SIO.NET

Plug & Play Contest System[1]

Marcin Michalski - cyfra@mimuw.edu.pl
Marcin Kosieradzki – m.kosieradzki@students.mimuw.edu.pl
Wojciech Rygielski – rygielski@mimuw.edu.pl
Piotr Stańczyk – stanczyk@mimuw.edu.pl

Krzysztof Ciebiera – ciebie@mimuw.edu.pl
Krzysztof Diks - diks@mimuw.edu.pl

Institute of Informatics, Warsaw University, Poland

[1] This work was partially supported by Microsoft grant.

Abstract

Programming competitions have become very popular over the recent years. There are competitions for secondary school students like IOI, as well as academic contests - like ACM ICPC. There are also open competitions (TopCoder for instance). Many services allowing students to develop their programming skills towards participation in these competitions have been created. Everything indicates that this is one of the fastest evolving ways of teaching young people Computer Science. The main problem is there is yet no software available on the market that would allow a simple user set up his own competition. All available software (even if freeware) is hard to configure, requires comprehensive knowledge of its construction and specialized hardware. An easy to use system for running programming competitions is indispensable. The system that can be used by a teacher to carry out a test for his/her students, the system that can be installed and maintain by anyone who knows the basis of computer usage.

SIO.NET was designed to satisfy these demands - it is an easy-to-use, plug & play contest hosting system that allows anyone to set up and run his own coding competition on any computer.

1 Introduction

SIO.NET is designed for Windows and developed in .NET technology. This software was created for carrying IOI and ACM-like programming contests. The main aim of the project was to make its installation and maintenance as simple as possible. It can be distributed on a single CD; primary installation requires only several mouse clicks and only one computer.

SIO.NET consists of four parts:

- SIO.Admin – an application used by system administrators for managing the contests
- SIO.Web – web page application, makes competition system available to the contestants
- SIO.Worker – windows service responsible for judging contestants programs
- SIO.Database – database of the system

SIO.NET supports any programming language, for which windows 32-bit / .NET compiler exists. It is possible to add additional compilers to the system on the fly (even without interrupting running contest).

System supports different formats of tests to the tasks. Not only they can be provided as normal files, but there can be also test generators and special programs for checking correctness of outputs of contestants' programs.

2 SIO.NET design

Properly installed SIO.NET system consists of several elements. Below is a list of all components that are required to run SIO.NET:

- Prerequirements
  - .NET Framework
  - MSMQ
- Database
  - MSQL
  - SIO.NET Database
- Web-server
  - IIS
  - SIO.NET web-server
- SIO.NET admin
- SIO.NET worker

Depending on the type of contest (number of contestants) SIO.NET is supposed to handle, it is possible to install all components on a single computer (in case of a small competitions), as well as on many computers – in case of required increased system performance. The system was created in such a way, that it is relatively easy to modify its configuration by rearranging components between computers. Below we give a precise description of all the components with described dependencies between them.

2.1 Prerequirements

Prerequirements are the elements that have to be installed on all machines running the system. There are two prerequirenets in case of SIO.NET - .NET Framework and MSMQ.

2.1.1 .NET Framework

SIO.NET was developed in .NET technology, which means that it is required to install .NET Framework upon running the system. If computer on which one tries to run SIO.NET CD does not have .NET Framework preinstalled, CD bootstrap will detect that, and will make it possible to install .NET Framework automatically. .NET Framework is required on all machines hosting any of the components of SIO.NET. It is not required to install .NET Framework within SIO.NET CD – one can download the recent version of framework from Microsoft's website. .NET is built in into the newest version of Windows – Vista, so in the future it will not be needed to install it on the computers.

2.1.2 MSMQ

MSMQ – Microsoft Message Queuing is used by SIO.NET for communication between its components. It has to be installed on all machines. If MSMQ is not installed on the machine, on which one tries to install SIO.NET, installer detects that and installs it automatically. For installing MSMQ it might be required to put into the computer Windows Installation CD.

One of the computers has to be selected as a MSMQ server. It can be any of the machines running SIO.NET - during installation process, while configuring MSMQ, it is required to type the network name of that computer.

2.2 Database

2.2.1 MSQL

SIO.NET uses MICROSOFT SQL SERVER 2000 as its database. SIO.NET is distributed together with redistributable version of this database - MICROSOFT SQL SERVER 2000 DESKTOP ENGINE. Every installation of SIO.NET system requires one instance of MSQL. It can be installed together with any other components of SIO.NET, as well as on a standalone computer.

2.2.2 SIO.NET Database

SIO.NET database is an instance of database running on MSQL database server. It contains information about tasks, rankings, statistics and users of the system. SIO.NET requires this database installed on one computer running MSQL server.

2.3 Web-server

2.3.1 IIS

SIO.NET uses Internet Information Services (IIS) as a web-server. If on the machine, on which SIO.NET web-server is going to be installed, there is no IIS installed, instalator will install it automatically, however it may require using Windows Installation CD.

2.3.2 SIO.NET web-server

SIO.NET web-server can be installed on a computer with IIS installed. Properly configured system consists of at least one instance of web-server. Installation of SIO.NET can consist of one or more web servers to increase its efficiency. Each web server can be installed separately on a computer, as well as with other components of SIO.NET.

2.4 SIO.NET admin

SIO.NET admin is a program that makes it possible for administrators to configure and execute competitions.

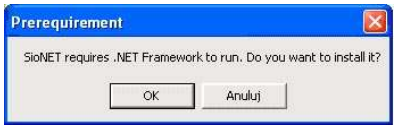
2.5 SIO.NET worker

SIO.NET worker is a program installed on a host machine as a system service and is responsible for judging contestants' solutions to the problems. Each solution is stored in a database and then picked up by an idle worker to be judged. Installation of SIO.NET has to consist at least of one worker, but more computers running workers means that faster contestants' solutions are judged. It is relevant for the just judgments to install all workers on similar computers, so execution time of the same program is similar, no matter which computer handles it. It is possible to install other components of SIO.NET on the same computer together with SIO.NET worker, however it is not recommended as it may have a negative impact on execution time measurements and finally may have some influence on the judgments.

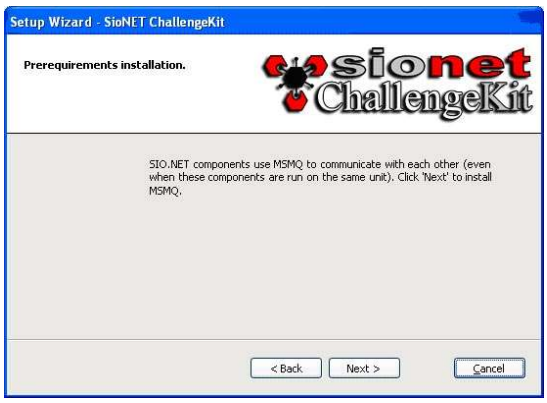
3 Installation of SIO.NET

Installation of SIO.NET is simple – the first step is to put SIO.NET CD into the computer. If autorun feature is turned on, the installator will be executed automatically. If not, *bootstrap.exe* program has to be executed from the root directory of the CD.

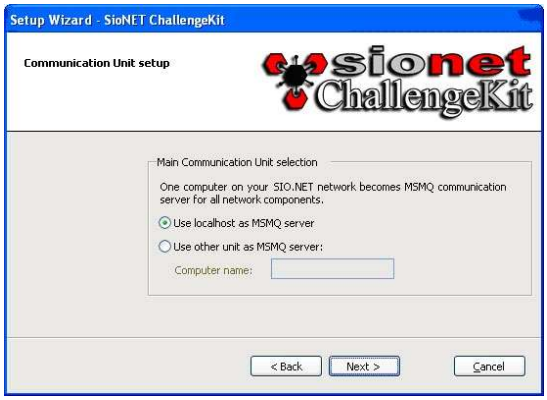
Bootstrap.exe will detect, if .NET Framework is installed on the computer and in case it is not, it will ask user if he wants to install it. After successful installation of .NET Framework, *SioCDLoader.exe* will be executed.



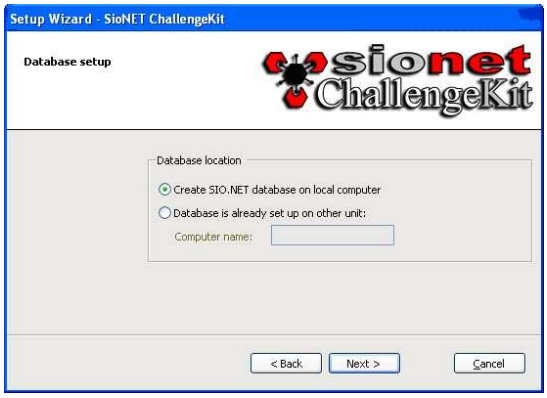
The next step of the installation process is detection of MSMQ. If it is not installed, installator will inform the user that it is required and perform an automatic installation. Installer may ask a user to put Microsoft Windows Installation CD into the computer during installation of MSMQ.



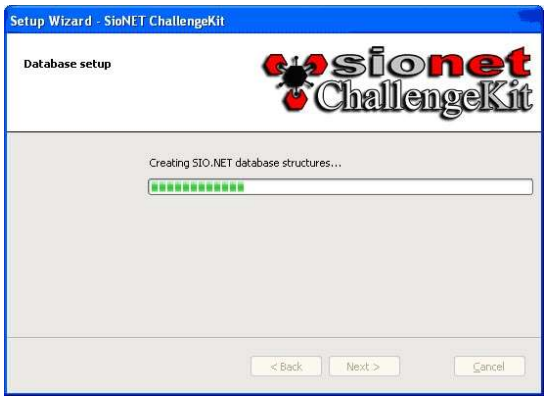
The next step is configuration of the main communication unit (server of MSMQ). In this point there are two options available – using localhost as a MSMQ server, or specifying the network name of the computer hosting it. All computers within a single installation of SIO.NET have to point to the same computer.



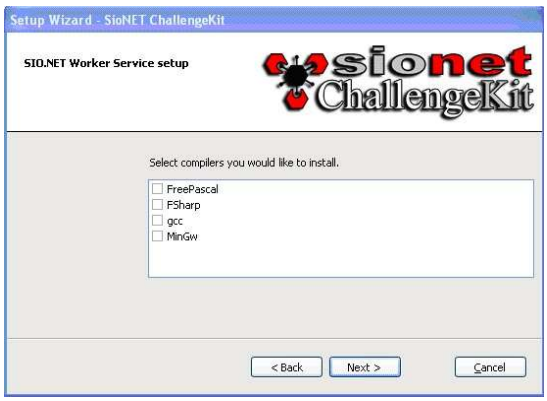
The following step is configuration of data-base. As with MSMQ, there is only one computer with data-base installed. The rest of computers have to be configured to use that database. In case of installing database on local computer, installer will check, if MSQL is installed and if not, it will be installed (this requires specifying a new administrator password for MSQL).



Afterwards, installer will ask a user to specify a new password protecting SIO.NET database and will create SIO.NET database within local instance of MSQL Server.

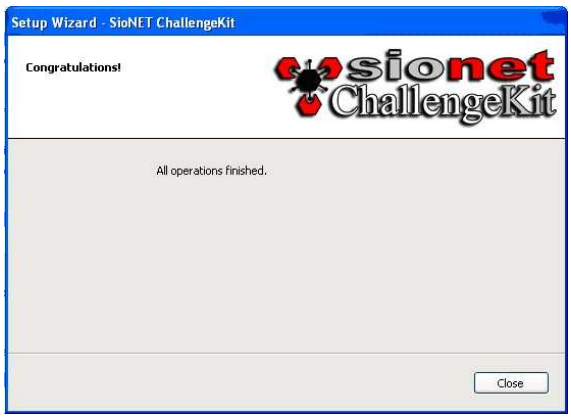


The next step of the installation is selection of programming languages, for which support is required. There are four compilers delivered together with current version of SIO.NET. It is also possible to create additional compilers and install them at any time. After selecting desirable compilers and pressing *Next* button, installer will populate database with selected compilers, making them available to the contestants.



The last step of installation is selecting required components – SIO.NET web-server, SIO.NET worker and SIO.NET administrator panel. The installation of these modules is simple and straightforward.

Installation has been performed successfully. Now it is possible to access SIO.NET competition web-site and use administrator panel to configure contests.



4 Possible system topologies

There are many ways SIO.NET can be installed. Below there are four possible configurations described – starting with simplest one and ending on a huge configuration capable of handling big programming competitions.

4.1 Basic configuration

This configuration consists of only one computer running following components:

- MSQL with SIO.NET Database
- IIS with SIO.NET web-server

- SIO.NET admin
- SIO.NET worker
- MSMQ

This configuration is the simplest possible, however is not recommended due to the fact that SIO.NET worker is installed on the same computer as the rest of components of the system, which may cause some perturbations in execution time measurements of contestants programs.

4.2 Simple configuration

This configuration consists of two computers:

One computer

- MSQL with SIO.NET Database
- IIS with SIO.NET web-server
- SIO.NET admin
- MSMQ

One computer

- SIO.NET worker

4.3 Medium configuration

This configuration consists of four computers:

One computer

- MSQL with SIO.NET Database
- MSMQ

One computer

- IIS with SIO.NET web-server
- SIO.NET admin

Two computers

- SIO.NET worker

4.4 Enterprise configuration

This configuration may consist of nine and more computers depending on the requirements of the contest:

One computer

- MSQL with SIO.NET Database

One computer

- MSMQ

One and more computers (in case of many administrators)

- SIO.NET admin

Two and more computers

- IIS with SIO.NET web-server

Four and more

- SIO.NET worker

5 Programming languages

There are two different programming languages SIO.NET supports - .NET languages and other languages, for which native Windows compilers exist. There are some differences in handling programs for these two types of languages - .NET languages introduce new possibilities for execution time measurement and permission granting to the executed applications, which makes it possible to modify competitions rules by for instance allowing contestants to create multi-threaded programs. In case of programs compiled with native compilers, the situation is similar to the other competition environments.

SIO.NET comes with built in C# and F# (Objective Caml .NET) compilers, as well as FreePascal and MinGw (C/C++) native compilers. It is possible to add additional compilers to the system on the fly (even without interrupting running contest). Releasing a new compiler requires creating an appropriate MSI package with that compiler and adding it to the compilers directory on the SIO.NET CD (just like the others compilers). Afterwards, added compiler can be added to the system using SIO.NET installation wizard.

Creation of MSI package can be achieved by modifying a template delivered together with SIO.NET on the CD in the *template* directory. This step requires Visual Studio 2003 or higher.

6 SIO.NET Engine

This paragraph contains description of some problems we encountered during development of SIO.NET and the way we managed to solve them.

6.1 Workers

Each contest checking system needs a program that will run participants' solutions and check the results. Our system is designed for Windows, so we decided to use all possibilities we can get from it.

We wanted workers to be "hot-plug" that means, it'd be easy to add a new worker during a contest (for example if users are sending so many submissions that current workers are working too slow) and also the second problem was, how to detect that worker is down and will not accept more submissions.

That's why we decided to use Microsoft Queue to communicate between workers and main server. When new worker is created, it connects to the MSMQ server and is ready to check users' submissions.

The next case is security. Previous versions of Windows were known to be “insecure by design”. Currently the Windows XP OS provides us with many security features SIO.NET uses. Workers are executed as local services that mean they are like Linux “daemons” and they cannot be accessed by anyone except administrators. Theoretically workers can be placed on the same computers that are used by contestants, but this is not a suggested solution, as worker during his work consumes a lot of CPU and can slow down the contestant.

6.2 SioSandBox

Each process in Windows XP has its own security parameters. We use them to create a “sandbox” process without any permission for network communication etc. to run contestant’s solution. Thanks to that we have a guarantee that his program will not be able to harm anything.

Windows XP also introduced a lot of built-in system counters. We use two of them: one that checks the maximum amount of memory used, and the second one, that returns the amount of processor cycles used by the process. Thanks to that we know how long the contestant program was running.

6.3 Test Formats

While creating a task there was always a problem with tests. Some tasks required huge input data, in others there could be more than one correct solution. That’s why we decided to create an XML scheme to describe each test. Thanks to that each test can be given as a file, or administrator can provide a test generator and put define appropriate command line parameters. Similar situation is with outputs. Contest creator can decide whether the system should take output from a file, or from a generator, or maybe there is a “checker” program that will decide if user’s output is correct.

6.4 Tests, test groups etc.

The biggest problem was with contest types. There are a lot of different contest types: like IOI and ACM (to mention the most known and popular ones). We wanted to create a system that will be able to handle as many different contest types as possible. That’s why we decided to introduce a new approach to tests and tasks. During a contest each task has a number of test groups. Test group is a set of tests and is accepted if and only if all tests in this set are accepted. This is much needed feature, as there are many tasks where you can simply output “NO” and it’d not be fair if the program that always returns such an output, got non-zero score. Worker is checking the participant submission against all test cases that are in all tests groups, but each test group has two parameters that describe the time when its results will be available for participant and for everyone. So IOI type contest will be a contest where public and private results for all tests groups will be available after the contest, but private results for one (the sample test) test group will be available since the beginning of the contest, while ACM is a contest where there is only one test group and its results are available since the beginning of the contest.