iTop 2.0.

## Installation (as an extension to iTop)

1. At the root of the iTop installation path, create (if it does not already exist) a directory named `extensions`
2. If iTop is already installed, make sure that the configuration file `conf/production/config-itop.php` is not read-only.
3. Download [TeemIp-Module-1.0.zip](#) from SourceForge
4. Expand the zip file and copy the content of the `web/extensions` directory in the `extensions` directory of iTop.
5. Launch the iTop setup by pointing your browser to `http://<your_itop_root>/setup` and follow the instructions.

6. When prompted to pick the extensions to install, select “IP management”:



If IP Management does not appear in the list of extensions (or if the whole extensions step is skipped during the setup), make sure that the web server process has enough rights to read the extensions directory.

## Mobile Access via Smartphone or Tablet

There is an Android App available for accessing parts of an iTop instance on any Android-Based device, in particular smartphones and tablets, and monitoring any information in iTop. It allows for the following

### Features

#### Ticketing

- Access to open and unassigned Incident tickets
- Access to open and unassigned Service Request (Helpdesk) tickets
- Access to tickets assigned to the person using the App

- Display priority and SLA escalation (if applicable)

## Service Delivery Monitoring (SDM)

- configure upper and lower limits for any Query in the phrasebook
- Notification (configurable) when these limits are exceeded
- Notification (configurable) for certain tickets (new e.g. Prio 3 incident)

## Examples for SDM

- Vibrate phone of Problem Manager when more than ten Incidents have been closed with “cannot reproduce” within the last two days
- Inform Management when an approved Change Request will potentially impact (another) business process within the next 30 days
- Notify Service Manager when (another) underpinning contract will terminate within the next 60 days

## Search

- simple search for Functional CIs and Person CIs
- display Query phrasebook contents
- display all Query phrasebook results

## How to install on device

The App has been written by Combodo's Partner [ITOMIG](#) and is not part of iTop itself. It is available in German language via the [Android Market](#) (including some screenshots) as a test version with access to Combodo's and ITOMIG's iTop instances. An unlimited version that allows access to any iTop instance is also available via Market, or free of charge for all of ITOMIG's iTop support customers.

If you need additional features or would like a translated version, please feel free to ask.

The app has been tested on various devices, including the Motorola Atrix (Android 2.2 and 2.3) and the Samsung Galaxy Tab (Android 3.2).



## Server Prerequisites

Your iTop instance needs to support URL authentication. How to do so is explained in the respective PDF document on the [Main Page](#) (how to set up authentication). Essentially, you need to modify the with the `allowed_login_types` to also contain `'allowed_login_types'` => `'URL|form|basic'`

It is highly recommended that you enable SSL support for your server (the App supports SSL-based access).

## Caveats

The app polls the server for tickets every `n` seconds, in an interval configurable in the app itself. The amount of data traffic generated depends on your number of tickets as well as the polling interval, although a single poll usually only consists of a few kb of data.

Please make sure you have a data plan for you mobile device. We recommend an interval of 15 minutes, which should be sufficient for most purposes.