Compiling Computer Programs Through Internet

Aleksander Malinowski and Bogdan M. Wilamowski
Bradley University, Peoria, IL and University of Wyoming, Laramie, WY
olekmali@ieee.org and wilam@ieee.org

Abstract

This paper describes a software package, which allows students compiling programs using web-based interface and Internet connections. Several different computer languages such as C, C++, Fortran, Pascal, and JAVA are implemented. In case of some languages several different compilers can be used and different error messages received. This helps students to track their errors. The access to compiles is controlled using two methods. First the IP address of client machine is verified and then password is requested. Therefore very strict control of licensing is enforced. The Internet Compilers Package (ICP) was developed using an Internet browser as a platform-independent user interface.

1. Introduction

With the increase of Internet bandwidth, the World Wide Web (WWW) could revolutionize design processes by ushering in an area of pay-per-use tools. With this approach, very sophisticated design tools will become accessible for engineers in large and small businesses, and for educational and research processes in academia. Currently, such sophisticated design systems are available only for specialized companies with large financial resources. The pay-per-use approach will have tremendous impact on engineering design since the number of engineers and researchers with an access to sophisticated design tools will increase and this may have a synergetic effect and lead to a significant acceleration of technological development.

The common problem faced by many electronic engineers in industry is that their design tools often work only on one or two among many operating platforms such as UNIX, DOS, Windows 95, Windows NT, Linux, or Macintosh. Another limitation is that software must be licensed for each computer where software is installed. Sometimes, very expensive licenses must be purchased no matter if the software is used very extensively, or very rarely. With the presented approach, only one user interface handled by a network browser would be required.

During the process of software development frequently more than one compiler package is required. When facing obscure error messages, which may result in a time-consuming search for the error, a different error message from another compiler frequently cuts that time dramatically. Therefore students should be to some extend exposed to different compilers at some point in their software courses curriculum.

Although all necessary software is installed in the computer laboratories, most students prefer to work on their computers at home or dormitory and connect to the university network. Web-page based front end allows to access them without any restrictions regarding the computer system requirements thus allowing for their use on different operating system platforms. Furthermore older machines with lesser performance may also be used as terminals. The transaction oriented nature of the hypertext transfer protocol (HTTP) allows for significant bandwidth reduction between server and clients because typically no connection is necessary in the time between receiving results and submitting new batch of data.

Furthermore, instead of purchasing a software license for each computer, compilers can be used on a pay-per-use basis [1]. A similar trend exists in electronic publishing (e.g. IEEE Transactions, books from O'Reilly), where readers are asked to pay per access to the material, instead of purchasing journals, books, or abstracts.

2. Most Commonly Used Network Programming Tools

Although it is possible to develop network applications using C++, or other compiled languages, it is much easier to develop networked applications using dedicated software. There are several well-developed network-programming tools available today [2]. These tools include HTML, JavaScript, VBScript, Java, ActiveX, CGI, and PERL. Another dilemma, is to decide which programming language should be used for which part of the software package. Short characterizations of different network programming tools are given below.

2.1. HTML - Hypertext Markup Language

Hypertext Markup Language (HTML) with associated graphical files is the most commonly used language for network programming. HTML was originally designed to describe a document layout regardless of the displaying device, its size, and other properties. It can be incorporated into networked application front-end development, as an interface layout describing tool and/or wraparound for Java applets. In a way HTML can be classified as a programming language, and the displayed document as a result of the execution of its code. The language is still being developed and every new generation of web browsers can interpret more commands (markers).

2.2. JavaScript

HTML itself lacks even basic programming constructs such as conditional statements or loops. A few scripting interpretive languages were developed to allow for use of programming in HTML. They can be classified as extensions of HTML and are used to manipulate or dynamically create portions of HTML code. One of the most popular among them is JavaScript. The only drawback is that JavaScript is still under development and different web browsers may differ a little in the number of predefined functions and events that they support [3].

2.3. Java

Java is an object oriented programming language compiled to bytecode. It was developed by Sun Microsystems, originally for a different purpose but quickly become popular as an Internet programming tool. All modern network browsers are capable of understanding this bytecode. Java coded programs can be executed either in a secure mode as so called applets, which are usually wrapped by HTML pages, or as applications. In both cases a so-called Java Virtual Machine is necessary to execute the compiled bytecode regardless of a computer platform. The price paid for platform independence is speed. Currently, bytecode is executed approximately ten times slower than compiled language code [4, 5]. The technology of the second step compilation of byte code to processor native code is very promising. However, the compilation time must be quick to avoid execution delays, and most advanced optimization techniques cannot be used, resulting in still slower code.

2.4. ActiveX

Microsoft developed ActiveX as another technology allowing for the automatic transfer of software over the network. ActiveX, however, can be executed presently only on a PC with a Windows operating system, thus making the application platform dependent. Although this technology is very popular already, it does not allow for the development of applications running on multiple platforms.

2.4. CGI - Common Gateway Interface

CGI, which stands for Common Gateway Interface, can be used for the dynamic creation of web pages. Such dynamically created pages are an excellent interface between a user and an application run on the server. CGI programs can be written in any language, but developers usually prefer scripting languages, and PERL is the most popular one. Either scripts, or other CGI-based programs, can handle and process data acquired from a user. These programs or scripts usually control the data flow in a networked application and run other applications that are installed and licensed to run only on the server.

2.5. PERL

PERL is an interpretive language dedicated for text processing. It is primarily used as a very advanced scripting language for batch programming and for text data processing. PERL interpreters have been developed for most of existing computer platforms and operating systems. It is especially suited for CGI network programming [6]. Unlike all the other tools mentioned above, PERL programs are executed only on the application server. It is used for handling data received from users, preprocessing them before running locally installed applications, running these applications, and then processing their results and sending them back to users, usually by wrapping them with HTML.

2.6. CORBA

CORBA (Common Object Request Broker Architecture) is a novel technology developed in the early 90's for network-distributed applications. It is a protocol for handling distributed data, which has to be exchanged among multiple platforms [7]. In addition to providing communication among applications and their data objects (object registration, location and activation; request demultiplexing; framing; error-handling) it also implements automatic translation of their structure in case of different standards used on different machine types. A CORBA server or servers must be installed to access distributed data.

2.7. Cookies

A cookie is a piece of data stored in the client computer. When a request is sent to a server to get an HTML file, some cookies may be transmitted with that request. The server may send different data depending on the information retrieved from the user. Furthermore JavaScript is also capable of browsing through all the cookies stored by the user machine [3]. This information may be used to enhance performance, for example by remembering the user's preferences. This very useful feature, however, is sometimes abused by some Internet providers, who can spy on the user [8] by analyzing what kinds of web pages are being used. Cookies are a technology that is lately considered somewhat controversial, because of the possibility of privacy invasion.

3. Intranet Compilers Package Architecture

During software development it is important to justify which part of the software should run on the client machine and which part should run on the server. Applets are transferred through network when requested and execution is performed entirely on the client machine that made the request. With CGI much less information has to be passed to the server. The server executes instructions based on the given information and sends the results back to the local machine that made the request. Fig. 1 shows the program component division and data flow in a network based application.

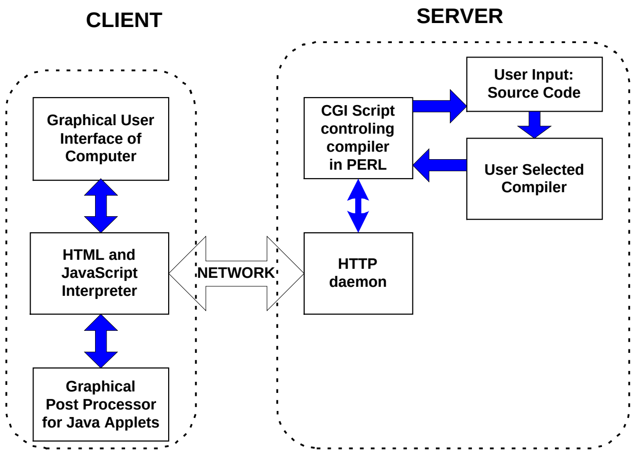

CLIENT
SERVER
Graphical User Interface of Computer
HTML and JavaScript Interpreter
Graphical Post Processor for Java Applets
NETWORK
CGI Script controling compiler in PERL
User Input: Source Code
User Selected Compiler
HTTP daemon

Fig. 1. Data flow in an Internet Compilers package.

Network programming uses distributed resources. Part of the computation is done on the server and another part on the client machine. Certain information must be frequently sent both ways between client and server. It would be nice to follow the JAVA applet concept and have most of the computations done on the client machine. This approach, however, is not feasible for three major reasons:

- Compliers are usually very large and thus not practical for sending entirely via network as applets.
- Software developers are giving away their software without the ability of controlling its usage.
- JAVA applets used on-line and on demand are slower than regular software.

When CGI approach is used then much less information has to be passed to the server. However, large number of clients that access the server simultaneously would make CGI-based approach undesired. It is therefore important to develop methods, which take advantage of networks and then platform independent browsers. This would require solving several issues such as:

- Minimization of the amount of data which must be sent by a network
- Task partitioning between the server and client
- Selection of programming tools used for various tasks
- Development of special user interfaces
- Use of multiple servers distributed around the world and job sharing among them
- Security and account handling
- Portability of software used on servers and clients
- Distributing and installing network packages on several servers

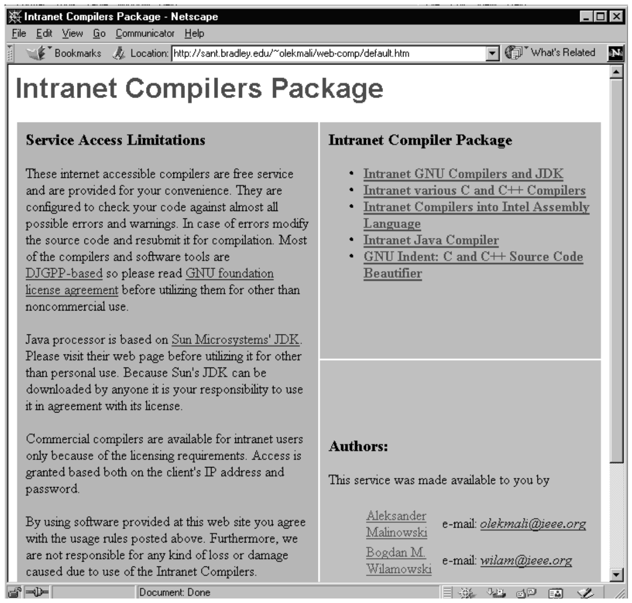
For example, should graphics be generated on the server and sent to a client as a GIF or JPG file, or should only text and binary data be sent to the client and a Java applet used to generate the graphics there. Both approaches have advantages and disadvantages. The data traffic between the server and client in the first case is bi-directional with little data sent as requests from client to server, and much more sent back to the client as images created on the fly after receiving the request. In the latter case, all data is transferred to the client machine together with a Java applet. The job of this applet is to process the user's requests of data visualization without further transmissions from the server. Should the software developer choose the latter case, there would usually be a large overhead in data transfer before data could be displayed.

The server side part of ICP is implemented using CGI scripts written in PERL that handle communication between a user and different compilers. To use ICP, paste the program code from your compiler text editor, or from any text editor, to the web page form. Then submit the form. The compilation will be performed by PERL script on the server in batch mode. That script does the file managing, runs compilers and processes the compilation results [8]. The result is both the source code listing and a binary code to download or a list of errors sent back to the user.

Although the front end is designed to be as simple as possible with only a few commonly used options, it is sufficiently functional and can be used quickly. The PERL script located on the server has to deal with the translation of these common options to the actual options of compilers from different vendors. It also handles the compilation errors and processes the report.

4. Intranet Compilers User Interface

Currently the Intranet Compilers package supports C, C++, Pascal, Fortran, 8086 Assembly and Java. It utilizes GNU, Borland, and Microsoft compilers for C and C++, GNU compiler for Fortran and Pascal, and Sun's JDK for Java. A common front end is used for all compilers is shown in Fig. 2.


Intranet Compilers Package - Netscape
File Edit View Go Communicator Help
Bookmarks Location: http://sant.bradley.edu/~olekmali/web-comp/default.htm
What's Related
Document: Done

Intranet Compilers Package

Service Access Limitations

These internet accessible compilers are free service and are provided for your convenience. They are configured to check your code against almost all possible errors and warnings. In case of errors modify the source code and resubmit it for compilation. Most of the compilers and software tools are DJGPP-based so please read GNU foundation license agreement before utilizing them for other than noncommercial use.

Java processor is based on Sun Microsystems' JDK. Please visit their web page before utilizing it for other than personal use. Because Sun's JDK can be downloaded by anyone it is your responsibility to use it in agreement with its license.

Commercial compilers are available for intranet users only because of the licensing requirements. Access is granted based both on the client's IP address and password.

By using software provided at this web site you agree with the usage rules posted above. Furthermore, we are not responsible for any kind of loss or damage caused due to use of the Intranet Compilers.

Intranet Compiler Package

- Intranet GNU Compilers and JDK
- Intranet various C and C++ Compilers
- Intranet Compilers into Intel Assembly Language
- Intranet Java Compiler
- GNU Indent: C and C++ Source Code Beautifier

Authors:

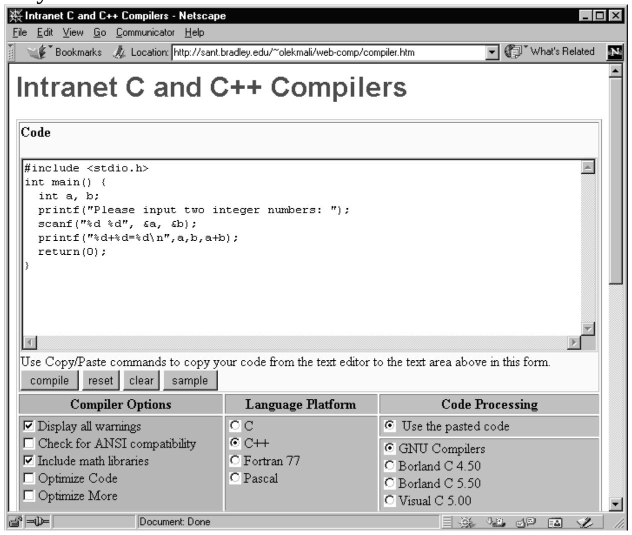
This service was made available to you by

Aleksander Malinowski	e-mail: *olekmali@ieee.org*
Bogdan M. Wilamowski	e-mail: *wilam@ieee.org*

Fig. 2. The common web-based front-end to C++ compilers

This HTML page allows for selecting one of five packages. For example if "Internet various C and C++ compiles" is selected then the front page of Fig. 3 is shown. The code to be complied can be then copy and paste into editing window. Proper compile could be selected and the compilation process starts once the "compile" button is clicked.

The process of compilation is performed in batch mode. As a result another web page with HTML wrapped information with compilation results is sent back to the user as shown in Fig. 4. It is displayed in another browser window so that the user can correct the source code in the original window and resubmit it if necessary. After successful compilation the binary code is available for download. If the user's operating system agrees with the destination platform set for the compiler the binary code may be executed with full user interaction.


Intranet C and C++ Compilers - Netscape
File Edit View Go Communicator Help
Bookmarks Location: http://sant.bradley.edu/~olekmali/web-comp/compiler.htm
What's Related
compile
reset
clear
sample
Document: Done

Intranet C and C++ Compilers

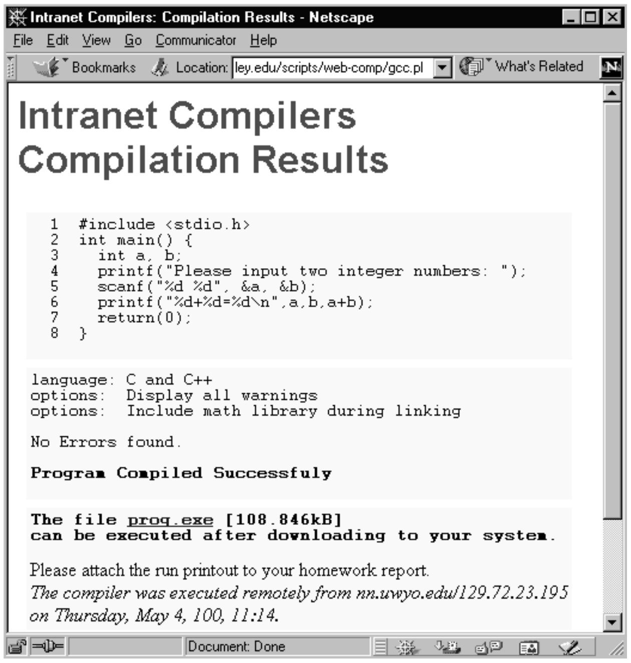
Code

```
#include <stdio.h>
int main() {
  int a, b;
  printf("Please input two integer numbers: ");
  scanf("%d %d", &a, &b);
  printf("%d+%d=%d\n",a,b,a+b);
  return(0);
}
```

Use Copy/Paste commands to copy your code from the text editor to the text area above in this form.

Compiler Options	Language Platform	Code Processing
☑ Display all warnings	○ C	● Use the pasted code
☐ Check for ANSI compatibility	● C++	● GNU Compilers
☑ Include math libraries	○ Fortran 77	○ Borland C 4.50
☐ Optimize Code	○ Pascal	○ Borland C 5.50
☐ Optimize More		○ Visual C 5.00

Fig. 3. Front page for C and C++ compilers


Intranet Compilers: Compilation Results - Netscape
File Edit View Go Communicator Help
Bookmarks Location: ley.edu/scripts/web-comp/gcc.pl
What's Related
Document: Done

Intranet Compilers Compilation Results

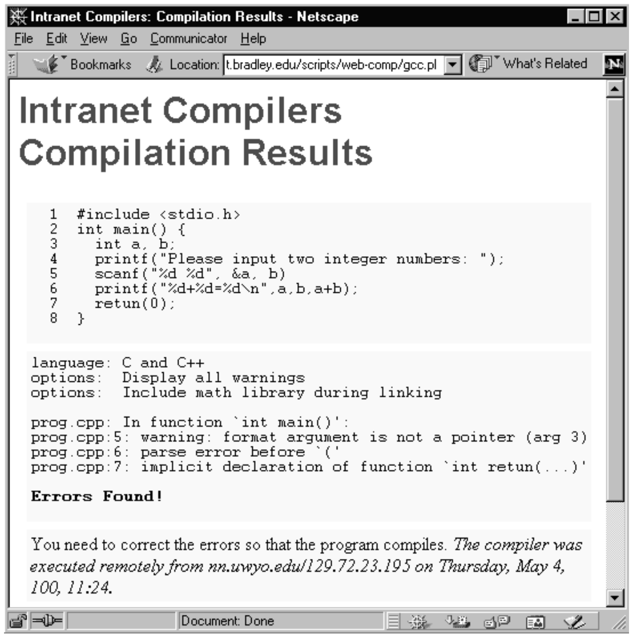
```
#include <stdio.h>
int main() {
  int a, b;
  printf("Please input two integer numbers: ");
  scanf("%d %d", &a, &b);
  printf("%d+%d=%d\n",a,b,a+b);
  return(0);
}
```

```
language:  C and C++
options:   Display all warnings
options:   Include math library during linking

No Errors found.

Program Compiled Successfuly

The file prog.exe [108.846kB]
can be executed after downloading to your system.
```

Please attach the run printout to your homework report.
The compiler was executed remotely from nn.uwyo.edu/129.72.23.195 on Thursday, May 4, 100, 11:14.

Fig. 4. Result page without error obtained using GNU compiler.

When code has errors that warning and error messages will be displayed on the result page. Figures 5 through 7 shows error messages obtained using different compilers such as GNU, Borland, and Microsoft. One of the major advantages of consolidating more than one compiler is the ability to cross-reference error message among different vendor products used from the same interface.



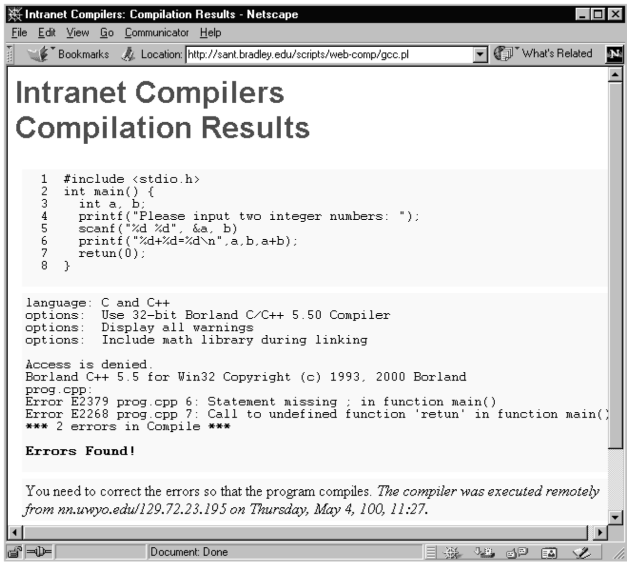
Intranet Compilers
Compilation Results

```
1  #include <stdio.h>
2  int main() {
3    int a, b;
4    printf("Please input two integer numbers: ");
5    scanf("%d %d", &a, b)
6    printf("%d+%d=%d\n",a,b,a+b);
7    retun(0);
8  }
```

```
language: C and C++
options:  Display all warnings
options:  Include math library during linking

prog.cpp: In function `int main()':
prog.cpp:5: warning: format argument is not a pointer (arg 3)
prog.cpp:6: parse error before `('
prog.cpp:7: implicit declaration of function `int retun(...)'

Errors Found!
```

You need to correct the errors so that the program compiles. *The compiler was executed remotely from nn.uwyo.edu/129.72.23.195 on Thursday, May 4, 100, 11:24.*

Fig. 5. Result page with errors obtained using DJ GNU (ver. 2.91) compiler.



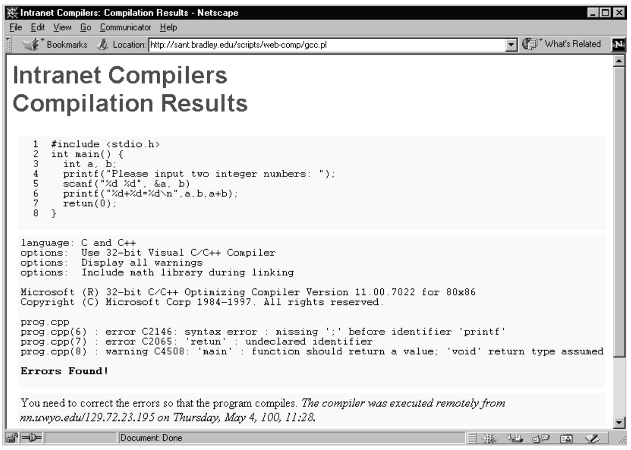
Intranet Compilers
Compilation Results

```
1  #include <stdio.h>
2  int main() {
3    int a, b;
4    printf("Please input two integer numbers: ");
5    scanf("%d %d", &a, b)
6    printf("%d+%d=%d\n",a,b,a+b);
7    retun(0);
8  }
```

```
language: C and C++
options:  Use 32-bit Borland C/C++ 5.50 Compiler
options:  Display all warnings
options:  Include math library during linking

Access is denied.
Borland C++ 5.5 for Win32 Copyright (c) 1993, 2000 Borland
prog.cpp:
Error E2379 prog.cpp 6: Statement missing ; in function main()
Error E2268 prog.cpp 7: Call to undefined function 'retun' in function main(
*** 2 errors in Compile ***

Errors Found!
```

You need to correct the errors so that the program compiles. *The compiler was executed remotely from nn.uwyo.edu/129.72.23.195 on Thursday, May 4, 100, 11:27.*

Fig. 6. Result page with errors obtained using Borland 5.5 compiler.



Intranet Compilers
Compilation Results

```
1  #include <stdio.h>
2  int main() {
3    int a, b;
4    printf("Please input two integer numbers: ");
5    scanf("%d %d", &a, b)
6    printf("%d+%d=%d\n",a,b,a+b);
7    retun(0);
8  }
```

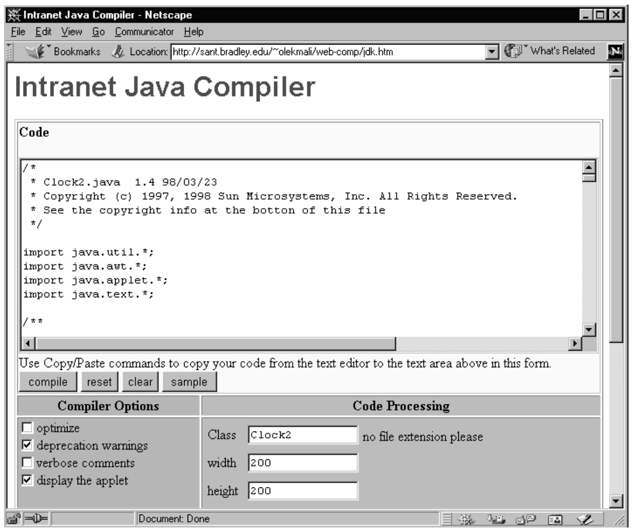
```
language: C and C++
options:  Use 32-bit Visual C/C++ Compiler
options:  Display all warnings
options:  Include math library during linking

Microsoft (R) 32-bit C/C++ Optimizing Compiler Version 11.00.7022 for 80x86
Copyright (C) Microsoft Corp 1984-1997. All rights reserved.

prog.cpp
prog.cpp(6) : error C2146: syntax error : missing ';' before identifier 'printf'
prog.cpp(7) : error C2065: 'retun' : undeclared identifier
prog.cpp(8) : warning C4508: 'main' : function should return a value; 'void' return type assumed

Errors Found!
```

You need to correct the errors so that the program compiles. *The compiler was executed remotely from nn.uwyo.edu/129.72.23.195 on Thursday, May 4, 100, 11:28.*

Fig. 7. Result page with errors obtained using Microsoft Visual C++ 5.0 compiler.

Because of additional requirements for displaying Java applets, an additional dedicated front end is created for that language, which is shown in Fig. 8. If the compilation into bytecode is successful, the user can inspect the applet immediately Fig. 9(b).

Everybody can inspect and utilize the Intranet Compilers Package by visiting http://sant.bradley.edu/web-comp/. When one wants to use ICP over the Internet, only the limited version of the package is accessible. For obvious reasons, everybody can access only the compilers that do not require a license. Commercial compilers (Borland and Microsoft) are accessible only in the Intranet environment for users with selected IP addresses for which the license is bought.



Intranet Java Compiler

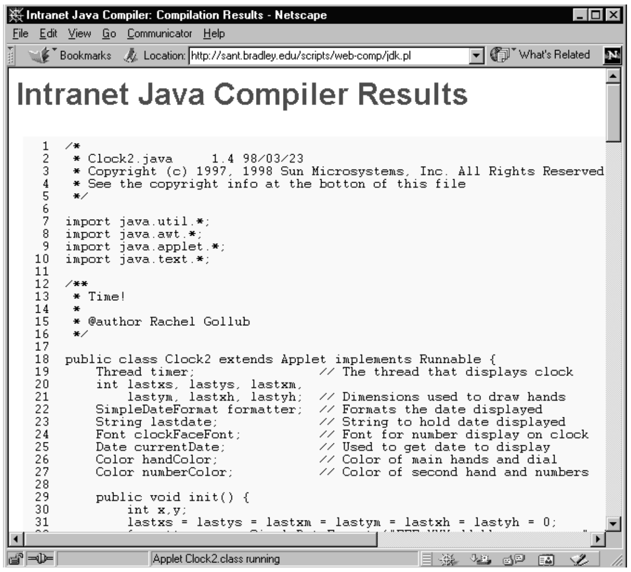
Code

```
/*
 * Clock2.java  1.4 98/03/23
 * Copyright (c) 1997, 1998 Sun Microsystems, Inc. All Rights Reserved.
 * See the copyright info at the botton of this file
 */

import java.util.*;
import java.awt.*;
import java.applet.*;
import java.text.*;

/**
```

Use Copy/Paste commands to copy your code from the text editor to the text area above in this form.

Compiler Options	Code Processing
☐ optimize	Class: Clock2 — no file extension please
☑ deprecation warnings	width: 200
☐ verbose comments	height: 200
☑ display the applet	

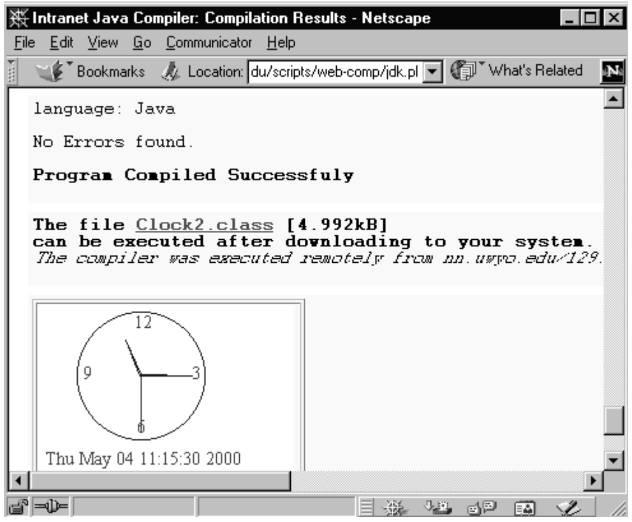
Fig. 8. Specialized front ends for Java.


Intranet Java Compiler: Compilation Results - Netscape
File Edit View Go Communicator Help
Bookmarks Location: http://sant.bradley.edu/scripts/web-comp/jdk.pl
What's Related
Applet Clock2.class running

Intranet Java Compiler Results

```
/*
 * Clock2.java     1.4 98/03/23
 * Copyright (c) 1997, 1998 Sun Microsystems, Inc. All Rights Reserved
 * See the copyright info at the botton of this file
 */

import java.util.*;
import java.awt.*;
import java.applet.*;
import java.text.*;

/**
 * Time!
 *
 * @author Rachel Gollub
 */

public class Clock2 extends Applet implements Runnable {
    Thread timer;                // The thread that displays clock
    int lastxs, lastys, lastxm,
        lastym, lastxh, lastyh;  // Dimensions used to draw hands
    SimpleDateFormat formatter;  // Formats the date displayed
    String lastdate;             // String to hold date displayed
    Font clockFaceFont;          // Font for number display on clock
    Date currentDate;            // Used to get date to display
    Color handColor;             // Color of main hands and dial
    Color numberColor;           // Color of second hand and numbers

    public void init() {
        int x,y;
        lastxs = lastys = lastxm = lastym = lastxh = lastyh = 0;
```

(a)


Intranet Java Compiler: Compilation Results - Netscape
File Edit View Go Communicator Help
Bookmarks Location: du/scripts/web-comp/jdk.pl
What's Related
12
9
3
6

```
language: Java

No Errors found.

Program Compiled Successfuly

The file Clock2.class [4.992kB]
can be executed after downloading to your system.
The compiler was executed remotely from nn.uwyo.edu/129.
```

Thu May 04 11:15:30 2000

(b)

Fig. 9. Result page from successful JAVA compilation (a) upper part of the page and (b) lower part of the page where applet is executed.

5. Conclusion

The paper shows how to use the new opportunity created by Internet technologies for the efficient and platform independent usage of software compilers. The authors are convinced that such an approach is suitable for pay-per-use access and it may revolutionize the general approach toward CAD tool development. Availability of compilers via the Internet may improve our education processes at universities by allowing students to use the same sophisticated software as is used by leading industries. We do hope that major software development companies will soon follow the proposed approach, which they will extend their software offers and sell their software on the pay per use basis.

The possibility to compare error messages from different compilers is very useful especially for students who do not have much experience in coding and debugging yet. The ability to use different compilers allows a programmer to pick up the fastest or the most convenient tool to compile the code and remove the errors. Since August 1988, when the compiler was installed at Bradley University, it was used about 24,000 times, approximately half of that were on-campus students. The package was used by people from more than 30 countries. The up to-date complete usage statistics can be requested any time via Web using the following link: http://sant.bradley.edu/scripts/web-comp/log.pl

6. References

[1] Sweet W., Geppert L., "http:// It has changed everything, especially our engineering thinking", *IEEE Spectrum*, January 1997, pp. 23-37.

[2] Jamsa K., Lalani S., Weakley S., *Web Programming*, Jamsa Press, Las Vegas, NV, 1996.

[3] Flanagan D., *JavaScript, The Definitive Guide*, O'Reilly & Associates, Sebastopol, CA, 1997.

[4] Hank Shiffman, *Making Sense of Java*, http://www.disordered.org/Java-QA.html

[5] Hank Shiffman, *Boosting Java Performance: Native Code and JIT Compilers*, http://www.disordered.org/Java-JIT.html

[6] Mall L., T. Christiansen, and R. L. Schwartz, *Programming PERL*, O'Reilly & Associates, 2nd edition 1996.

[7] Object management Group, *The Common Object Request Broker: Architecture and Specification, v. 2.2*, published by Object Management Group, February 1998. see also http://www.cobra.org/

[8] Wall,L., Christiansen, T., Schwartz, R.L. *Programming Perl*, OReilly & Associates, Inc., 1996