



INFN/TC-05/13

7 Ottobre 2005

**REQUEST TRACKER:
UN SOFTWARE OPEN-SOURCE PER IL TROUBLE TICKET MANAGEMENT**

Francesca Del Corso, Riccardo Veraldi

INFN, Sezione di Firenze

Abstract

RT rappresenta una soluzione valida per la gestione delle richieste di help desk da parte degli utenti e dei certificati rilasciati dell'INFN Certification Authority. Si tratta di un pacchetto open-source, flessibile, personalizzabile e di facile manutenzione.

Di seguito vengono delineate le procedure per la sua installazione, configurazione e manutenzione, con una panoramica sull'uso del CLI per l'automazione e l'integrazione con tool esterni e di add-on aggiuntivi.

Il seguente documento vuole essere una guida operativa per il LAN Manager che intende adottare questa soluzione, in produzione presso la Sezione INFN di Firenze da oltre due anni.

PACS: 89.70.+c

INDICE

Introduzione	3
Il pacchetto Request Tracker	4
Prerequisiti	4
Installazione e configurazione	5
Come funziona	7
La web UI	7
Personalizzazioni file HTML e Perl	17
Gestione della sicurezza in RT	18
Backup e manutenzione	19
Command Line Interface (CLI)	20
Statistiche	21
Problemi riscontrati	22
Conclusioni	22
Bibliografia	23

Introduzione

Questo documento vuole essere una linea guida per l'installazione del software open-source *Request Tracker* della Best Practical LLC., utilizzato dalla Sezione INFN di Firenze per gestire le richieste di help desk inoltrate al Servizio Calcolo e Reti dagli utenti e i certificati dell'INFN Certification Authority.

La scelta di questo software è stata dettata da vari motivi: si tratta di un tool open-source, disponibile sotto la GNU GPL, personalizzabile utilizzando Perl object-oriented e HTML, estremamente flessibile essendo usabile con i più comuni sistemi di posta (sendmail, postfix), di web server (Apache, IIS), di DBMS (mysql, postgres, Oracle); è inoltre disponibile per le maggiori piattaforme Linux (RedHat, Suse, Debian), Unix (Solaris, FreeBSD), Windows (2000, XP), Mac OS X.

Il suo utilizzo è semplice e non è richiesta particolare manutenzione da parte degli amministratori: gli utenti vengono autocreati la prima volta che inviano una richiesta, evitando il loro inserimento manuale nel sistema. Inoltre attraverso un sistema integrato di interrogazione del database in cui sono memorizzati i ticket, è possibile recuperare in maniera veloce e semplice le informazioni di interesse (la history del ticket, ossia lo scambio di e-mail intercorso fra gli amministratori e chi ha richiesto l'intervento, con query ad hoc per Requestor, contenuto, ecc.)

Il tool è costantemente aggiornato e mantenuto dallo staff di sviluppo, e grazie alle mailing list, a cui tutti gli utilizzatori mondiali possono rispondere, possono essere inviate domande tecniche o apportare il proprio contributo.

Il CLI permette l'automazione e l'integrazione con tool esterni mentre il programma Crystal Report o il pacchetto aggiuntivo delle Statistiche permettono di effettuare grafici e studi sui dati raccolti.

Le note che seguono fanno riferimento alle installazioni effettuate su due macchine del Servizio Calcolo e Reti della Sezione INFN di Firenze, in produzione da oltre due anni.

Per una più esauriente guida sul funzionamento del software si rimanda alla documentazione citata nella Bibliografia e al sito ufficiale.

Il pacchetto Request Tracker

Prerequisiti

Per il corretto funzionamento dell'applicativo non sono richiesti particolari requisiti hardware; abbiamo installato il pacchetto su un Intel Pentium 4 a 3 GHz e un Intel a 2 GHz.

Sono richiesti un *DBMS* per la memorizzare di tutti gli oggetti utilizzati dal tool, un *web server*, indispensabile per il funzionamento della web UI, un *mail gateway* per l'inoltro delle richieste sotto forma di mail fra gli utenti e l'applicativo.

Il DBMS

Nel pacchetto di distribuzione non è compreso il database system poiché è lasciata agli amministratori libertà di scelta su quale DBMS utilizzare (mysql, postgres, Oracle, ecc.). Nel nostro caso abbiamo installato mysql, installato su Debian tramite il comando:

```
# apt-get install mysql-server
```

L'installazione dell'applicativo RT crea il database *rt3* (nome di default) che contiene le tabelle con gli utenti, i gruppi, i ticket, i permessi e tutti gli oggetti usati dall'applicativo (*ACL, Attachments, CachedGroupMembers, CustomFieldValues, CustomFields, GroupMembers, Groups, Links, Principals, Queues, ScripActions, ScripConditions, Scripts, Templates, TicketCustomFieldValues, Tickets, Transactions, Users, sessions*)

Il web server

Il server web che abbiamo utilizzato è *Apache2 ver. 2.0.54* (Debian GNU/Linux) e *mod_perl 1.999.21* (la versione di perl installata è la v5.8.4); viene supportato anche un webservice con FastCGI.

L'installazione del pacchetto sotto Debian installa automaticamente il web server Apache, le librerie perl ed i moduli richiesti.

Per far colloquiare il server web con RT va impostata la sezione *VirtualHost* dei file di configurazione affinché vengano caricati i moduli necessari, per esempio *HTML::Mason* su cui si basa la web UI.

Nel nostro caso (Debian con Apache2) è stata aggiunta la riga:

```
Include /etc/request-tracker3.4/apache2-modperl2.conf
```

alla fine della sezione *VirtualHost* nel file */etc/apache2/sites-available/default*.

Il mail gateway

Per funzionare il mail gateway richiede che sia installata e funzionante la web UI. Su Debian Sarge il processo di mailing viene abilitato installando il pacchetto *rt3.4-client*:

```
# apt-get install rt3-clients
```

Successivamente viene configurato opportunamente il Mail Transfer Agent, nel nostro caso *sendmail*, per istruirlo su come indirizzare i messaggi al mail gateway di RT. Per fare questo vanno aggiunte le seguenti righe nel file */etc/mail/aliases*:

```
#RT
rt: "\usr/bin/rt-mailgate --queue general --action correspond \
--url http://rt.fi.infn.it/"

# RT comment
rt-comment: "\usr/bin/rt-mailgate --queue general --action comment \
--url http://rt.fi.infn.it/"
```

Installazione e configurazione

Le istruzioni di installazione del pacchetto dipendono dalla versione di Linux usata; sulle nostre macchine è installato il sistema operativo Linux Debian Sarge ver. 3.0.2 con versioni di kernel rispettivamente Linux 2.6.8-2-686-smp e Linux 2.4.27-1-386.

Le istruzioni che seguono fanno riferimento a questa distribuzione; la versione di RT installata al momento è la 3.4 .

Il package *request-tracker3.4* fa parte dei pacchetti standard distribuiti dalla Debian, inclusi la maggior parte delle dipendenze (librerie, moduli, ecc.).

Con il comando:

```
# apt-get install request-tracker3.4
```

viene installato il pacchetto nelle diverse cartelle di sistema:

<i>/etc/request-tracker3.4</i>	contiene i file di configurazione
<i>/usr/share/request-tracker3.4/</i>	contiene i file HTML e le librerie
<i>/usr/share/doc/request-tracker3.4/</i>	contiene la documentazione
<i>/usr/local/share/request-tracker3.4</i>	contiene i file di installazione personalizzati
<i>/usr/bin/rt-3.4</i>	contenente i binari del pacchetto
<i>/var/log/request-tracker3.4/</i>	contenente i file di log

A questo punto va configurato il file */etc/request-tracker3.4/RT_SiteConfig.pm* specificando tutti i valori che si vogliono impostare a partire dal file di configurazione di default *RT_Config.pm*.

Un esempio di configurazione è mostrato in Tab. 1.

```
Set($rtname, 'Ticket');
Set($Organization, 'fi.infn.it');
Set($CorrespondAddress, 'a@b.c.d');
Set($CommentAddress, 'INFN Sezione di Firenze');

Set($MinimumPasswordLength, "8");
Set($RTAddressRegexp, '^rt\b.c.d$');
Set($CanonicalizeEmailAddressMatch, 'b.c.d$');
Set($CanonicalizeEmailAddressReplace, 'b.c.d');
```

```

Set($MailCommand , 'sendmail');
Set($SendmailArguments , "-oi ");
Set($UseFriendlyFromLine , 0);
Set($FriendlyFromLineFormat , " Servizio di calcolo e Reti");

# LOG
Set($LogToSyslog , undef);
Set($LogToScreen , 'error');
Set($LogToFile , 'debug', 'error');
Set($LogDir, '/var/log/request-tracker3.4');
Set($LogFileNamed , "rt.log"); #log to rt.log

# THE DATABASE
Set($DatabaseType, 'mysql');
Set($DatabaseUser , 'rt_user');
Set($DatabasePassword , 'rt_pass');
Set($DatabaseName , 'rt3');

# THE WEBSERVER
Set($WebPath , "");
Set($WebBaseURL , "http://rt.b.c.d");
Set($WebImagesURL , $WebPath . "/NoAuth/images/");
Set($LogoURL , $WebImagesURL . "inf-n-new.png");

```

Tab. 1. – Configurazione del file *RT_SiteConfig.pm*

Come funziona

Gli utenti che vogliono richiedere un intervento tecnico inviano un'e-mail ad un indirizzo di mail predefinito; sul server di posta queste e-mail vengono reindirizzate al server RT dove per effetto dell'impostazione nel file *aliases* (vd. pag. 5) viene eseguito il comando *rt-mailgate* e prese in gestione dal sistema; grazie al filtro impostato in */usr/local/request-tracker3.4/lib/RT/Interface/Email/Auth/MailFrom_Local.pm* vengono scartate tutte quelle mail marcate SPAM dal server di posta o inviate da indirizzi non autorizzati. Le e-mail che non soddisfano questo vincolo verranno scartate dal sistema.

Gli amministratori di RT gestiscono tali richieste utilizzando l'interfaccia grafica di RT, la Web User Interface.

L'interfaccia Web

Collegandosi alla pagina web del Servizio Calcolo, nella sezione *Gestione Ticket* veniamo reindirizzati all'interfaccia web di RT. Dopo essersi autenticati

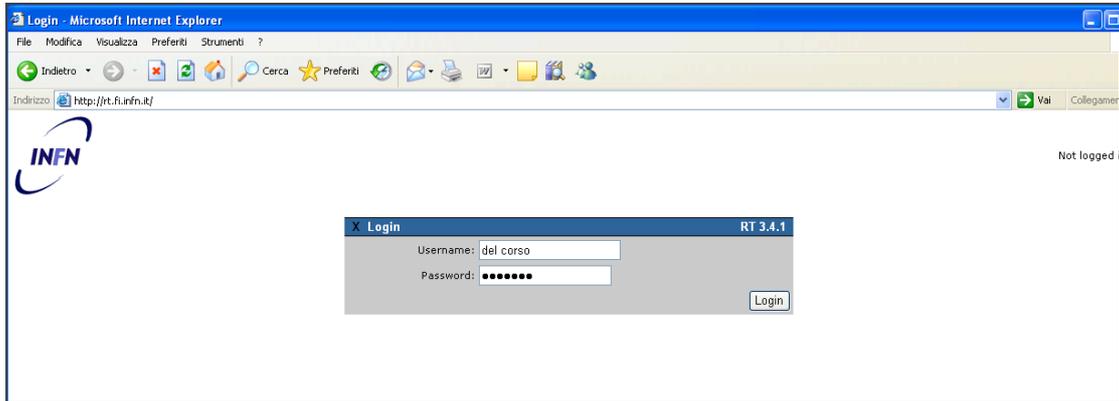


Fig. 1: Login ad RT

(Fig. 1) viene mostrata a la struttura di Fig. 2. con i permessi di amministratore.

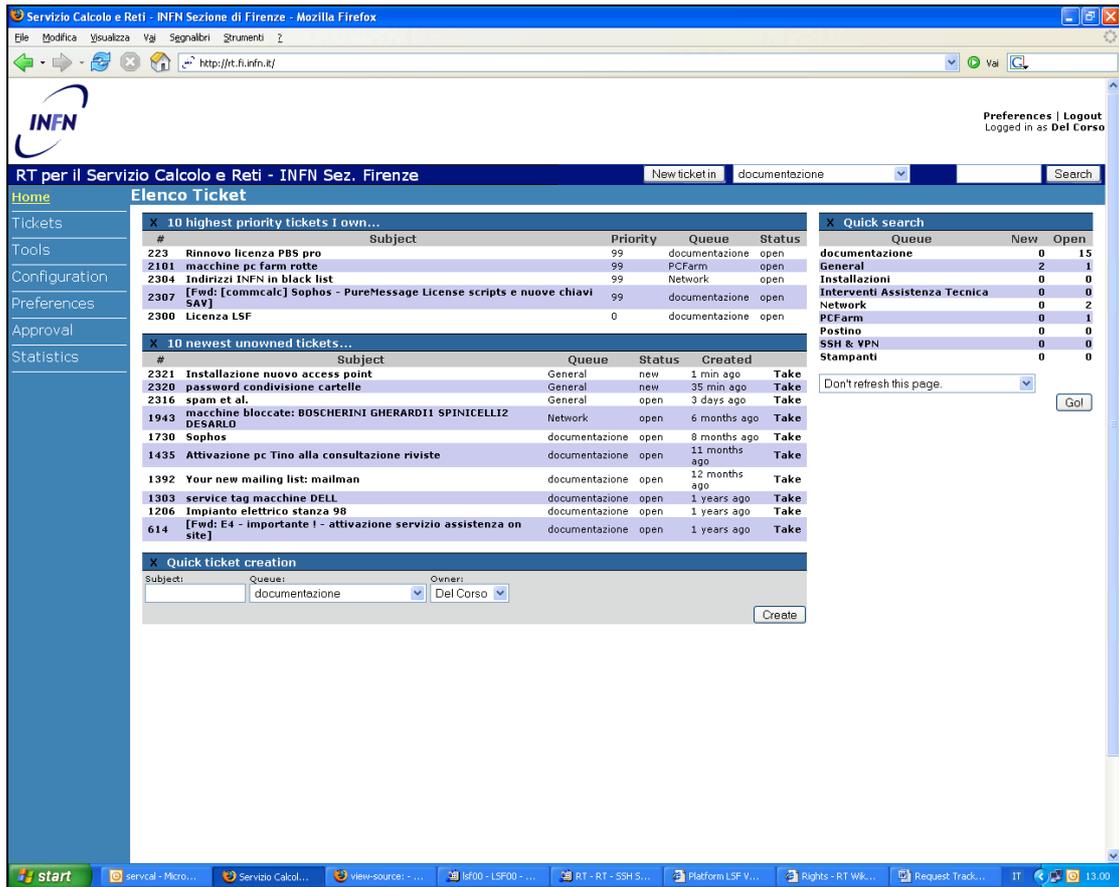


Fig. 2 – L'Interfaccia web.

Nella sezione *Home* viene visualizzato l'elenco dei ticket suddivisi per coda (*queue*) e status (nuovo, aperto). Appena un utente invia un ticket questo apparirà nella coda *General* con status *new*. Un amministratore del sistema se ne prenderà carico (*Take*) diventandone l'*Owner*, lo aprirà (*Open*) e gestirà la richiesta (Fig. 3).

In conseguenza di questa operazione al *Requestor* del ticket, colui che ha sottomesso la richiesta, arriverà un messaggio automatico del tipo:

Questo messaggio è stato automaticamente generato in risposta alla creazione di un ticket relativo alla richiesta:

"Account NAS "

Si prega di non rispondere a questo messaggio.

A questa richiesta è stato assegnato il seguente ID:

[Ticket #2334].

Per favore includere nel subject di ogni e-mail per ogni corrispondenza relativa a questo ticket:

[Ticket #2334]

Il formato del messaggio che RT spedisce è totalmente personalizzabile e può variare a seconda del tipo di transazione.

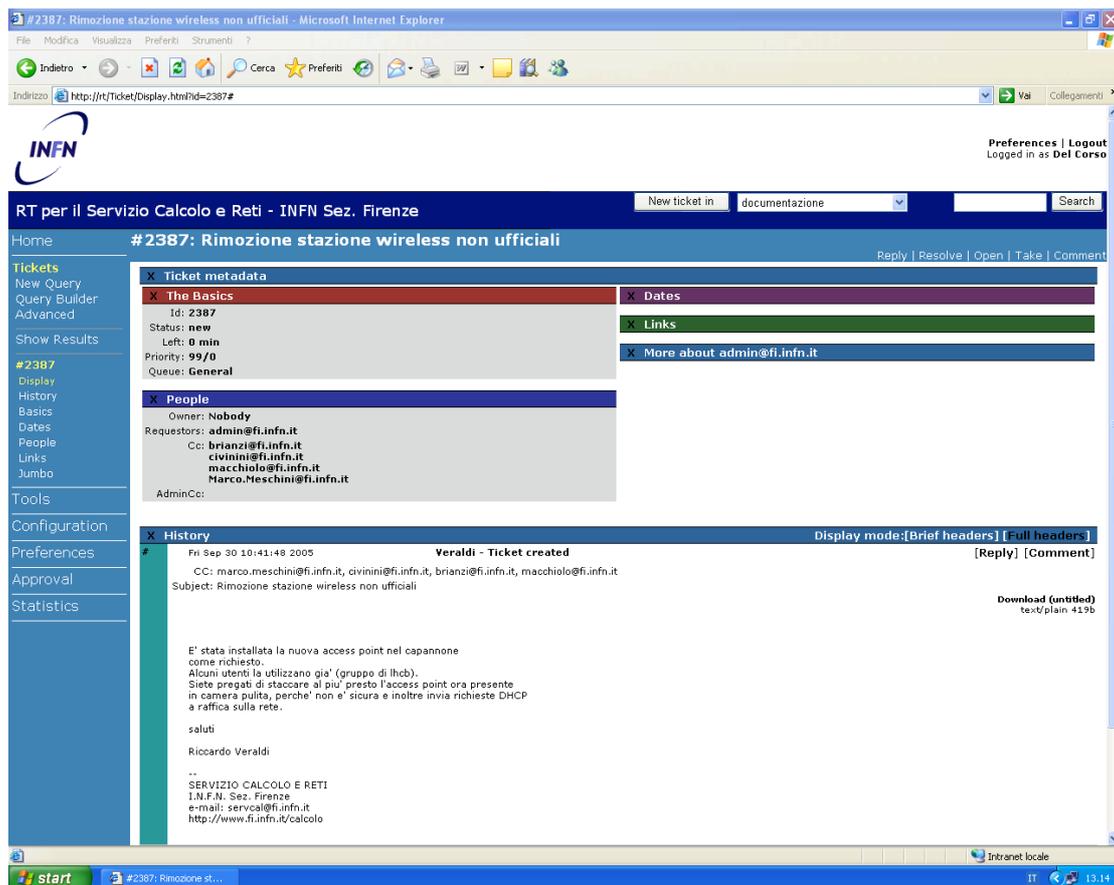


Fig. 3 – Esempio di ticket

L'*owner* del ticket viene avvisato dell'aggiornamento del ticket da una mail nella propria casella di posta grazie ad uno *scrip* opportuno.

Tutti i messaggi che intercorreranno fra l'*owner* del ticket ed il richiedente fino alla sua chiusura verranno memorizzati nella *history* del ticket. Possono essere aggiunti commenti al ticket, inviati file in attachment, eseguita la gestione del ticket solo dopo l'approvazione di un superiore (possibile con opportuna impostazione dal menu in **Approval**), spostato di coda, cambiata la priorità (*Priority*), impostati i *Watchers* (coloro che riceveranno per mail tutti i messaggi che intercorrono fra il richiedente ed il gestore del ticket ed altro ancora).

Una volta che la richiesta è stata soddisfatta il ticket verrà chiuso (*Resolved*) e al Requestor arriverà un messaggio automatico di chiusura del ticket.

Tutti i ticket chiusi possono essere eventualmente riaperti e riprocessati.

Su tutti i ticket sottomessi al sistema, *New*, *Open*, *Resolved*, rigettati (*Rejected*), cancellati (*Deleted*), in stallo (*Stalled*) può essere effettuata una ricerca dal menu di sinistra in *Tickets / New Query* : le ricerche possono essere fatte per Owner, Requestor, Data, contenuto della richiesta, ecc. (vd. Fig. 4).

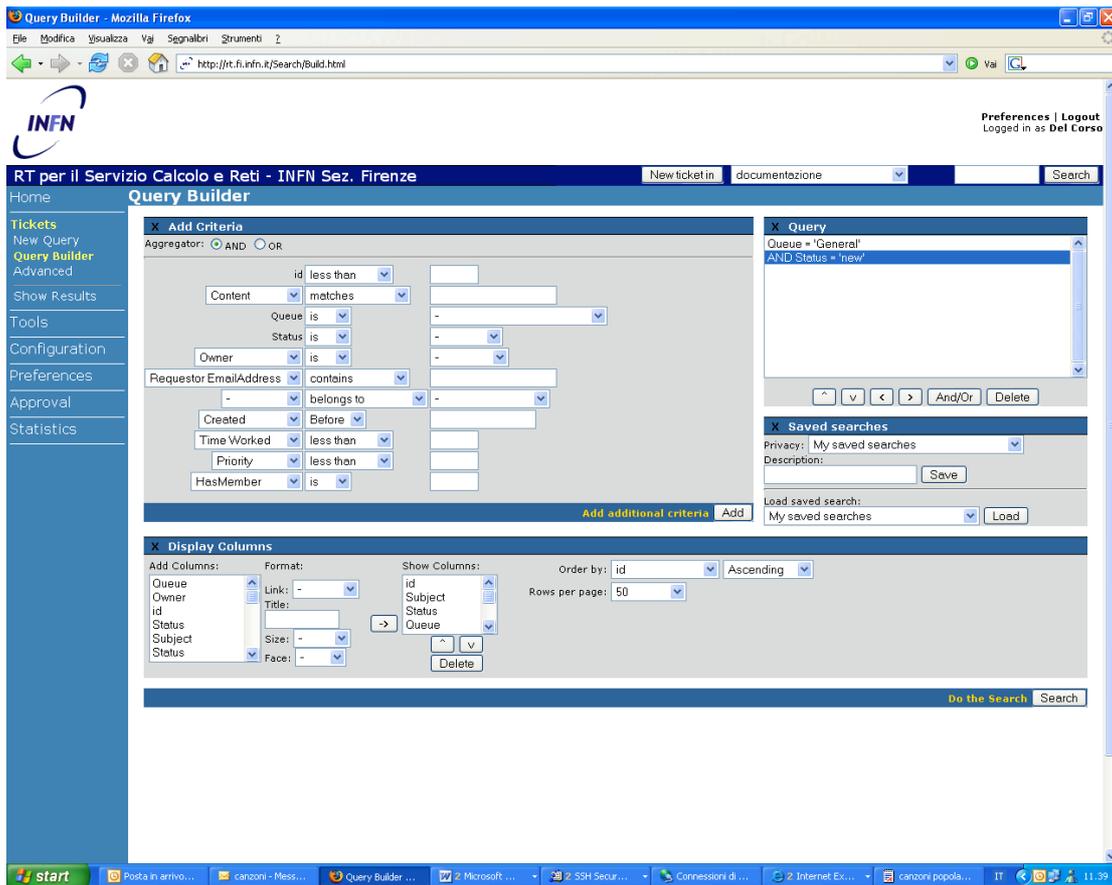


Fig. 4: Configuration /Query Builder.

Ad esempio è stata gestita una chiamata effettuata da un utente che lamentava un problema hardware su una motherboard circa 10 mesi fa, è stato aggiunto da colui che ha gestito la chiamata un commento al ticket (*Comment*) non inviato all'utente ma memorizzato nella sua history in cui aveva indicato i passi eseguiti, il numero di telefono dell'assistenza chiamata, il serial number della scheda madre. Tale ticket può essere ricercato con una query per una qualunque parola inserita nel ticket (serial number, modello motherboard, ecc.), per data, richiedente, owner, priorità, coda, ecc..

RT utilizza un database di utenti interni per memorizzare chi può accedere al sistema e quali diritti possiede. La prima cosa da fare dopo avere installato il tool è creare gli utenti che si occuperanno della sua gestione (il gruppo *RT Administrators*, nel nostro caso). Tutte le volte che un utente sottometta una richiesta RT automaticamente crea un *non staff account* per quella persona, il cui username coincide con la propria e-mail. Questi *autocreated users* avranno i permessi impostati al ruolo di Requestor. Questa feature permette di diminuire i costi di amministrazione e di gestire in maniera più efficiente il sistema.

Nella sezione **Configuration /Users / New user** vengono creati manualmente i nuovi utenti o cambiate le proprietà degli altri impostando le varie sezioni *Identity* (username, email, real name, ecc.), *Location* (organizzazione, indirizzo, stato, ecc.), *Access Control*, ecc. (vd. Fig. 5).

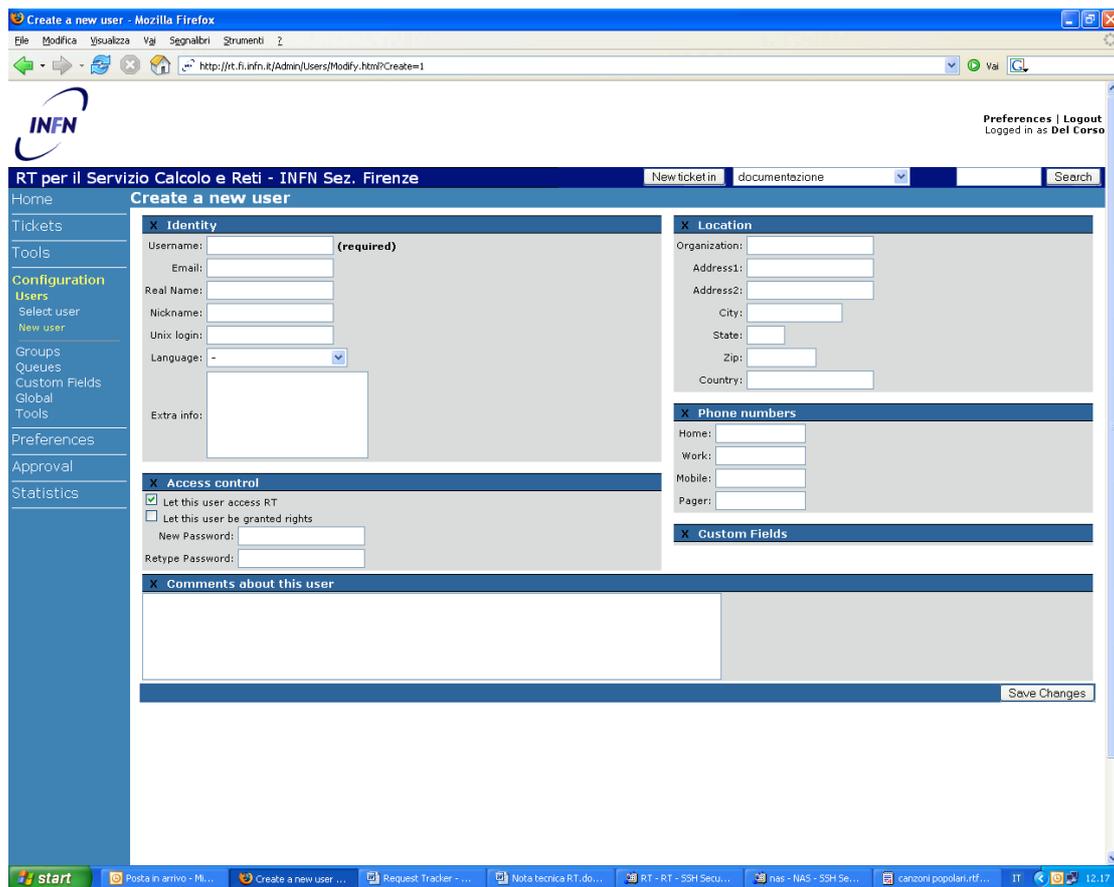


Fig. 5: Configuration / Users.

In particolare nella sezione Access Control:

Let this user access RT, per garantire all'utente l'accesso alla RT web interface; l'utente potrà in questo modo connettersi alla sua web UI e controllare lo status della sua richiesta.

Let this user be granted access: per permettere l'inserimento dell'utente nel gruppo *Privileged*, coloro che possono gestire il sistema.

New password: per impostare una password per l'utente. Gli utenti eseguendo il logon via web potranno cambiarsi questa password ed altre proprietà impostate di default.

Nella sezione **Configuration /Users /Select user** sono visualizzati i *Privileged Users* e possono essere ricercati tutti gli altri utenti, inclusi quelli autocreati, eseguendo l'opportuna query.

RT possiede diversi gruppi predefiniti:

- gruppi di sistema:
 - Everyone* – tutti gli utenti
 - Privileged* – coloro che possono configurare RT; hanno il permesso 'Let this user be granted rights' impostato
 - Unprivileged* – tutti i non privileged, tra cui gli automatic created users
- gruppi per ruolo - ogni utente è assegnato ad uno di questi gruppi nel contesto di un ticket:
 - Owner** colui che prende in gestione il ticket
 - Requestor** colui che ha sottomesso la richiesta
 - CC** colui a cui vengono inviate in CC

AdminCC l'amministratore che riceve in CC lo scambio di mail

- gruppi definiti dall'utente e gruppi personali (nel nostro caso sono stati creati due gruppi di utenti, *RT Administrators* e *Utenti INFN*, contenete tutti gli utenti INFN sia della Sez. di Firenze che di altre sezioni).

Nella sezione **Configuration / Groups / New group** possono essere creati nuovi gruppi; in **Configuration / Groups / Select group** si visualizzano i gruppi creati.

L'utilizzo dei gruppi è molto vantaggioso soprattutto per l'assegnazione dei permessi, poiché l'assegnazione degli stessi ai gruppi piuttosto che ai singoli utenti semplifica la gestione del sistema.

RT gestisce diversi oggetti come *ticket*, *Queue* (code), *Scrip*, *Template*, *CustomFields*, utenti, gruppi, ecc. Su ogni oggetto sono definiti permessi specifici affinché un utente possa compiere determinate azioni, definite a livello di gruppo (*GroupSpecificRights*), di singolo utente, a livello di coda (*QueueSpecificRights*) o in generale per tutte le code (*GlobalRights*).

A seconda quindi di ciò che vogliamo fare dobbiamo impostare opportunamente i permessi. Questo viene fatto nella sezione **Configuration / Global**.

Nel nostro caso le impostazioni sono state le seguenti:

1) al gruppo di sistema *Everyone*:

CreateTicket
ReplyToTicket
Commen OnTicket

per abilitare un qualunque utente a sottomettere ticket al sistema, ad effettuare un reply alle richieste e inviare un commento.

2) all'utente *Requestor*:

ShowTicket
SeeQueue

per dare la possibilità a colui che ha inviato un ticket di utilizzare la web UI di RT per verificarne lo stato (se è stata presa in carico, lo scambio di risposte che è eventualmente intercorso fra lui e chi ha preso in gestione il ticket (owner); rispondere allo stesso, crearne di nuovi (anziché utilizzare un client di posta), sulla coda *General* o le code a cui è dato il permesso di vedere e sottoscrivere richieste.

3) al gruppo *RT Administrators*:

SeeQueue,
ShowTicket
ShowTicket Comments
Watch
WatchAsAdminCC
OwnTicket
TakeTicket
ModifyTicket
DeleteTicket

per far sì che le persone preposte possano lavorare con RT (*Staff team*).

Nella sezione **Configuration / Queue** sono state create le seguenti code:

Documentazione: ad uso degli amministratori del sistema, contiene i ticket relativi a lavori e documentazione svolti dagli amministratori

General: è la coda di default dove vengono indirizzati tutte le mail inviate dagli utenti

Installazioni: coda di ticket per le installazioni e configurazioni pacchetti distribuiti e gestiti dal Servizio Calcolo

Interventi in assistenza tecnica: coda per i ticket gestiti dall'assistenza tecnica in outsourcing

Network: contiene i ticket inerenti problematiche di rete

PCFarm: richieste per la PC Farm (problematiche PBS, job, compilatori, ecc.)

Postino: ticket relativi a interventi sui client e server di posta.

SSH & VPN: contiene i ticket relativi a richieste di accesso ai servizi VPN e SSH gateway

Stampanti: richieste relativi alle stampanti in gestione al Servizio Calcolo di Sezione

Nella sezione **Configuration /Global /Scripts** vengono creati o modificati gli *Scrip* a livello di singola coda o globalmente per tutte le code.

Gli scrip contengono una condizione, un'azione ed un template: l'azione viene eseguita in risposta al verificarsi della condizione utilizzando il template specificato.

L'elenco degli scrip utilizzati per la coda General è mostrato in Fig. 6.

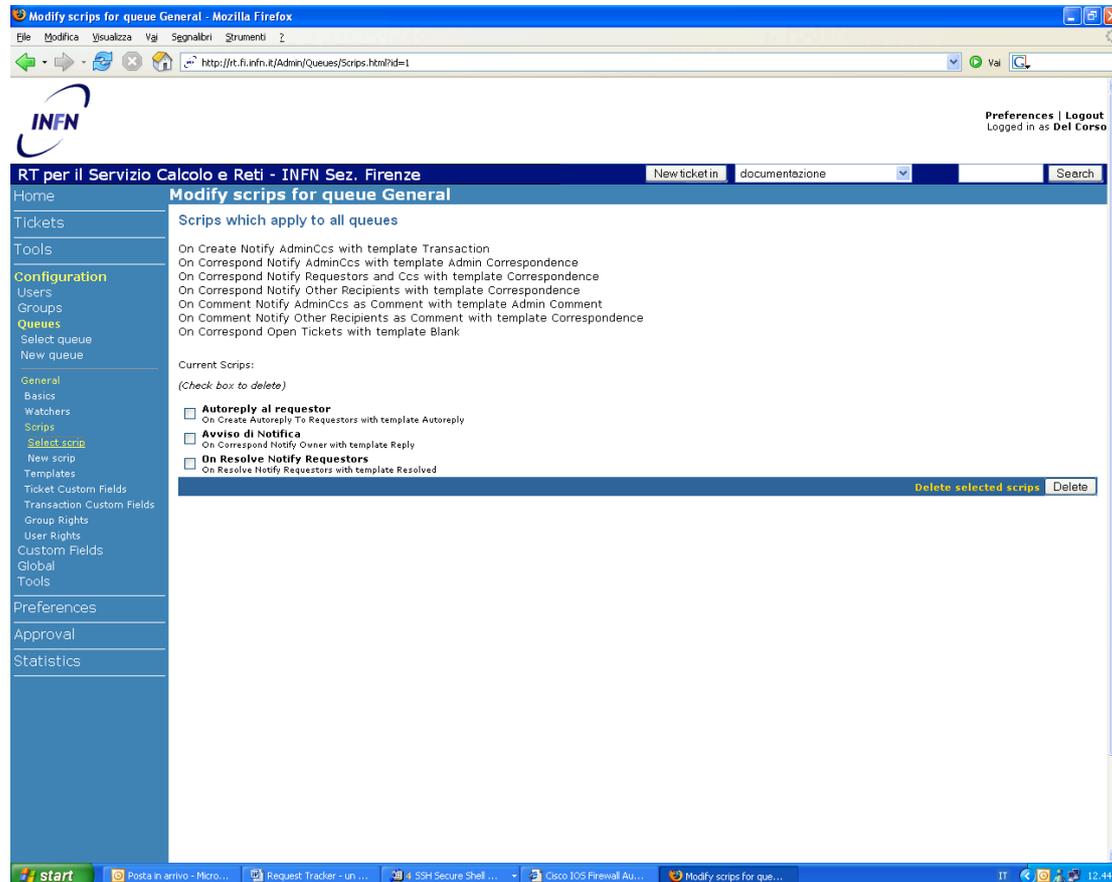


Fig. 6 - Scrip personalizzati per la coda General.

Ad esempio lo scrip *Autoreply al requestor* fa sì che venga inviata una mail, in un determinato formato e contenuto, al Requestor del ticket per avvisarlo che la sua richiesta è stata presa in carico da un gestore di RT .
Se si vuole che un ticket risolto non venga riaperto se il Requestor esegue un reply va disabilitato lo scrip: *On Correspond Open Ticket with template Blank*

Nella sezione **Configuration /Global /Templates** vengono creati o modificati i *Templates* utilizzati dagli Scrip (vd. Fig. 7).

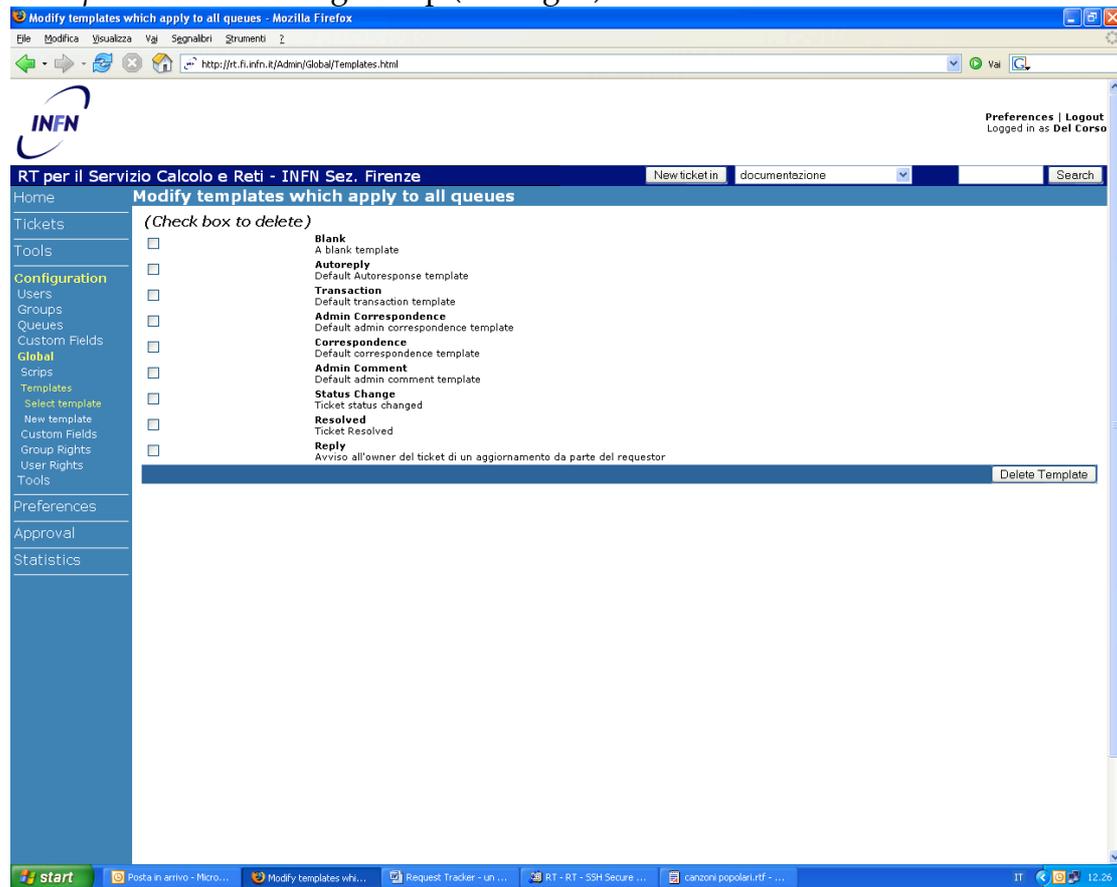


Fig. 7. Elenco Templates applicabili a tutte le code.

Anche i Template come gli Scrip possono essere applicati globalmente a tutte le code o singolarmente e permettono di personalizzare il contenuto ed il formato di una mail in risposta ad un determinato evento.
In Fig. 8 è mostrato il template che gestisce il messaggio di notifica via mail ricevuto dall'owner del ticket in risposta ad un suo aggiornamento.

Name:	Reply
Description:	Avviso all'owner del ticket di un aggiornamento da parte del requestor
Content:	<pre>Subject: Notifica aggiornamento E' stato inviato un aggiornamento per il ticket (\${Ticket->id()}). Queue: (\${Ticket->QueueObj->Name}) Subject: (\${Transaction->Subject \$Ticket->Subject "(No subject given)"} Requestors: (\${Ticket->RequestorAddresses}) Status: (\${Ticket->Status})</pre>

Fig. 8. Modifica di un Template

Personalizzazioni file HTML e Perl

Alcuni tipi di personalizzazioni sono possibili solo andando ad agire sui file sorgenti HTML e Perl. Nel nostro caso le principali personalizzazioni hanno riguardato:

1. filtro antispam:

in `/usr/share/request-tracker3.4/lib/RT/Interface/Email/Auth/MailFrom_Local.pm`
è stato aggiunto il seguente codice:

```
# ---- aggiunta per filtrare le e-mail in arrivo a rt-mailgate-----
my $head = $args{'Message'}->head;
my $Subject = $head->get('Subject') || '';
chomp $Subject;
if ( ( $Subject =~ /\*SPAM\*/ ) ||
    ( $Subject =~ /RAV Antivirus\.: risultati di scan/ ) ||
    ( $Subject =~ /whisker \ log/ )
  ) {
    $RT::Logger->crit( "Auth::MailFrom_Local: probable SPAM: $Subject" );
    return ( $args{'CurrentUser'}, -1 );
}
# Regola per le e-mail dell'antivirus Sophos
if ( $Address =~ /viralalert/)
{
    $RT::Logger->crit( "Auth::MailFrom_Local e-mail da Sophos antivirus:
$Address $Name" );
    return ( $args{'CurrentUser'}, -1 );
}
if ( ! (
    ( $Address =~ /infn\.it$/) ||
    ( $Address =~ /arcetri\.astro\.it$/) ||
    ( $Address =~ /unifi\.it$/) ||
    ( $Address =~ /ff\.oc\.uh\.cu$/) ||
    ( $Address =~ /cern\.ch$/ )
  )
  ) {
    $RT::Logger->crit( "Auth::MailFrom_Local wrong address: $Address $Name" );
    return ( $args{'CurrentUser'}, -1 );
}

# -----fine aggiunta-----#
```

2. In `/usr/share/request-tracker3.4/lib/RT/Interface/Email.pm` è stato impostato il campo password di `NewUser` ad usare l'e-mail.
3. In `/usr/share/request-tracker3.4/Header`, `/usr/share/request-tracker3.4/html/Elements/PageLayout` e `/usr/share/request-tracker3.4/html/index.html` sono state apportate le modifiche al logo presentato nella web UI ed il link alla URL del sito di riferimento.
4. In `/usr/share/request-tracker3.4/libexec/webmux.pl` è stata aggiunta alla linea 112 la riga:

```
system("touch $RT::MasonDataDir/obj/fake_file")
```

per ovviare ad un problema riscontrato in Apache2.
La MasonDataDir è */var/cache/request-tracker3.4/mason_data*.

5. timeout di sessione per la web UI: in */usr/share/request-tracker3.4/html/Elements/SetupSessionCookie* aggiunta la riga
-expires => '+2h' nel codice :

```
.....  
if( !$cookies{'RT_SID'}) {  
    my $cookie = new CGI::Cookie(  
        -name => 'RT_SID',  
        -value => $session{_session_id},  
        -expires => '+2h',  
        -path => '/',  
    );
```

6. Per la visualizzazione dei ticket stalled vanno modificate in */opt/rt3/share/html/Elements/* i file *MyTickets*, *MyRequests*, *Quicksearch* aggiungendo il valore stalled:

```
Status = 'new' OR Status = 'open' in Status = 'new' OR Status =  
'open' OR Status = 'stalled'
```

In tutti questi casi ogni volta che viene effettuata una modifica per renderla effettiva va riavviato il server web Apache:

```
# /etc/init.d/apache2 restart
```

Gestione della sicurezza in RT

Per connettersi in maniera sicura alla web RT UI utilizzando il protocollo HTTPS (HTTP over SSL) è stato installato `mod_ssl 2.0.54` e `OpenSSL/0.9.7e`.

La procedura seguita in Debian ha reso necessari i seguenti passi:

1. creato un certificato temporaneo lanciando lo script `apache2-ssl-certificate`; sono così creati due file che dovranno essere messi in `/etc/apache2/ssl`, `apache.pem`, contenente sia la chiave pubblica che quella privata, e 1 altro file che è un link simbolico allo stesso file.

2. copiato il file `/etc/apache2/sites-available/default` in `/etc/apache2/sites-available/ssl`:

```
# cp /etc/apache2/sites-available/default ssl
```

3. creato in `/etc/apache2/sites-enabled` un link simbolico di nome `ssl` al file `/etc/apache2/sites-available/ssl` precedentemente creato:

```
# ln -s /etc/apache2/sites-available/ssl /etc/apache2/sites-enabled/ssl
```

4. configurato il file `/etc/apache2/ports.conf`:

```
# vi /etc/apache2/ports.conf
```

```
Listen 80
```

```
Listen 443
```

5. apportate le seguenti modifiche al file `/etc/apache2/sites-available/ssl` :

```
NameVirtualHost *:443
```

```
<VirtualHost *:443>
```

```
LoadModule ssl_module /usr/lib/apache2/modules/mod_ssl.so
```

(se viene già caricato in `/etc/apache2/mods-available/ssl.load`, non occorre specificarlo qui).

```
.....
```

```
SSLEngine On
```

```
SSLCertificateFile /etc/apache2/ssl/apache.pem
```

```
.....
```

```
Include "/etc/request-tracker3.4/apache2-modperl2.conf"
```

```
</VirtualHost>
```

6. far ripartire il server Apache:

```
# /etc/init.d/apache2 restart
```

Backup e manutenzione

E' stato implementato un sistema di backup giornaliero del database rt3 usato da RT su un server NAS; per i file di configurazione di RT (/etc/request-tracker3.4/RT_SiteConfig.pm) e di Apache2 basta fare una copia di riserva ogni volta che vengono modificati.

Dentro Mysql sul server RT è stato creato l'utente *backup* che esegue il backup del database in una cartella locale; quindi l'utente *backupRT* locale alla macchina copia il file prodotto dal server RT al NAS utilizzando ssh con RSA;

```
# cd /root/.ssh/  
# ssh-keygen -t rsa -N "" -f id_rsa
```

vengono generati i file *id_rsa* (contenente la chiave privata) e *id_rsa.pub* (che contiene la chiave pubblica) per l'utente *backupRT* che effettuerà la copia del file sul NAS.

```
# scp id_rsa.pub backup@x.y.z:.ssh/authorized_keys
```

Il file contenente la chiave pubblica dell'utente viene copiato nella cartella *.ssh* della home directory dell'utente *backupRT* sul NAS e ridenominato *authorized_keys*.

In */etc/request-tracker3.4/backup* sono conservati i file di backup del db rt3. */etc/cron.daily/backupRT* mostrato in Tab. 2 è lo script per fare il backup del database rt3 su NAS.

```
#!/bin/sh  
##-----  
-----  
## Backup giornaliero di Mysql database rt3 di RT su NAS in  
/home/backupRT/RT  
##-----  
-----  
DIR=/etc/request-tracker3.4/backup  
TIME=rt3_`date "+%H:%M_%d-%m-%y"`  
/usr/bin/mysqldump -u backup --password=XXX rt3 > $DIR/$TIME.sql  
scp $DIR/$TIME.sql backupRT@nas:RT
```

Tab. 2 – script *backupRT*

Command Line Interface (CLI)

Il *Command Line Interface* permette di interagire con una macchina su cui è presente un'installazione di RT per l'automazione e l'integrazione con altri tool. Ad ogni invocazione del comando `/usr/bin/rt-3.4` viene specificata un'azione da eseguire su uno o più oggetti con uno o più argomenti.

Le azioni supportate, talune specifiche di determinati oggetti, sono:

show, create, list, edit, grant, revoke, link, merge, comment, correspond, mentre gli oggetti gestibili via CLI sono: *ticket, users, groups, queues*.

Per ogni oggetto sono supportati tutta una serie di attributi, ad esempio per l'oggetto *utente* gli attributi sono: *id, Name, Password, EmailAddress, RealName, Comments*, mentre per l'oggetto *ticket*: *id, Queue, Owner, Subject, Status, Priority, InitialPriority, FinalPriority, Requestors, Cc, AdminCc, Starts, Started, Due, Resolved, Told, TimeEstimated, TimeWorked, TimeLeft*.

Esempi di comandi supportati:

```
# ./rt ls -t ticket "Priority < 5"          elenco ticket con priorità < 5
# ./rt ls -t ticket "Status='open'"        elenco ticket con status open
# ./rt edit ticket/2 set status=open        impostazione status da new a open
                                           del ticket con ID 2

# ./rt help <topic>                        help dei comandi
# ./rt <comment | correspond> [options] <ticket-id>  sintassi del comando rt
# ./rt ls "Priority > 5 and Status='new'"      lista dei ticket con
                                           status new e priorità >
                                           5
# ./rt ls -i "Priority < 5" | ./rt edit - set status=resolved  impostazione a
                                           resolved dei ticket con
                                           priorità < 5

# ./rt ls -t ticket "Subject like '[PATCH]%"
# ./rt show -t ticket -f id,subject,status 1-3
# ./rt show ticket/3/attachments/29
# ./rt show ticket/1-3/links
# ./rt show -t user 2
# ./rt correspond -m 'aspetto tue notizie' k    esegue un reply al ticket id k
                                           con messaggio "Aspetto tue
                                           notizie"

# ./rt edit ticket/2 set status=open          cambio status per il ticket ID
                                           da new ad open
# ./rt create -t ticket set subject='Messaggio da CLI' owner='root'
requestors=delcorso@fi.infn.it' queue='general' priority=20
                                           Creazione ticket con soggetto "Messaggio
                                           da CLI", richiedente delcorso@fi.infn.it per la
                                           coda general e priorità 20
# ./rt show ticket/26                        mostra tutti i dettagli del ticket numero 26
```

Statistiche

Attualmente sulle entrambe le macchine è installato il package per le Statistiche di RT *Statistics3-0_1_5.tgz*, ver. 0.1.5.

Per l'installazione su Linux Debian il pacchetto è stato scompattato e messo in `/usr/local/share/request-tracker3.4/` nelle cartelle `html/` e `lib/`.

Per far funzionare la parte grafica è necessario installare le seguenti librerie: *libgd-gd2-noxpm-perl*, *libgd-graph-perl*, *libgd-text-perl*, *libgd2-noxpm*, quindi riavviare il server web:

```
/etc/init.d/apache2 restart
```

Dopo l'installazione nella barra di sinistra del web RT compare il tab **Statistics**. Le statistiche si dividono in: Ticket per giorno, Ticket per status (nuovi, aperti, risolti), Ticket Trends by Day, Time to Resolve, Resolve Time Graph.

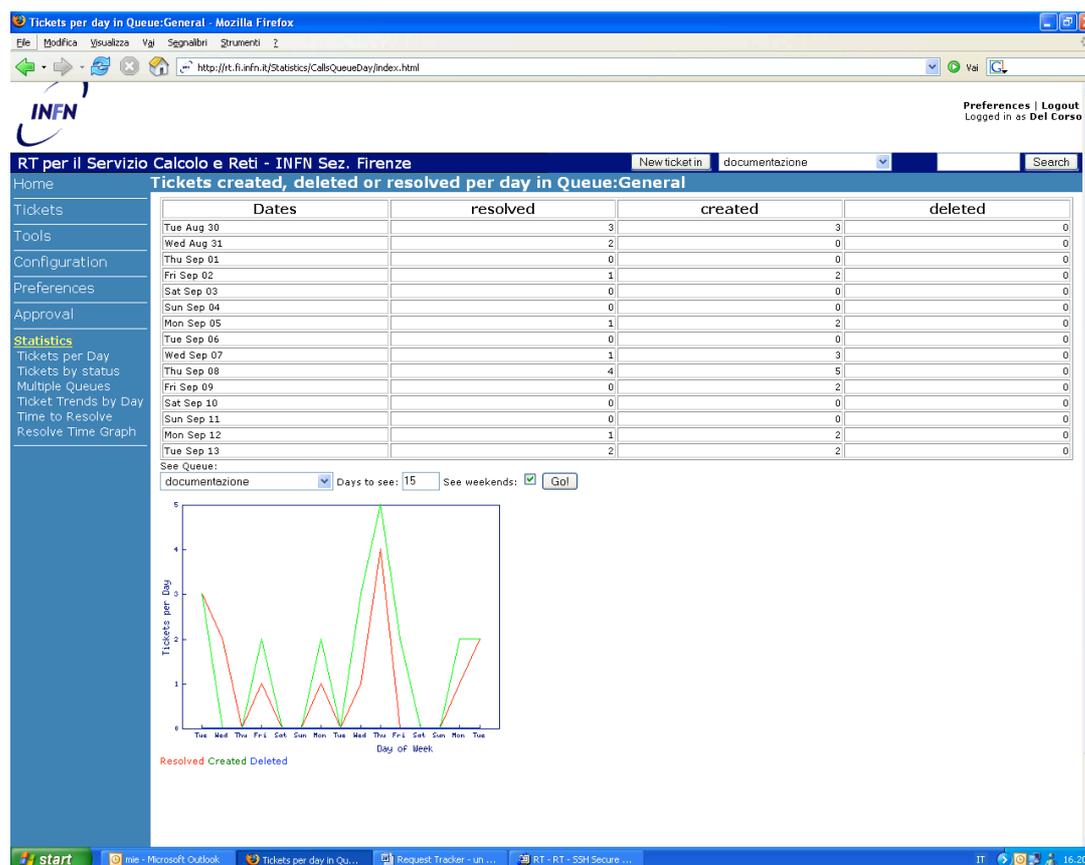


Fig. 9 . Grafico ticket creati e risolti negli ultimi 15 giorni.

Problematiche riscontrate

I problemi più frequenti che sono stati riscontrati sono legate all'estrema flessibilità del prodotto: talvolta le ultime versioni rilasciate presentano dei malfunzionamenti sulla piattaforma che si decide di utilizzare; questo tipo di problema è molto meno presente se si utilizzano le versioni pacchettizzate del prodotto, ossia le versioni rilasciate per Linux Debian e FreeBSD.

Avvalendosi del supporto delle mailing list si ottengono comunque informazioni importanti al loro superamento.

Effettuando installazioni su Linux RedHat sono stati riscontrati problemi di *memory leak* e di conseguenza di perdita di performance da parte della

macchina dovuti in massima parte ad una cattiva gestione della memoria da parte di mysql.

Conclusioni

RT si presenta come un tool open-souce, disponibile sotto la GNU GPL, personalizzabile utilizzando Perl object-oriented e HTML, utilizzabile sia per la gestione delle richieste di Help Desk da parte degli utenti di sezione sia per la gestione dei certificati dell'INFN Certification Authority.

Lo strumento è estremamente flessibile, utilizzabile con i più comuni sistemi di posta (sendmail, postfix), di DBMS (mysql, postgres, Oracle), è disponibile per le maggiori piattaforme: Linux (RedHat, Suse, Debian), Unix (Solaris, FreeBSD), Windows (2000, XP), Mac OS X (<http://www.osxgnu.org/>).

Il funzionamento è molto semplice: l'utente invia una e-mail al supporto tecnico; il sistema genera, se non presente, automaticamente l'utente per cui non è richiesto da parte degli amministratori, la loro registrazione manuale.

E' continuamente migliorato e mantenuto dallo staff di sviluppo, e avvalendosi delle mailing list, a cui tutti gli utilizzatori mondiali di RT possono rispondere e quindi non solo gli sviluppatori o il personale tecnico della Best Practical, possono essere indirizzate domande tecniche o apportare il proprio contributo

Esiste inoltre la possibilità di avvalersi del supporto tecnico e di corsi organizzati, a pagamento.

Bibliografia

Sito ufficiale di RT:

<http://www.bestpractical.com>

Sito per le statistiche:

<http://wiki.bestpractical.com/index.cgi?RT3StatisticsPackage>

RT Essentials - Jesse Vincent, Dave Rolsky, Darren Chamberlain & Richard Foley, Robert Spier, O'Reilly Media, Agosto 2005,

<http://www.oreilly.com/catalog/rtelements/chapter/index.html>

Mailing lists:

lists.bestpractical.com

rt-user@lists.bestpractical.com

rt-devel@lists.bestpractical.com