RIS d.o.o.
Address: Pilepcic 10, 51215 Kastav, Croatia
Phone: +385 51 687 500, Fax:+385 51 687 501
e-mail: info@ris.hr, web: http://www.ris.hr

Last Update: August 2015

Thin@ Software as a Service Model (SaaS) V3.2

Contents

Thin@ support for hosted IT service: Software as a Service (SaaS)

Introduction to SaaS

Software as a Service (SaaS) is a software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network or over the Internet.

SaaS is becoming an increasingly prevalent delivery model in particularly for small and medium enterprises (SMEs). The trend has been supported by the maturation of underlying enabling technologies: Web supporting services, service-oriented architecture (SOA) and new development tools.

Thin@ and SaaS

Thin@ fully supports the final SaaS maturity level (SaaS maturity level 4), as well as the benefits of the SaaS model.

In the following chapters you can find out what are the common benefits of the SaaS model, what are SaaS maturity levels, how does Thin@ support the most frequent SaaS billing options, what is the Thin@ advanced usage-based billing calculation and, finally, how to transform your application into a Thin@ SaaS Level 4 application.

Benefits of the SaaS model

- **Lower IT costs**

When a company subscribes to a SaaS application, it avoids the overhead associated with implementing conventional software. A typical software implementation involves purchasing and maintaining servers, housing them securely, and installing and maintaining the software. This requires the time and effort of experienced IT personnel and deflects the efforts of employees at a number of levels away from the core mission of the company. According to reputable technology analysts the cost of implementing conventional enterprise software is four to five times the cost of the original license.

- **Economies of scale**

Subscription costs for SaaS applications reflect the economies of scale achieved by “multitenancy.” Multi-tenancy means that many customers run their applications on the same unit of software.

- **Pay as you go**

When a company subscribes to a SaaS application, it pays a monthly or annual subscription fee. Compared to a traditional software license and maintenance this subscription payment structure brings substantial savings.

An on-going monthly expense is easier to incorporate into the budget than a large one-time outlay. The subscription can be canceled or changed at any time without losing a large initial investment.

There are a few SaaS billing options, but the central idea behind them all is that a user/company is charged on the basis of metered usage (see chapter Billing options for more information about the most common SaaS billing options).

- **Save time**

Because the user eliminates many of the typical implementation tasks associated with licensed software and because the software is already up and running on the provider data center, deployment time tends to be much shorter (5 to 10 times) with a SaaS application than a traditional one.

- **Focus technology budgets on competitive advantage rather than IT infrastructure**

When a company subscribes to a hosted application, it frees the organization from supporting high-cost, time-consuming IT functions, including:

- Purchasing and supporting the server infrastructure necessary to install and maintain the software in-house.
- Providing the equipment redundancy and housing necessary to ensure security, reliability, and scalability.

- Maintaining a labor-intensive bespokes and upgrade process.

- **Gain immediate access to the latest innovations**

With traditional licensed software, a company typically has to wait for the next release to benefit from the latest innovations or to move to a new browser or operating system. Given the cost and complexity of moving to a new version, it may not even be practical to upgrade each time a new release becomes available.

With a SaaS subscription, on the other hand, a company benefits from innovations on an on-going basis. As soon as a new or improved feature appears in the application, it can start using it.

- **Join a community of interest**

Purchasing a traditional software license is very much an individual affair. When a company subscribes to a SaaS application, however, it becomes a member of a community that has the application at the center. SaaS changes the relationship between software vendors and ensures a convergence of interest between customer, its peers and vendor (provider).

- **Business benefits to end users**

Instead, more and more companies want to consume software as a service, and are using the SaaS can cover all areas of business activities but the most recommendable are mandated applications, especially Customer Relationship Management, Spend Management, and those that fall under the IAS (International Accounting Standards) compliance umbrella - such as financials and human resources. By using the SaaS model, the Company ensures that all users are using the correct application software version and, therefore, the format of the information being recorded and conveyed is accurate and transparent.

- **Business benefits to employers**

Although usage metrics, a standard part of almost every SaaS system, is primarily used to determine the service fee, service providers can produce application usage-based reports per user, which can help employers to determine how much their employees are actually (actively) using the application, which is always a valuable information for the Company management.

SaaS Maturity Levels

There are four SaaS “maturity levels“, whose key attributes are configurability, multi-tenant efficiency and scalability. Each level is distinguished from the previous one by the addition of one of those three attributes:

- **Level 1 - Ad-Hoc/Custom**

At the first level of maturity, each customer has its own customized version of the hosted application and runs its own instance of the application on the host's servers. Migrating a traditional non-networked or client-server application to this level of SaaS typically requires the least development effort and reduces operating costs by consolidating server hardware and administration.

- **Level 2 – Configurable**

The second maturity-level provides greater program flexibility through configurable metadata, so that many customers can use separate instances of the same application code. This allows the vendor to meet the different needs of each customer through detailed configuration options, while simplifying maintenance and updating of a common code base.

- **Level 3 - Configurable, Multi-Tenant-Efficient**

The third maturity level adds multi-tenancy to the second level, so that a single program instance serves all customers. This approach enables more efficient use of server resources without any apparent difference to the end user, but ultimately comes up against limits in scalability.

- **Level 4 - Scalable, Configurable, Multi-Tenant-Efficient**

The fourth and final SaaS maturity level adds scalability through a multitier architecture (usually three-tier) supporting a load-balanced farm of identical application instances, running on a variable number of servers. The provider can increase or decrease the system's capacity to match demand by adding or removing servers, without the need for any further alteration of applications software architecture.

NOTE: SaaS Maturity Level 4 is fully supported by Thin@.

SaaS most common billing options

- **Subscription plans**

This billing model seems to be the mainstream for SaaS at the moment. One benefit for SaaS provider is a predictable revenue stream. This model is well understood by customers, but may not be ideal for customers who are only need the service sporadically. It also means a heavy-usage customer may be prevented from using the service simply because he is over the quota.

- **Usage-based billing**

In this model a service is essentially charged on the basis of how much it is being used. Customers pay only what they consume, which can mean varied month-over-month charges.

- **A subscription with a per-usage charge over quota**

A preset number of "usage" is included into the subscription plan and a customer is changed on a per-use basis when he goes over the quota.

- **A running account**

In this model a customer maintains a balance in an account with the service provider. This account is charged on usage basis and is refilled when it runs empty. This can be done manually (in which case the customer won’t be able to use the service until he refills the account) or automatically if the customer is willing.

- **Pre-paid tickets**

In this model a customer buys a pre-paid ticket with a fixed amount of money on it. The ticket usually has a unique ID which the customer needs to enter in order to access the hosted application for the first time. After using-up the ticket, the customer will have to buy a new ticket to continue to use the service.

NOTE: Thin@ currently supports *Usage-based billing* and *subscription with a per-usage charge over quota billing option,* and the other billing options will be implemented in the upcoming version of the software.

Thin@ advanced usage-based billing calculation

- **Only user activity time taken into consideration**

Thin@ uses an advanced usage-based billing algorithm, which takes into consideration only the user activity time (the total time a user was actively doing something in the application), and not the user session time (the total time a user session lasted).

The Thin@ usage calculation formula:
User activity time = Session Time – Inactive Time – Report Preview Time – Long Process Jobs

- **20 seconds default billing control unit**

The default billing control unit is 20 seconds, which means that any user action in a 20-seconds time frame will add 20 seconds to the Total Billing time.

- **Random keyboard and mouse clicks not calculated**

Thin@ does not take into consideration mouse and keyboards clicks that did not trigger any event in the application, meaning that only the meaningful interaction between the user and the application is calculated.

- **Timer events not calculated**

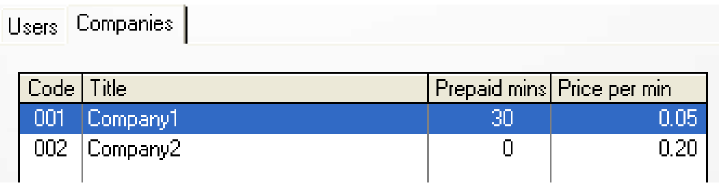
If your application users automatic timer events, those are not calculated as user actions.

In this way, customers really pay only for the time they were actively using the application, and not for the time they were logged in but were not using the application, which solves one of the primary customer concerns regarding SaaS.

Transforming your Thin@ application into a Thin@ SaaS Level 4 application

Transforming your Thin@ application into a Thin@ SaaS Level 4 application is quick and easy.

All you need to do is set the price per minute and, optionally, prepaid minutes for each company.



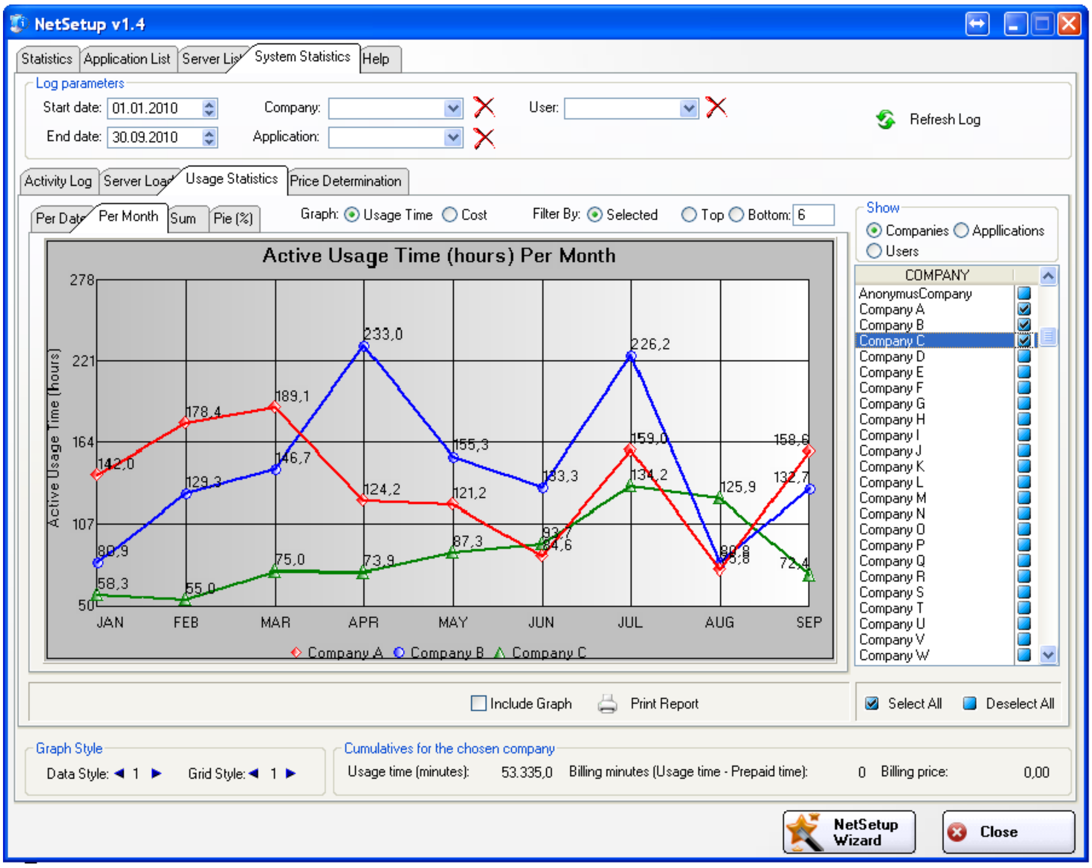
Users | Companies

Code	Title	Prepaid mins	Price per min
001	Company1	30	0.05
002	Company2	0	0.20

Prepaid mins (monthly): The number of minutes a company pays in advance each month for using Thin@ application(s).
Price per min: The price per each minute of active Thin@ application usage.

NOTE: If you set the number of Prepaid minutes for a company to a value different than zero, you are basically using the **subscription with a per-usage charge over quota** billing option for that company. In that case, be careful to set the start date to the first day of the month, and the end date to the last day of the month, in order for the calculation to be correct.
If you set this value to zero, you are using the classic **Usage-based billing** for that company.

The picture below shows just an example of what you can do with the Usage Statistics section of the System Statistics module.



NetSetup v1.4

Statistics | Application List | Server List | System Statistics | Help

Log parameters
Start date: 01.01.2010
End date: 30.09.2010
Company:
Application:
User:

Activity Log | Server Load | Usage Statistics | Price Determination

Per Date | Per Month | Sum | Pie (%)

Graph: ● Usage Time ○ Cost

Filter By: ● Selected ○ Top ○ Bottom: 6

Show
● Companies ○ Applications
○ Users

COMPANY
AnonymusCompany
Company A ☑
Company B ☑
Company C ☑
Company D
Company E
Company F
Company G
Company H
Company I
Company J
Company K
Company L
Company M
Company N
Company O
Company P
Company Q
Company R
Company S
Company T
Company U
Company V
Company W

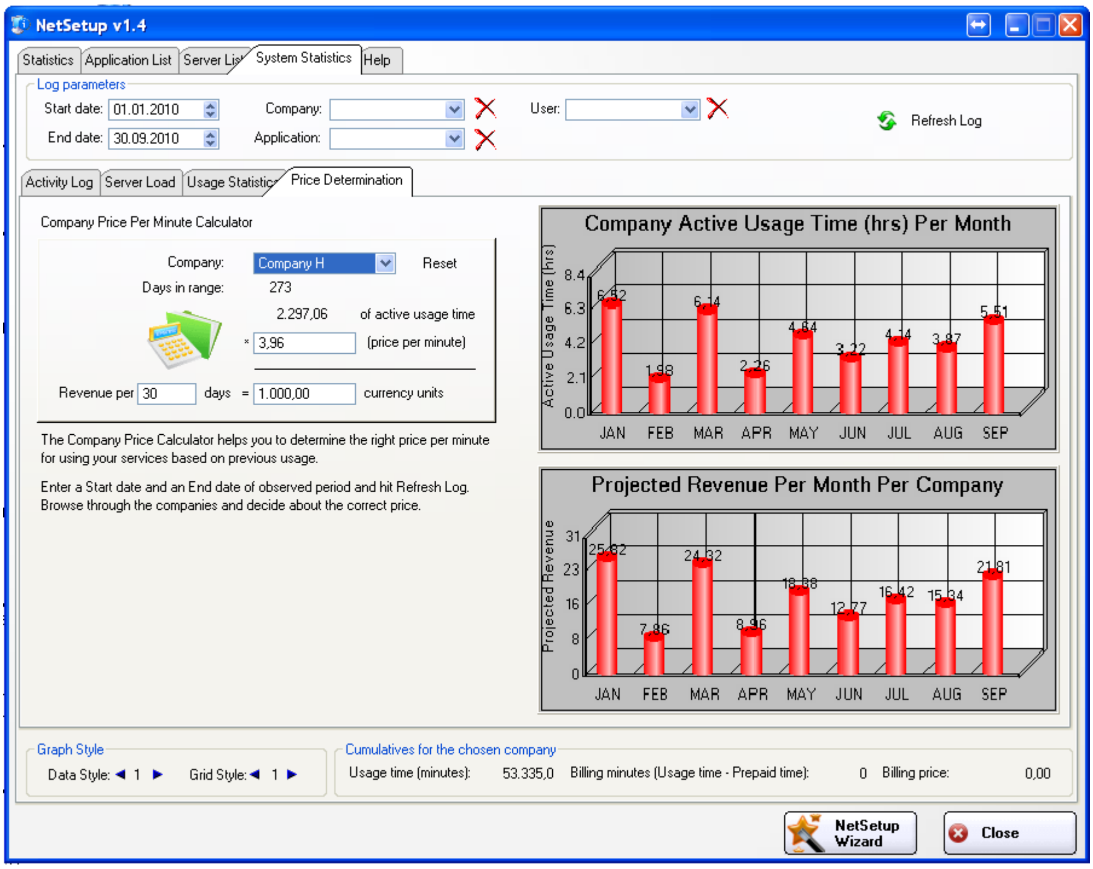
□ Include Graph

☑ Select All | Deselect All

Graph Style
Data Style: ◄ 1 ►
Grid Style: ◄ 1 ►

Cumulatives for the chosen company
Usage time (minutes): 53.335,0
Billing minutes (Usage time - Prepaid time): 0
Billing price: 0,00

There are also tools that help you to determine the optimal Price Per Minute for using your applications.



Log parameters

Start date: 01.01.2010
End date: 30.09.2010
Company:
Application:
User:

Company Price Per Minute Calculator

Company: Company H
Days in range: 273
2.297,06 of active usage time
* 3,96 (price per minute)
Revenue per 30 days = 1.000,00 currency units

The Company Price Calculator helps you to determine the right price per minute for using your services based on previous usage.

Enter a Start date and an End date of observed period and hit Refresh Log. Browse through the companies and decide about the correct price.

Graph Style

Data Style: ◄ 1 ►
Grid Style: ◄ 1 ►

Cumulatives for the chosen company

Usage time (minutes): 53.335,0
Billing minutes (Usage time - Prepaid time): 0
Billing price: 0,00

For more information about implementing and using SaaS in your Thin@ application read the Thin@ Administrator’s Guide.