

---

**Blue Diamond**  
**An Internet Transaction Server for Progress based E4GL/4GL programs**

**Blue Diamond Studio**  
**Web based application development**

Written by Scott Auge

sauge@amduus.com

---

## Table of Contents:

What is Blue Diamond? .....	5
System Requirements: .....	5
About Performance: .....	5
About Licensing: .....	6
Blue Diamond License .....	6
Progress License Issues .....	7
Price Comparisons to Webspeed .....	8
Feature Comparisons to Webspeed .....	9
The Blue Diamond Architecture: .....	10
How To Install Blue Diamond?.....	12
Creating the BD database: .....	12
bd.st .....	12
mkdb.ksh .....	12
setenv.ksh .....	12
bd.df.....	13
Administrating the DB: .....	13
bd.pf.....	13
startdb.ksh.....	13
logindb.ksh .....	13
stopdb.ksh.....	13
How to configuring the web server? .....	13
How To Configure Blue Diamond? .....	14
The Blue Diamond runner .....	14
Using the Blue Diamond Studio.....	16
Studio Home Page .....	17
Studio File Operations.....	18
Studio Source Lab .....	19
Studio Compile Files .....	21
Studio Edit File.....	23
Viewing a file .....	24
Data Dictionary .....	25
Tables Available.....	25
Fields and Indices Available .....	27
Compiling the entire project with Studio: .....	27
Explanation of the commands .....	29
Source Clean Up Script .....	29

---

The E4GL to 4GL converter.....	30
The E4GL to 4GL Directory Converter .....	30
The 4GL Command Line Compiler .....	30
The 4GL Directory Compiler .....	30
The Build 4GL command.....	31
The Directory wide Build 4GL command.....	31
The Single File Xref command .....	31
The Directory wide file Xref command .....	31
How to make a program with Blue Diamond? .....	32
How do I convert my current Webspeed E4GL programs to Blue Diamond? .....	39
How do I convert my Blue Diamond programs to Webspeed? .....	40
How can I change the content-type in a Blue Diamond program? .....	40
Supported WSMETA tags.....	40
Blue Diamond Routines .....	41
Blue Diamond HTML and support Routines .....	41
IntToHex ().....	41
URL-ENCODE().....	41
HexToInt().....	41
URL-DECODE().....	42
SetContent .....	42
GET-FIELD() .....	42
GET-CGI() .....	42
GET-COOKIE() .....	42
GET-VALUE().....	42
SET-COOKIE().....	42
DELETE-COOKIE().....	42
HTML-ENCODE() .....	42
IntToHex().....	43
URL-ENCODE().....	43
HexToInt ().....	43
URL-DECODE().....	43
Application Configuration Routines.....	43
DeleteParm.p .....	43
ReadParm.p .....	43
SetConf.p .....	43
WriteParm.p .....	44
inird.p .....	44
Application User Session Saving Routines .....	44
ClearState.p .....	44
ReadState.p.....	44
WriteState.p.....	44
Miscellaneous Routines.....	44

---

---

RandomString.p.....	45
MakeID3() .....	45
Error.i .....	45
PostFix() .....	45
SetPostFix() .....	45
BaseName() .....	45
FilePath() .....	46
FilePath() .....	46
Application User Accounting Routines.....	46
LoginUser.p.....	46
LogoutUser.p.....	46
Handling Mail With The Application.....	46
smtpmail.p .....	46
PROCEDURE ConnectToServer: .....	47
PROCEDURE DisconnectFromServer: .....	47
PROCEDURE Login:.....	47
PROCEDURE ListMessages: .....	47
PROCEDURE RetrieveMessage:.....	47
PROCEDURE DeleteMessage:.....	47
PROCEDURE Logout:.....	47
Application Security Routines.....	48
Loggedin.i.....	48
Database Tables For Blue Diamond.....	49
Appendix A: General flow of steps to installing Blue Diamond.....	56
Appendix B: Converting your Webspeed E4GL program to work with Blue Diamond.....	57
Appendix C: Creating an E4GL program for Webspeed or Blue Diamond.....	58
Appendix D: Problem Resolution Strategies.....	59
Appendix E: Change Log .....	60
Appendix F: Using Blue Diamond on Windows.....	65
What does support include?.....	66

---

## **What is Blue Diamond?**

Blue Diamond allows programmers familiar with Progress Software Corporation's E4GL programming language and 4GL language to extend their applications to the web interface without the use of Webspeed Transaction Server or Webspeed Workshop.

VARs that use E4GL to construct their applications for Webspeed can modify those programs in a very simple manner to run it on Blue Diamond.

The name Blue Diamond came from the name of a road in Las Vegas. As I traveled from the Silicon Valley area into Las Vegas, seeing Blue Diamond meant that I was near the entrance to Las Vegas Blvd. and I should keep an eye out! (The funny thing of it is, I do not gamble or drink or smoke – but Vegas does have cheap hotel rooms and a change of environment!)

Blue Diamond uses a build number to identify different versions. The build number is part of the name of the distribution file:

```
BlueDiamond.2002.035.20.53.zip  
      yyyy ddd hh mm
```

where yyyy is the year, ddd is the julian date, hh is the hour, and mm is the minute.

## **System Requirements:**

- Progress V8 or better CHUI with the 4GL Development System or Webspeed Workshop (does not need Provision)
- UNIX
- Currently Windows is NOT supported

## **About Performance:**

Here are some results of a small benchmark (I plan on using it for my Service Express program and didn't want to end up in performance trouble.)

Machine: VA Linux 420 Workstation, 128 MB RAM , Pentium III 833MHz, IDE Harddrive (albeit a really fast one).

---

OS: RedHat Linux, Progress V9.1A (5 license), Apache Webserver

Program: Creates a record of 6KB, looks up a record of 6KB (not the created record), deletes a record of 6KB (not the created record), decent size of HTML code sent out (6+KB). I figure working with 6KB pretty much might be the amount of data a program might work with concerning the database on a usual page hit, all together the program manipulates 24 KB of data (there is a variable holding 6K for the CREATE of a record.)

The test is a page divided up into six frames, with each frame sourced to the above program. Hence I can prove pages per second: 6 (It probably is much faster, but the test was good enough for my purpose.)

Hence 30 pages per second on 5 licenses.

1800 pages per minute

108,000 pages per hour

2,592,000 pages per day

I would think a larger machine with more licenses could handle more.

### **About Licensing:**

#### ***Blue Diamond License***

Blue Diamond is open source under the Berkley License. The license reads as follows:

Written by Scott Auge [scott\\_auge@yahoo.com](mailto:scott_auge@yahoo.com) [sauge@amduus.com](mailto:sauge@amduus.com)  
Copyright (c) 2002 Amduus Information Works, Inc.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:  
This product includes software developed by Amduus Information Works Inc. and its contributors.
4. Neither the name of Amduus nor the names of its contributors may be used to endorse or promote products derived from this

---

software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY AMDUUS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AMDUUS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A Blue Diamond Programming Guide is available from Amduus Information Works, Inc. for learning how to program with Blue Diamond.

### ***Progress License Issues***

The tool depends on Progress ([www.progress.com](http://www.progress.com)) licensing. Progress licensing is NOT FREE.

The type of license that Progress requires for use of Blue Diamond is per concurrent user. The licenses you will need are:

- Progress RDBMS (Enterprise or Workgroup) OR Progress DataServer
- 4GL Development License (on the development machine, not the deployment machine)
- Client Networking (if your database is not on the same machine as the clients.)
- Character Interface clients for your operating system.

Amduus Information Works, Inc. is a Progress ISV. We can sell you Progress licenses to run Blue Diamond on. We can sell you licenses for a stand alone Blue Diamond application and the licenses for a Blue Diamond based integration application to an existing Progress application.

If you wish to integrate your Blue Diamond application to an existing Progress application, you will need to check if your existing Progress application has enough licenses for the additional number of concurrent Blue Diamond users.

Contact [sauge@amduus.com](mailto:sauge@amduus.com) for more information<sup>1</sup>.

---

<sup>1</sup> Support at a fee, as well as programming is available from Amduus Information Works, Inc.

---

## Price Comparisons to Webspeed

Prices are Amduus Information Works, Inc. licensing offers.

Blue Diamond		Webspeed	
5 User Workgroup	\$1,625.00	25 User Workgroup	\$8,125.00
Blue Diamond Trx Server	-0-	25 Agent Transaction Server	\$13,000.00
Minimum Offering	\$1,625.00	Minimum Offering	\$21,125.00
Remote DB Support	\$910.00	Included	-0-
Minimum Offering	\$2,535	Minimum Offering	\$21,125.00
Blue Diamond Support	\$500.00 per month	Depends on level of support	
Minimum Offering	\$3,035.00	Minimum Offering	> \$21,125.00
Blue Diamond Studio <sup>2</sup>	-0-	Workshop per developer	\$1,235
Minimum Offering	\$3,035.00	Minimum Offering	> \$22,360.00

The minimum break even point for Blue Diamond to Webspeed is 65 users. On a machine requiring remote database access, it would be 41 users.

Remember, Blue Diamond is Concurrent User licensing.

Webspeed uses agent licensing – which means you can squeeze as many users on it as the machine will bear.

Other considerations:

Deploying Blue Diamond on multiple machines is free (minus the described needed Progress licenses for those machines.) Webspeed requires additional licenses for these machines.

---

<sup>2</sup> You will need either the Webspeed Workshop (\$1,235.00) or 4GL Development System (\$3,640) installed to create .r's. Studio will create .p's for execution, but you will need the Developer Tools Kit to create encrypted code that will run on run-time versions of Progress.



---

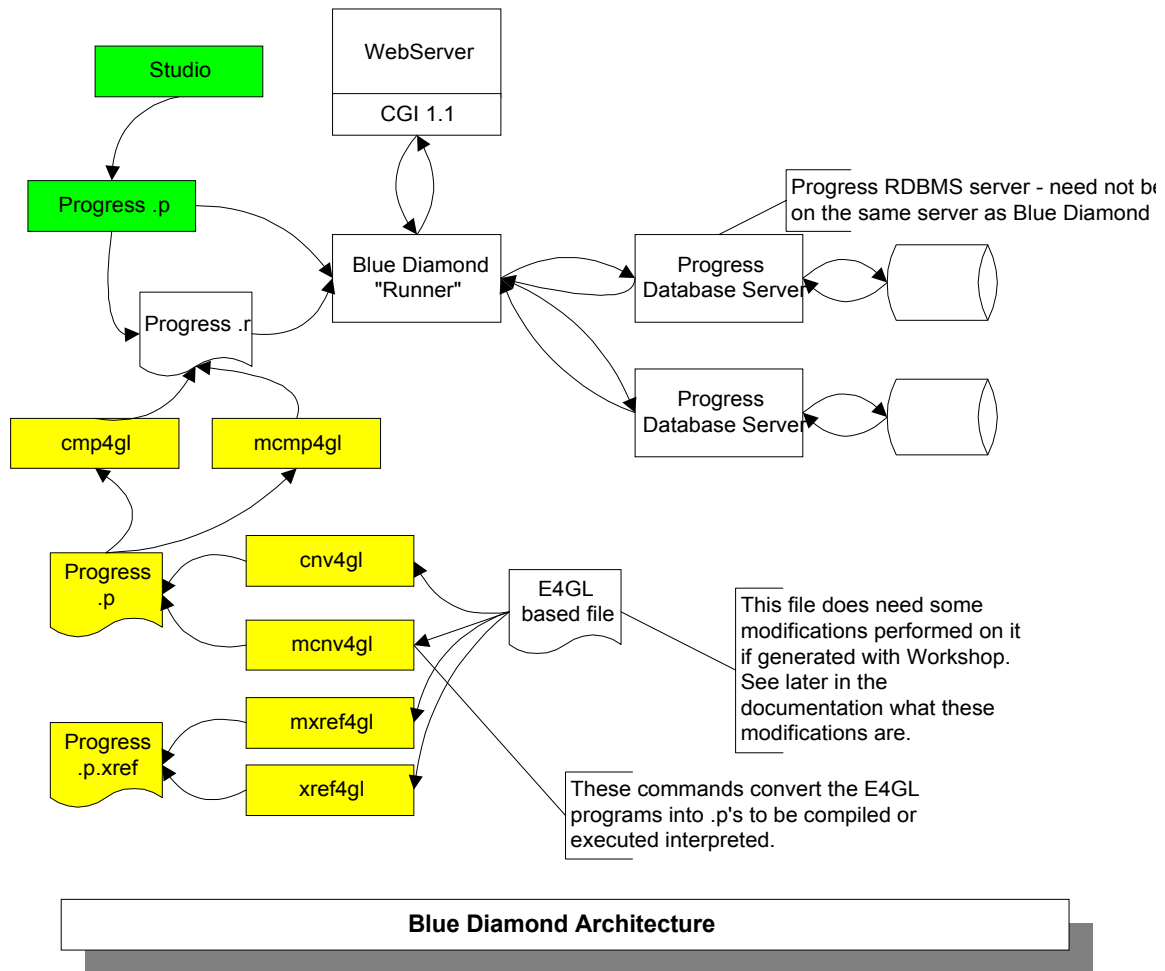
### Feature Comparisons to Webspeed

Feature	Blue Diamond	Webspeed
Windows Support	Partial	X
UNIX Support	Nearly all UNIX available for Progress	Limited UNIXs
Progress 4GL	X	X
SpeedScript	X	X
HTML-Mapping		X
Two Tiers (remote databases)	X	X
Three Tiers (messenger/broker/database)		X
Web Based Development Tools	X	X
Command Line Based Development Tools	X	
Out of Box Mail routines	X	
Out of Box Session routines	X	
Out of Box App Conf routines	X	
Out of Box User Management routines	X	
Open Source	X	X
“Project” Compiling	X	

---

## The Blue Diamond Architecture:

The items in Green are the Studio web interfaces to Blue Diamond. The items in Yellow provide command line interfaces to Blue Diamond.



Blue Diamond uses off the shelf Progress Database Servers. For scaling, the database servers can be on different machines than Blue Diamond.

Blue Diamond requires Progress character clients that operate on UNIX and Windows.

For Windows support, perl is also required to be installed.

---

Blue Diamond must be installed on the web server. Scaling can be achieved by having multiple web servers handled by a product like Cisco's LocalDirector found here <http://www.cisco.com/warp/public/cc/pd/cxsr/400/index.shtml>

The system using Blue Diamond follows these steps:

- A browser connects to the web server.

*The web server handles SSL connections; so encrypted sessions can be achieved with Blue Diamond by virtue of the web server supporting this.*

- The web server passes the message to Blue Diamond through the CGI interface.
- The runner script executes a progress character.

*CGI Myth Control: Under UNIX, the \_progres file and script file can have it's sticky bit set. Doing so eliminates the need to have the program loaded from disk, making the CGI interface efficient. All that is required is a "fork" impeded by speed of the processor and memory available; as well a connection to the DB – a network operation impeded only by the speed of the network (a slow speed will effect DB manipulation as well as connection.)*

*The sticky bit is set with chmod I### where ### represents the rwx values of the other, user, and group permissions in octal. It can also be accomplished with chmod +t [filename] – you will usually need to be root to do this.*

- The runner program is executed which determines what program should be run and executes the .p or .r according to Progress' rules.
- The program is executed, outputting the results to the CGI interface of the web server
- The web server then passes the results back to the browser.

---

## How To Install Blue Diamond?

You will be presented with a compressed tar file entitled `bd.2.2.tar.Z`.

1. Create a directory (a directory called `BlueDiamond` will be created in this directory)
2. Uncompress the tar with `uncompress bd.2.2.tar.Z`
3. Un-archive the tar with `tar -vxf bd.2.2.tar`
4. Copy the `script/bluediamond` script into your web server's CGI directory. Edit the script to connect with your databases, as well to use the `PROPATH` of the application it should work against.
5. Proceed to configuring Blue Diamond

### Creating the BD database:

The Blue Diamond database stores parameter, session, etc. information within it. Since this database may be altered in future versions of Blue Diamond, you should construct a separate database for it.

**Note: If you do not plan on using the Parameters and Session tools provided with Blue Diamond, you do not need to set up this database. If you plan on loading the Blue Diamond tables into your own database, you do not need another one.**

Once the database is created, one should load in the `bd.df` file.

There are scripts available to ease in the construction of the blue diamond database.

#### ***bd.st***

This is the configuration for the blue diamond database. Modify it as you see fit, but the default version should work for most instances.

#### ***mkdb.ksh***

This script will create the DB. Before running this script, you must edit the `setenv.ksh` script with the correct settings.

#### ***setenv.ksh***

---

This script contains such information as to where to find progress, what the database directory is, and where to find the PF file for the database.

### ***bd.df***

Once the database is created and you can login, you should use the database dictionary tool to load this into the schema area.

### **Administrating the DB:**

Some scripts are available in the bd/ directory for simple administration of the database.

### ***bd.pf***

This pf file is given as a template for your bd database.

### ***startdb.ksh***

This script will start the database.

### ***logindb.ksh***

This script will log into the database using a character client.

### ***stopdb.ksh***

This script will stop the database.

### **How to configuring the web server?**

You will need to configure a script directory for your web server. The following documents setting up the Apache Web Server to call Blue Diamond.

In the httpd.conf file, you will find a section as follows:

```
# ScriptAlias: This controls which directories contain server scripts.
# ScriptAliases are essentially the same as Aliases, except that
# documents in the realname directory are treated as applications and
# run by the server when requested rather than as documents sent to the client.
# The same rules about trailing "/" apply to ScriptAlias directives as to
# Alias.
```

---

Under this section, you would enter your SCRIPT\_NAME for the URL, as well the physical directory it would fall under. Then you would use the DIRECTORY options phrase to describe the kinds of accesses that can be performed on it. See your Web Server documentation guide for more information.

```
ScriptAlias /online/ "/appl/cgi/"

<Directory "/appl/cgi">
    Options ExecCgi
    AllowOverride AuthConfig FileInfo Indexes Limit Options
    Order allow,deny
    Allow from all
</Directory>
```

This example tells the server, that when ever /online is referenced in the URL, it actually means a file or subdirectory/file within the /appl/cgi directory.

It is in the /appl/cgi directory that one would place a copy of the /script/bluediamond script (renamed and tailored to the needs of your application).

***You should NOT use the Blue Diamond /script directory as a CGI script directory, as your scripts can be called by anyone with access to the web server via a web browser.***

Replace such entries with one that is appropriate for your installation.

### **How To Configure Blue Diamond?**

Currently Blue Diamond has multiple scripts that will need modification to operate on your system. You will need the following information:

- Where the Progress Character Clients can be found.
- Where the source code to Blue Diamond can be found.
- Where the source code to the application to run on Blue Diamond can be found.
- The name of an environment.

### ***The Blue Diamond runner***

The Blue Diamond runner is called by the web server by it's placement in the server's CGI or "script" directory. It's purpose is to call runner.p, a program within Blue Diamond that identifies the desired program to run based on the URL presented to the web server from the browser.

Below is the script for a Linux set-up.

```
#!/bin/bash
```

---

```
# $Header: /home/appl/BlueDiamond/script/RCS/bluediamond,v 1.2 2002/04/20
14:32:20 sauge Exp sauge $

# This is the script you will use in your Web Server CGI directory.
# Copy and rename as you need.
# Edit the following line to point to the correct environmental settings
# for your application. This should be the location and environmental
# name you used when creating a blue diamond application environment.
# Or, simply set these variables:

# Where to find the PF file

export PFFILE=

# This is the propath for your application. It is used in the compile all
# option in the Studio. It is separated out from the PROPATH because Progress
# tags in additional directories for Progress only routines we don't want
# Blue Diamond to be fiddling with!

export APPL_PROPATH=

# Where to find Blue Diamond. This is so you can easily switch between dif-
# ferent installs of BD.

export BDPROPATH=/appl/BlueDiamond/prgsrc

# Where to find Progress

export DLC=/usr/dlc

export PROPATH=$BDPROPATH:$PROPATH:$APPL_PROPATH

# Used for controlling what directories Studio can traverse on the system.
# By leaving blank, the Studio can traverse what ever directories it can
# get into. By adding directories, the Studio will be able to work in those
# directories and their sub-directories only.

export STUDIOPATH=

# You can give the Studio a working directory here. When studio first starts
# up, it will default to this directory.

export WORKDIR=

# ----- Leave below alone unless you know what you are doing -----

export TERM=vt100

# We need someplace to tuck away data that is POSTed, so we place it in this
# environmental variable. See sga_util.i for how it is used.
```

---

---

```
export POSTDATA=$(cat -)

# Don't forget to set the sticky bit on this file, it will stay in
# memory instead of being pulled off disk for each hit.

$DLC/bin/_progres -b -p runner.p -d mdy -pf $PFFILE
```

The DLC directory identifies where the Progress RDBMS and client software can be found.

The PROPATH is used to find the byte code or source to interpret that Blue Diamond is to run. Since part of Blue Diamond is written in Progress' 4GL, it is important to include the path to the Blue Diamond installation.

**Note: If you wish to run Blue Diamond Studio, you will need the BlueDiamond/studio directory included in your propath. If you wish to merely run only your application without Studio, do not include this directory.**

**Note: Blue Diamond requires the BlueDiamond/prgsrc directory is present in the PROPATH in order to run the transaction server against your application.**

Additional paths are required for your Blue Diamond based application, and your developers or vendor should provide this information to you.

A "pf" file is required to be attached to the call to the Progress client. Inside this file are connection parameters to Progress based databases. These parameters can also be placed in the script, but it can be more convenient to place them in one place and have all possible scripts reference them.

### Using the Blue Diamond Studio

Blue Diamond Studio provides the user with a web interface to the commands that would normally be done via a command line. It provides the following functionality:

- Traversing directories on the machine
- Compiling selected files<sup>3</sup>
- Viewing a selected file
- Editing a selected file

---

<sup>3</sup> In order to convert the files into a .r, you will need an installation that accepts COMPILE filename SAVE. The 4GL Development System license or the Workshop license will allow this. Else you will have to settle for .p's with the Developer's ToolKit to encrypt them for execution on run-time licenses.



- 
- On the fly execution of programs

### ***Studio Home Page***

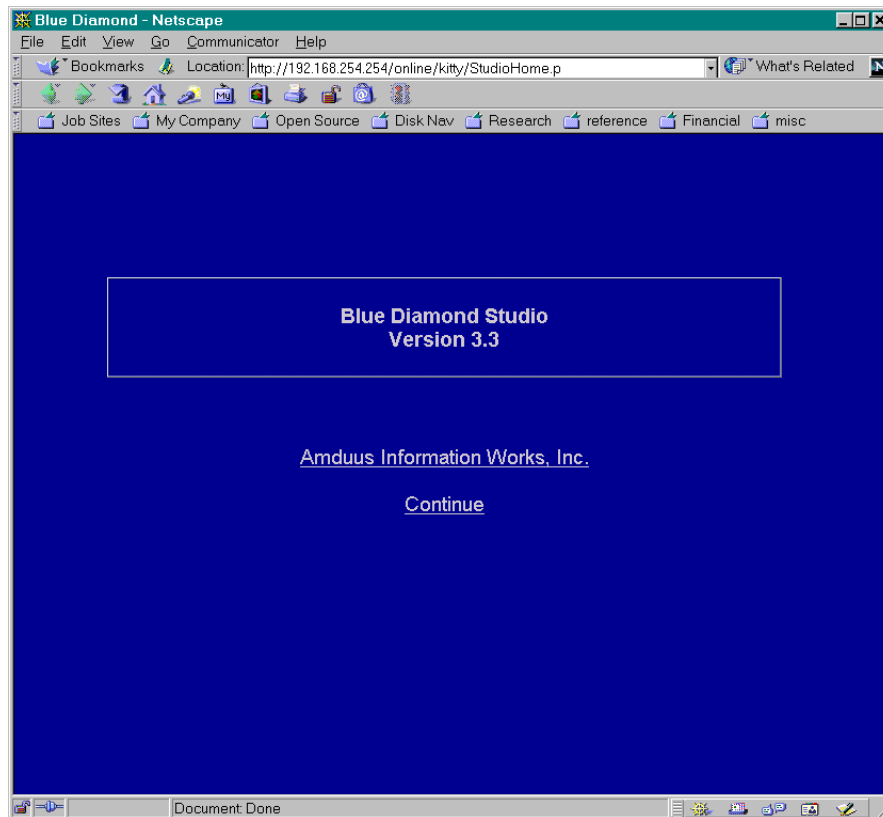
The studio home page is the entry point to reach the Blue Diamond Studio web application. This web application is actually a Blue Diamond program, so it should give you an idea of what can be accomplished with Blue Diamond.

In order for your browser to reach Studio, you will need to include the *install/BlueDiamond/studio* directory in your CGI script under the BDPROPATH variable.

If you do not wish your web users to access Studio, do NOT include the directory in the BDPROPATH of the CGI script.

To access the Studio via your browser, it would be :

<http://youmachine/yourcgidir/yourcgiscript/StudioHome.html>



*Home Page of Blue Diamond Studio*

---

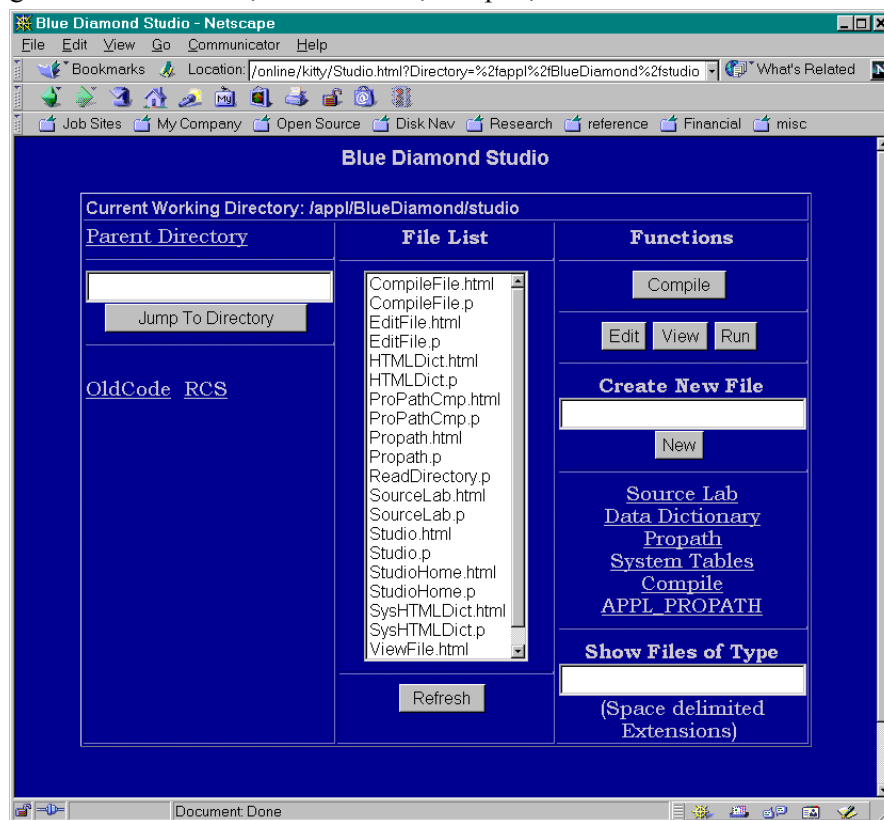
*Blue Diamond studio does not require the Blue Diamond database to be installed, unless you are using the tables in that database.*

### ***Studio File Operations***

The file operations page is where the developer and administrator will spend their time. It is not meant to be used by the end user, as it allows them to modify source code files.

One can navigate through the directory structure on the machine by the hyperlinks on the left side of the page. The files available in the current working directory (noted at the top) will be presented in a list box.

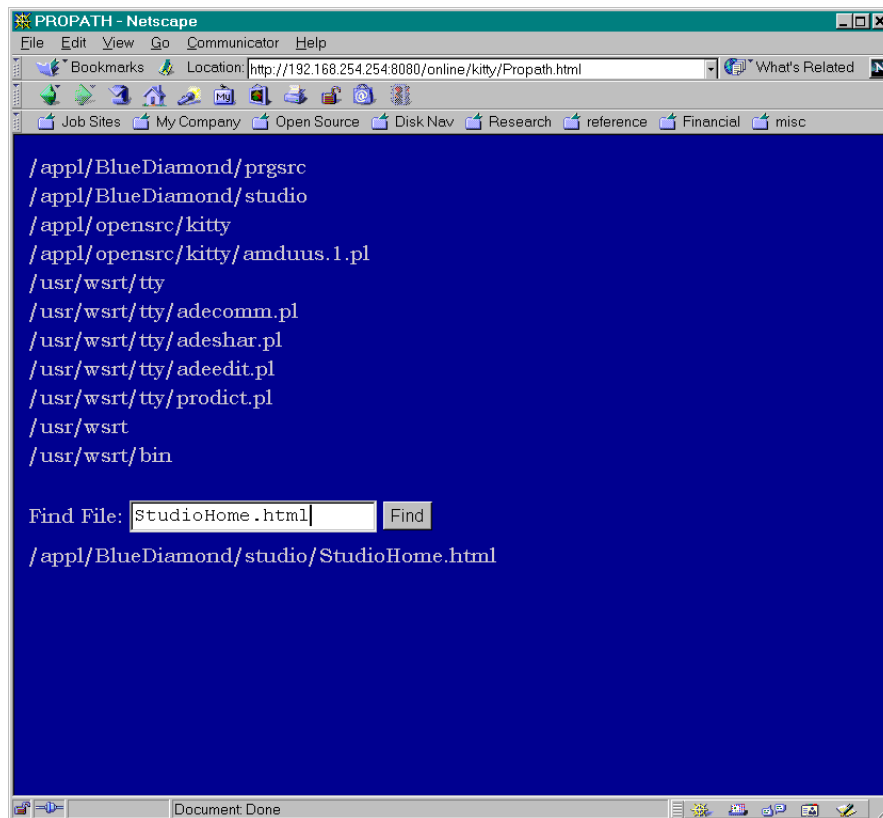
By choosing files in the list box, one can view, compile, run and edit the files.



*File operations screen of Blue Diamond*

Using the Jump To Directory, one can hop immediately to a directory (security of STUDIOPATH may prevent jumping to a certain directory.) One can hit the parent directory merely by clicking the Parent Directory hyperlink.

An additional hyperlink to the Source Lab tool is available under the box with the file operation buttons. Also included is a link to the ProPath screen that shows the current working PROPATH as well finding files under the PROPATH.



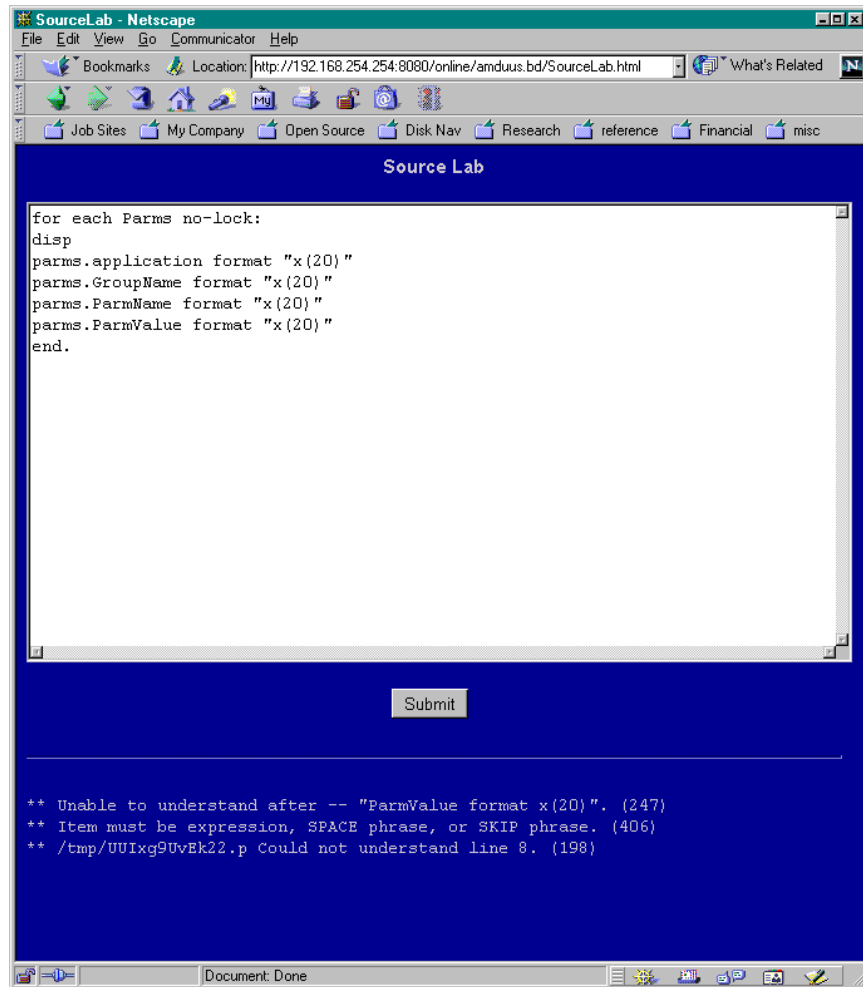
*Propath and file location in Blue Diamond Studio*

New files can be created with the blank and New button below the file list. This file will be created in the current working directory as noted at top of the screen.

### ***Studio Source Lab***

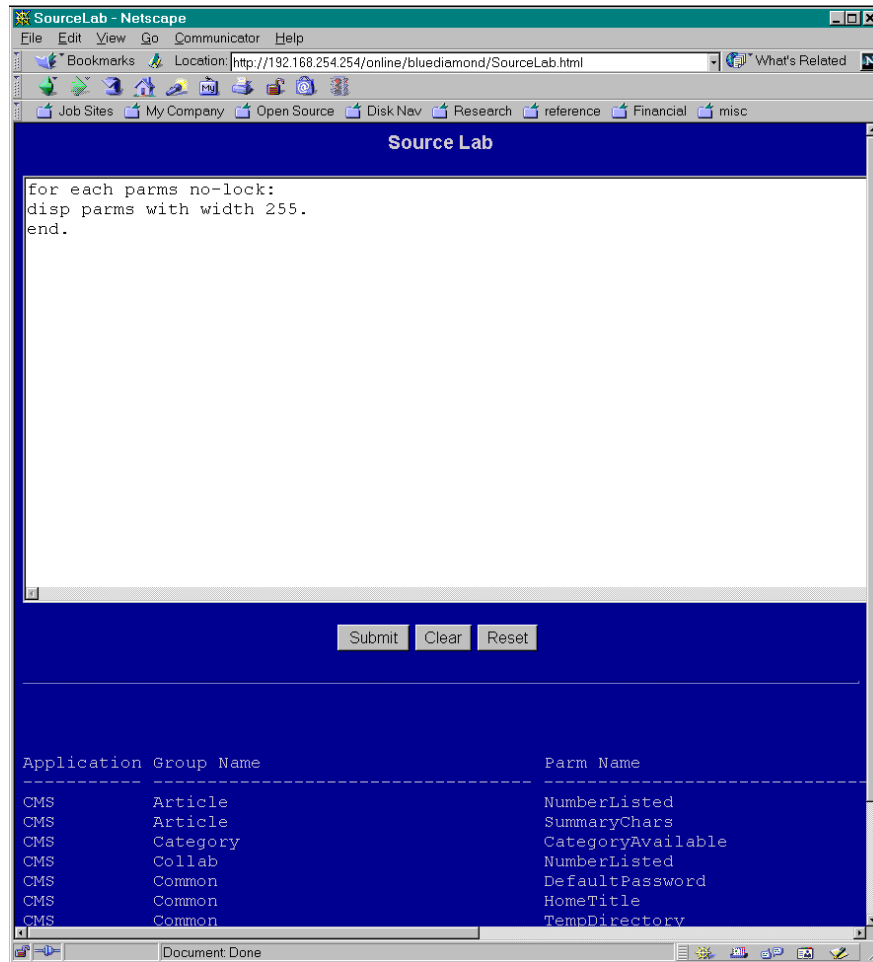
At times you will find yourself needing to perform some queries on the database or to execute some 4GL code. This can be done with the Source Lab tool accompanying Blue Diamond.

To use the tool, one merely points their web browser at <http://hostname/script/SourceLab.html>. One will be presented with a box to store the program into. Upon pressing the Submit button, the program is executed, and any output from the program is displayed below the input box. Previous programs are placed into the input box for modification or the like.



*A progress compile error shown in the web window*

If errors present themselves in your program, they will usually be displayed on the web page. Under certain circumstances, the user may need to access the log file of the web server to find the error messages.



*Successful query of the database*

***Note that in a production environment, this allows people to access your database & manipulate it with 4GL instructions from the web.***

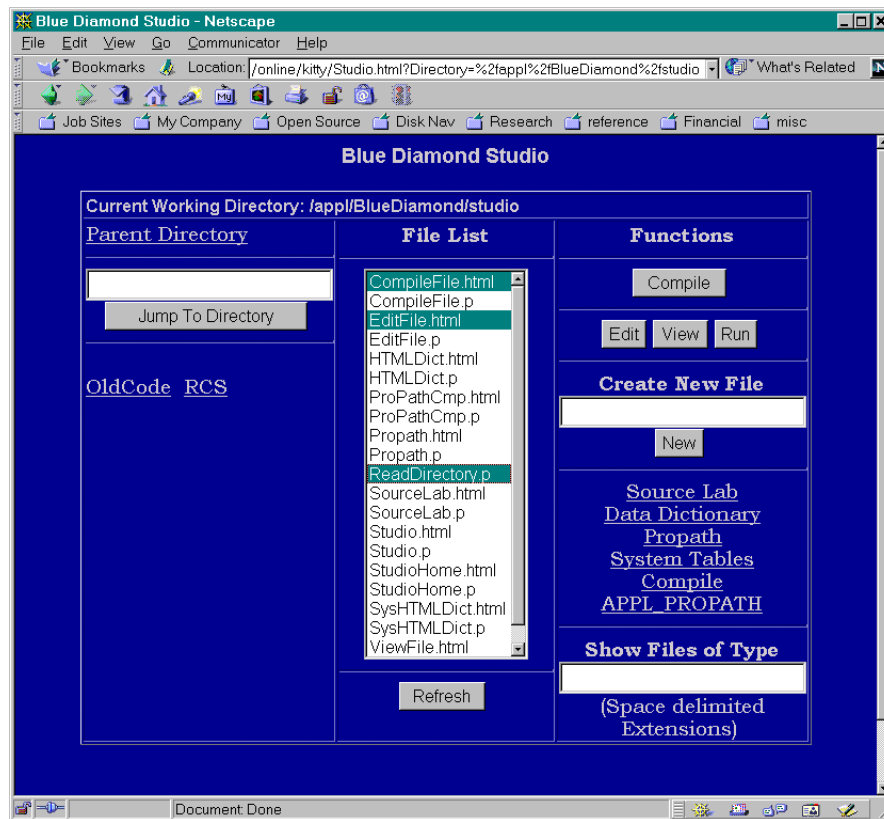
You can use the usual 4GL constructs that are not associated with displaying output to the screen. So such things as IF-THEN, CREATE, DELETE, FIND, etc. One can also use {include.i} files as in any other 4GL program.

For those elements of your program that you wish to put to the screen, you can use the PUT and DISPLAY statements as you normally would. The MESSAGE statement and such are not supported as of yet.

### ***Studio Compile Files***

---

In order for the Blue Diamond Transaction Server to handle your E4GL programs, they need to be converted into Progress .p files. Once done so, and with the proper Progress licensing, you can convert these files into .r's.



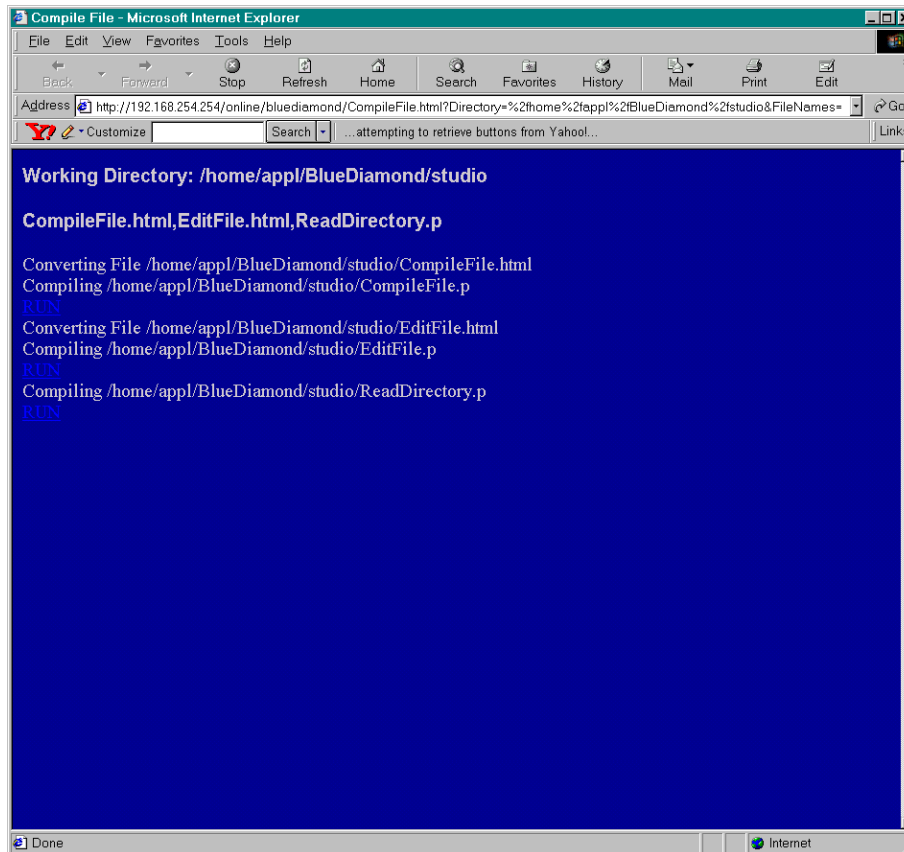
*Selecting multiple files for compiling*

The underlying architecture used by the Blue Diamond Transaction Server is different from the architecture of Webspeed. So you will NOT be able to use Studio to go from E4GL to .r's for use on Webspeed. You can use Studio to compile regular 4GL programs for Blue Diamond or Webspeed.

*The main point of Webspeed compatibility is at the source code level. You can compile E4GL source files on both Webspeed and Blue Diamond. However, the resulting .r files have different assumptions about the architecture available underneath<sup>4</sup>.*

---

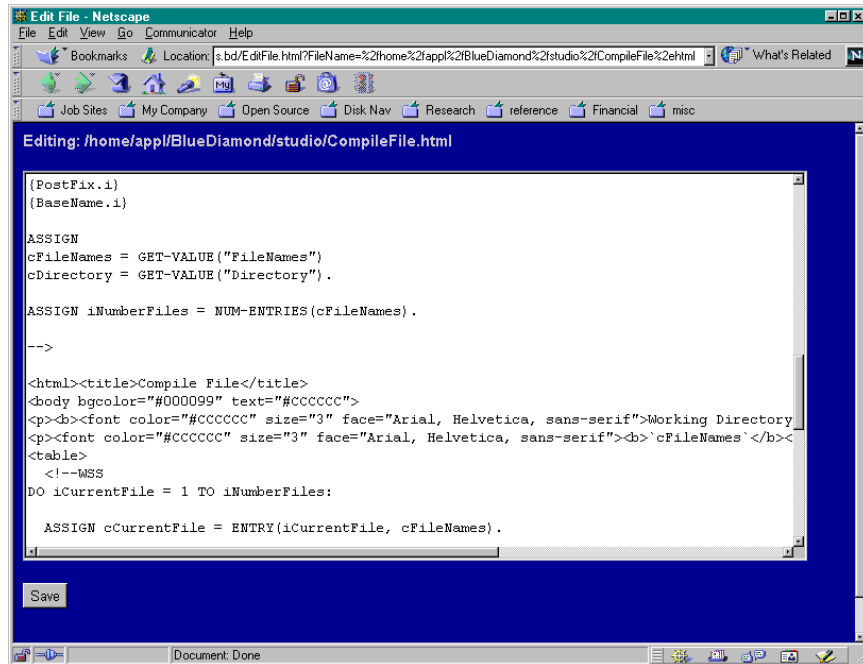
<sup>4</sup> By examining the resulting .p file under Blue Diamond, you will see the main difference is a STREAM called StdOut. This will be used in the .r files, but Webspeed has no such stream in it's architecture so the .r's will bomb out.



*Compile results*

### ***Studio Edit File***

Editing a file is done by navigating the directory structure of the machine via the file operations screen. By choosing ONE file (you cannot edit multiple files yet), you will be presented the following screen.



*Editing a file on the server*

It provides the file's source code in a simple editing box . By clicking on save, the file will be saved under it's given file name.

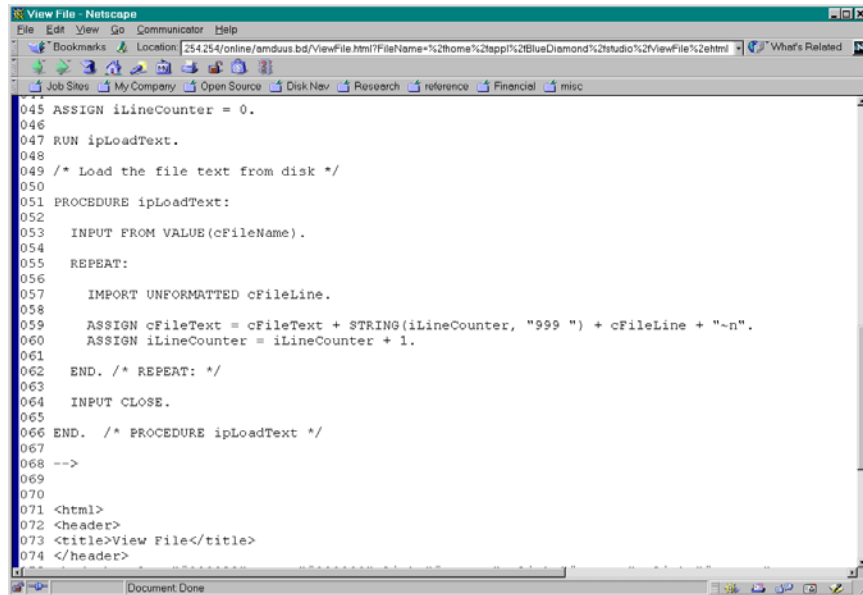
To create new files, you will need to access the UNIX box. If enough support money and license money comes in, this tool will be enhanced to create, modify, and delete files.

**Note:** Be sure to leave a blank line at the end of your files.

### *Viewing a file*

Sometimes one just wants to view a file. This can be accomplished by navigating the directory structure of the machine via the file operations screen. By selecting one file and clicking on the View button, the following screen will appear.



A screenshot of a Netscape browser window titled "View File - Netscape". The address bar shows a URL: "254254/online/amduus.bd/ViewFile.html?FileName=%2home%2appl%2BlueDiamond%2studio%2ViewFile%2html". The browser's menu bar includes File, Edit, View, Go, Communicator, and Help. The toolbar contains icons for Bookmarks, Location, and a search icon. Below the toolbar is a navigation bar with links: Job Sites, My Company, Open Source, Disk Nav, Research, reference, Financial, and misc. The main content area displays a file with line numbers from 045 to 074. The code is as follows:

```
045 ASSIGN iLineCounter = 0.
046
047 RUN ipLoadText.
048
049 /* Load the file text from disk */
050
051 PROCEDURE ipLoadText:
052
053     INPUT FROM VALUE(cFileName).
054
055     REPEAT:
056
057         IMPORT UNFORMATTED cFileLine.
058
059         ASSIGN cFileText = cFileText + STRING(iLineCounter, "999 ") + cFileLine + "~n".
060         ASSIGN iLineCounter = iLineCounter + 1.
061
062     END. /* REPEAT: */
063
064     INPUT CLOSE.
065
066 END. /* PROCEDURE ipLoadText */
067
068 -->
069
070
071 <html>
072 <header>
073 <title>View File</title>
074 </header>
```

The status bar at the bottom of the browser window shows "Document Done".

*Viewing a file on the server*

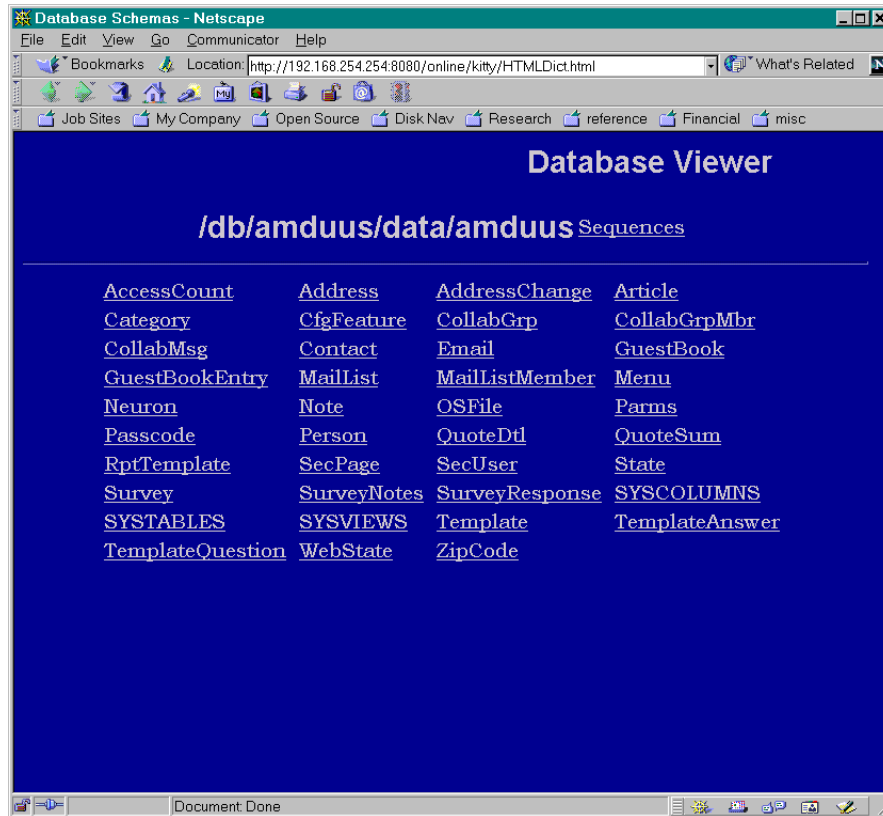
Viewing a file on the server shows line numbers next to the lines of code. This can be helpful in identifying lines in the generated .p files that are having compilation problems.

### ***Data Dictionary***

Through Blue Diamond Studio, you can access the tables available within the database via the metaschema. Remember that a database must be attached to your Blue Diamond session in order to use these screens.

On the file operations page, is a link called Dictionary. This link will bring you to the following screens.

### ***Tables Available***



The opening screen presents all the tables available to the program. {Warning: There is a bug with multiple DBs connected – it will only show the files of the current working DB – if you know how to fix this, let me know at sauge@amduus.com!}

By clicking on the table that one desires, the following screen will be presented.

Remember often you can open a new window off the hyperlink with you browser (usually called Open Link in New Window) which helps navigate the dictionary in a convenient manner.

---

### *Fields and Indices Available*

Database Viewer

/db/amduus/data/amduus

Parms Parameters

Field Name	Type	Format	Extent	Default	Description
Application	character	x(8)	1		Application Name
GroupName	character	x(35)	1		Used For Grouping Various Parameters Together
ParmName	character	X(35)	1		Parameter Name
ParmValue	character	x(60)	1		Parameter Value; Used like a Name/Value Pair

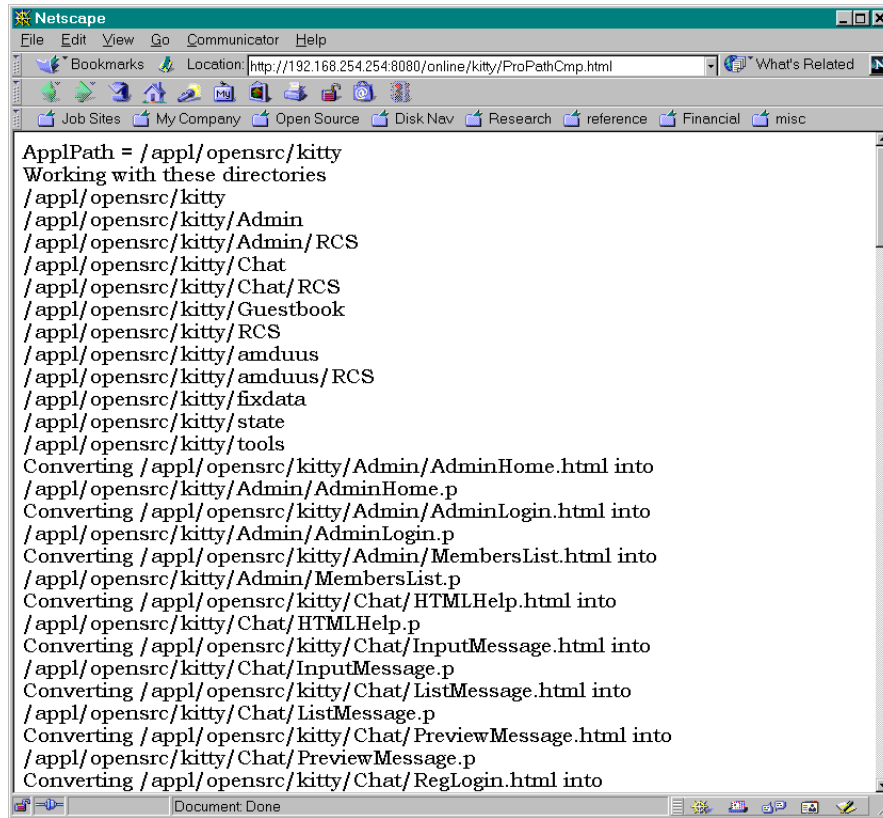
Primary Unique ParmsKey Application GroupName ParmName

This screen presents the fields available within the table.

### *Compiling the entire project with Studio:*

By setting the APPL\_PROPATH variable in the bluediamond script, you can inform Studio of all the directories that compose your application. It works just like PROPATH does and is appended to the PROPATH environmental variable late on in that script.

By clicking on the “Compile APPL\_PROPATH” link in the Studio, the Studio will identify all the directories and sub-directories in APPL\_PROPATH it will work with. It will then convert .html and .htm files into .p’s (or .i’s if the WSOPTIONS tag is present) and compile them.



*Results of the project compile*

Results of the compile are presented in the web page rendered.

---

## Explanation of the commands

### *Source Clean Up Script*

The `cleanp` script will look for .p's for HTML files that were compiled with Studio/Blue Diamond and delete them. It only operates on the current working directory.

**Note: The following scripts are from a time when Blue Diamond was a command line only development environment. If you plan on using Studio, you will not need to be concerned with this information.**

The configuration of the scripts can be done automatically done for you when you use the `mkcmds.bash` script.

The scripts will be postfixed with an environment tag. For example, for the `rpg` application, the scripts will read:

```
-rwxrwxrwx 1 sauge root 387 Oct 5 13:42 bdenv.rpg
-rwxrwxrwx 1 sauge root 456 Oct 5 13:28 bld4gl.rpg
-rwxrwxrwx 1 sauge root 229 Oct 5 13:28 cmp4gl.rpg
-rwxrwxrwx 1 sauge root 493 Oct 5 13:28 cnv4gl.rpg
-rwxrwxrwx 1 sauge root 232 Oct 5 13:28 mbld4gl.rpg
-rwxrwxrwx 1 sauge root 190 Oct 5 13:28 mcmp4gl.rpg
-rwxrwxrwx 1 sauge root 210 Oct 5 13:28 mcnv4gl.rpg
```

By this naming convention, all the scripts can be stored in the same directory as part of the `PATH` of the user, yet reference different databases and `PROPATHs`.

To create these scripts, one uses the `mkcmds.bash` script. It is invoked as follows:

```
mkcmds.bash environment_tag target_directory
```

where the `environment_tag` is the postfix to add to the end of the scripts and `target_directory` will be where the scripts are to be placed.

A sample invocation to make the above scripts is as follows:

```
mkcmds.bash rpg /home/sauge/bin
```

---

The mkcmds.bash script will create all the new scripts, internally modify the new scripts to call each other, and set the security to 777.

The following are commands that are set up when one creates an environment as in the preceding configuration section. The following command line interfaces are configured with the `bdenv.environment` shell. The only points of configuration for a Blue Diamond environment are the CGI script and the `bdenv.environment` shell.

### ***The E4GL to 4GL converter***

The E4GL to 4GL converter will convert the Webspeed code that has been slightly modified into pure 4GL with the requirements to be executed by the Blue Diamond Internet Transaction Server.

It is called by: `cnv4gl.environment [HTML File Name]`

This script should be made available to the PATH environmental variable of developers working on an application.

### ***The E4GL to 4GL Directory Converter***

The directory converter will locate all files ending in the prefix `.html` and pass them into the `cnv4gl` script for conversion to pure 4GL executable under the auspices of Blue Diamond.

It is called by: `mcnv4gl.environment`

### ***The 4GL Command Line Compiler***

Compiling can be done via the command line. This will find a lot of syntax or other types of errors in the program before submitting it to the web server.

It is called by: `cmp4gl.environment [Name of .p File]`

This script should be made available to the PATH environmental variable of developers working on an application.

### ***The 4GL Directory Compiler***

The directory compiler will locate all files ending in the prefix `.p` and pass them into the `cmp4gl` script for conversion to a `.r` executable under the auspices of Blue Diamond.

---

It is called by: `mcomp4gl.environment`

### ***The Build 4GL command***

The `bld4gl` command will convert an html file into a .p, and then compile that .p in one step.

It is called by: `bld4gl.environment [name of html file]`

### ***The Directory wide Build 4GL command***

The `mbld4gl` command will convert all HTML files in a directory into .p's and then compile those .p's.

It is called by: `mbld4gl.environment`

### ***The Single File Xref command***

The `xref4gl` command will compile and create XREF files for the .p file. The output file is `filename.p.xref`.

It is called by: `xref4gl.environment [name of .p file]`

### ***The Directory wide file Xref command***

The `mxref4gl` command will compile all the .p's in the directory and create Xref files for them. The output file is `filename.p.xref`.

It is called by: `mxref4gl.environment`

---

## How to make a program with Blue Diamond?

This question can make for a book in it's self. However, Blue Diamond works with E4GL like Webspeed (see the next section where there are inconsistencies.) So if one learns how to program with E4GL for Webspeed, one will have a very good idea of how to program for Blue Diamond.

The steps to take are the following:

1. Edit the .html file.
2. Execute the `bld4gl.environment` command or use Studio to update and compile.
3. Any Errors? Go to 1.
4. Execute the program via the web.
5. Any Run Time Errors? Errors usually end up in the web server's error log. Go to 1.
6. Any Semantic Errors? Go to 1.
7. Working Program!

The example program designed for Webspeed will function on Blue Diamond with no modifications. Note that HTML mapped programs cannot be executed on Blue Diamond.

An example program:

```
<!--WSS
/* ListTicket.html
    • Written by Scott Auge
    *
    • List Tickets owned by the login
*/

DEF VAR RCSVersion AS CHARACTER INIT "$Header:
/home/appl/srvexp/source/web/RCS/ListTicket.html,v 1.1 2001/01/21 01:21:35 root
Exp root $" NO-UNDO.

DEF VAR lUserID AS CHARACTER NO-UNDO.
DEF VAR lLoginID AS CHARACTER NO-UNDO.
DEF VAR pError AS CHARACTER NO-UNDO.

{api/Error.inc}
{web/AlertBox.inc}.

/* Determine our user */

ASSIGN
```



---

```

lUserID = GET-VALUE("UserID").

RUN api/state/GetLogin.p(INPUT lUserID,
                        OUTPUT lLoginID,
                        OUTPUT pError).

FIND SystemUser NO-LOCK
WHERE SystemUser.LoginID = lLoginID
NO-ERROR.

IF NOT AVAILABLE SystemUser THEN DO:

    AlertBox ("You do not appear to be logged in").
    RETURN.

END.

/* Find out address of user.  Tickets have their own addresses for customers */
/* with more than one site.                                                    */

FIND Address OF SystemUser NO-LOCK.

/* List the tickets */

-->

<head>
<title>Service Express List Tickets</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body bgcolor="#FFFFFF">
<table width="100%" border="0">
    <tr>
        <td><font size="4"><i><font face="Times New Roman, Times, serif"><b>Service
            Express</b></font></i></font></td>
        <td>
            <div align="right"><b><font face="Times New Roman, Times, serif"
color="#FF3333">Your
                Logo/Motto Here</font></b></div>
        </td>
    </tr>
</table>
<p align="center"><b><font face="Times New Roman, Times, serif">List
Tickets</font></b></p>
<p align="center"><b><font face="Times New Roman, Times,
serif">`SystemUser.FirstName` `SystemUser.LastName`<br>
    `Address.Street1` <br>
    `Address.Street2` <br>
    `Address.City` `Address.State`, `Address.Zip` <br>
    Phone: `SystemUser.Phone`<br>

```

---

---

```

Fax: `SystemUser.Fax`<br>
Pager: `SystemUser.Pager`<br>
Mobile: `SystemUser.Mobile`<br>
Email: `SystemUser.Email` </font></b></p>
<table width="75%" border="0" align="center" cellspacing="0">
  <tr bgcolor="#CCCCFF">
    <td>
      <center>
        <b><font face="Times New Roman, Times, serif">Ticket Reference</font>
        </b>
      </center>
    </td>
    <td>
      <center>
        <b><font face="Times New Roman, Times, serif">Status</font> </b>
      </center>
    </td>
    <td>
      <center>
        <b><font face="Times New Roman, Times, serif">Short Description</font>
        </b>
      </center>
    </td>
  </tr>
<!--WSS

/* Look up tickets owned by this person that are work in process. */

FOR EACH Ticket NO-LOCK
WHERE Ticket.OwnerSeq = SystemUser.UserSeq
  AND Ticket.IsWIP = TRUE
BY Ticket.Reference:

-->

  <tr>
    <td>
      <center>
        <font face="Times New Roman, Times, serif"><a
href="ViewTicket.html?TicketSeq=`Ticket.TicketSeq`">`Ticket.Reference`</a></font>
        </center>
      </td>
    <td>
      <center>
<!--WSS
FIND TicketStatus OF Ticket NO-LOCK.
-->
        <font face="Times New Roman, Times, serif">`TicketStatus.Name`</font>
        </center>
      </td>

```

---

---

```

        <td>
            <center>
                <font face="Times New Roman, Times,
serif">`Ticket.ShortStatement`</font>
            </center>
        </td>
    </tr>
<!--WSS
END.
-->
</table>
<p>&nbsp;</p>
<p align="center"><font color="#FF3333" face="Times New Roman, Times,
serif"><b>Any
    other links or information you would like here. </b></font></p>
<p align="center"><font color="#FF3333" face="Times New Roman, Times,
serif"><b>Customize
    with internal programmers, or use an acknowledged Service Express Programmer.
    </b></font></p>
<p>&nbsp;</p>
</body>
</html>

```

Once the programmer believes the file to be complete, s/he should issue the following command:

**Note: If you are using Blue Diamond Studio, all you need to do is click the file to compile and then the compile button. This command line program still works, but is not needed if you are using the Studio portion of Blue Diamond.**

```
cnv4gl ListTicket.html
```

After running the E4GL to 4GL converter, the programmer will have the following program which can be executed by Blue Diamond. It is provided to give the reader a deeper understanding of what is being accomplished:

```

/* DO NOT EDIT - THIS IS A GENERATED FILE */

{sga_util.i}
/* <!--WSS */
/* ListTicket.html
    • Written by Scott Auge
    *
    • List Tickets owned by the login
*/

```

---

```

DEF VAR RCSVersion AS CHARACTER INIT "$Header:
/home/appl/srvexp/source/web/RCS/ListTicket.html,v 1.1 2001/01/21 01:21:35 root
Exp root $" NO-UNDO.

DEF VAR lUserID AS CHARACTER NO-UNDO.
DEF VAR lLoginID AS CHARACTER NO-UNDO.
DEF VAR pError AS CHARACTER NO-UNDO.

{api/Error.inc}
{web/AlertBox.inc}.

/* Determine our user */

ASSIGN
lUserID = GET-VALUE("UserID").

RUN api/state/GetLogin.p(INPUT lUserID,
                        OUTPUT lLoginID,
                        OUTPUT pError).

FIND SystemUser NO-LOCK
WHERE SystemUser.LoginID = lLoginID
NO-ERROR.

IF NOT AVAILABLE SystemUser THEN DO:

    AlertBox ("You do not appear to be logged in").
    RETURN.

END.

/* Find out address of user. Tickets have their own addresses for customers */
/* with more than one site. */

FIND Address OF SystemUser NO-LOCK.

/* List the tickets */

/* --> */

PUT STREAM StdOut UNFORMATTED '<head>~n'.
PUT STREAM StdOut UNFORMATTED '<title>Service Express List Tickets</title>~n'.
PUT STREAM StdOut UNFORMATTED '<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">~n'.
PUT STREAM StdOut UNFORMATTED '</head>~n'.
PUT STREAM StdOut UNFORMATTED '<body bgcolor="#FFFFFF">~n'.
PUT STREAM StdOut UNFORMATTED '<table width="100%" border="0">~n'.
PUT STREAM StdOut UNFORMATTED ' <tr> ~n'.
PUT STREAM StdOut UNFORMATTED ' <td><font size="4"><i><font face="Times New
Roman, Times, serif"><b>Service ~n'.
PUT STREAM StdOut UNFORMATTED ' Express</b></font></i></font></td>~n'.

```

---

---

```

PUT STREAM StdOut UNFORMATTED `      <td> ~n'.
PUT STREAM StdOut UNFORMATTED `      <div align="right"><b><font face="Times
New Roman, Times, serif" color="#FF3333">Your ~n'.
PUT STREAM StdOut UNFORMATTED `      Logo/Motto Here</font></b></div>~n'.
PUT STREAM StdOut UNFORMATTED `      </td>~n'.
PUT STREAM StdOut UNFORMATTED ` </tr>~n'.
PUT STREAM StdOut UNFORMATTED `</table>~n'.
PUT STREAM StdOut UNFORMATTED `<p align="center"><b><font face="Times New
Roman, Times, serif">List Tickets</font></b></p>~n'.
PUT STREAM StdOut UNFORMATTED `<p align="center"><b><font face="Times New
Roman, Times, serif">' SystemUser.FirstName ` ` SystemUser.LastName `<br>~n'.
PUT STREAM StdOut UNFORMATTED ` ` Address.Street1 ` <br>~n'.
PUT STREAM StdOut UNFORMATTED ` ` Address.Street2 ` <br>~n'.
PUT STREAM StdOut UNFORMATTED ` ` Address.City ` ` Address.State `, `
Address.Zip ` <br>~n'.
PUT STREAM StdOut UNFORMATTED ` Phone: ` SystemUser.Phone `<br>~n'.
PUT STREAM StdOut UNFORMATTED ` Fax: ` SystemUser.Fax `<br>~n'.
PUT STREAM StdOut UNFORMATTED ` Pager: ` SystemUser.Pager `<br>~n'.
PUT STREAM StdOut UNFORMATTED ` Mobile: ` SystemUser.Mobile `<br>~n'.
PUT STREAM StdOut UNFORMATTED ` Email: ` SystemUser.Email `
</font></b></p>~n'.
PUT STREAM StdOut UNFORMATTED `<table width="75%" border="0" align="center"
cellspacing="0">~n'.
PUT STREAM StdOut UNFORMATTED ` <tr bgcolor="#CCCCFF"> ~n'.
PUT STREAM StdOut UNFORMATTED `      <td> ~n'.
PUT STREAM StdOut UNFORMATTED `      <center>~n'.
PUT STREAM StdOut UNFORMATTED `      <b><font face="Times New Roman, Times,
serif">Ticket Reference</font> ~n'.
PUT STREAM StdOut UNFORMATTED `      </b> ~n'.
PUT STREAM StdOut UNFORMATTED `      </center>~n'.
PUT STREAM StdOut UNFORMATTED ` </td>~n'.
PUT STREAM StdOut UNFORMATTED ` <td> ~n'.
PUT STREAM StdOut UNFORMATTED `      <center>~n'.
PUT STREAM StdOut UNFORMATTED `      <b><font face="Times New Roman, Times,
serif">Status</font> </b> ~n'.
PUT STREAM StdOut UNFORMATTED `      </center>~n'.
PUT STREAM StdOut UNFORMATTED ` </td>~n'.
PUT STREAM StdOut UNFORMATTED ` <td> ~n'.
PUT STREAM StdOut UNFORMATTED `      <center>~n'.
PUT STREAM StdOut UNFORMATTED `      <b><font face="Times New Roman, Times,
serif">Short Description</font> ~n'.
PUT STREAM StdOut UNFORMATTED `      </b> ~n'.
PUT STREAM StdOut UNFORMATTED `      </center>~n'.
PUT STREAM StdOut UNFORMATTED ` </td>~n'.
PUT STREAM StdOut UNFORMATTED ` </tr>~n'. /* <!--WSS */

/* Look up tickets owned by this person that are work in process. */

FOR EACH Ticket NO-LOCK
WHERE Ticket.OwnerSeq = SystemUser.UserSeq
  AND Ticket.IsWIP = TRUE

```

---

---

BY Ticket.Reference:

```
/* --> */

PUT STREAM StdOut UNFORMATTED ` <tr> ~n'.
PUT STREAM StdOut UNFORMATTED ` <td> ~n'.
PUT STREAM StdOut UNFORMATTED ` <center>~n'.
PUT STREAM StdOut UNFORMATTED ` <font face="Times New Roman, Times,
serif"><a href="ViewTicket.html?TicketSeq=' Ticket.TicketSeq `"'>'
Ticket.Reference `</a></font> ~n'.
PUT STREAM StdOut UNFORMATTED ` </center>~n'.
PUT STREAM StdOut UNFORMATTED ` </td>~n'.
PUT STREAM StdOut UNFORMATTED ` <td> ~n'.
PUT STREAM StdOut UNFORMATTED ` <center>~n'. /* <!--WSS */
FIND TicketStatus OF Ticket NO-LOCK.
/* --> */

PUT STREAM StdOut UNFORMATTED ` <font face="Times New Roman, Times,
serif">' TicketStatus.Name `</font> ~n'.
PUT STREAM StdOut UNFORMATTED ` </center>~n'.
PUT STREAM StdOut UNFORMATTED ` </td>~n'.
PUT STREAM StdOut UNFORMATTED ` <td> ~n'.
PUT STREAM StdOut UNFORMATTED ` <center>~n'.
PUT STREAM StdOut UNFORMATTED ` <font face="Times New Roman, Times,
serif">' Ticket.ShortStatement `</font> ~n'.
PUT STREAM StdOut UNFORMATTED ` </center>~n'.
PUT STREAM StdOut UNFORMATTED ` </td>~n'.
PUT STREAM StdOut UNFORMATTED ` </tr>~n'. /* <!--WSS */
END.
/* --> */

PUT STREAM StdOut UNFORMATTED `</table>~n'.
PUT STREAM StdOut UNFORMATTED `<p>&nbsp;</p>~n'.
PUT STREAM StdOut UNFORMATTED `<p align="center"><font color="#FF3333"
face="Times New Roman, Times, serif"><b>Any ~n'.
PUT STREAM StdOut UNFORMATTED ` other links or information you would like
here. </b></font></p>~n'.
PUT STREAM StdOut UNFORMATTED `<p align="center"><font color="#FF3333"
face="Times New Roman, Times, serif"><b>Customize ~n'.
PUT STREAM StdOut UNFORMATTED ` with internal programmers, or use an
acknowledged Service Express Programmer. ~n'.
PUT STREAM StdOut UNFORMATTED ` </b></font></p>~n'.
PUT STREAM StdOut UNFORMATTED `<p>&nbsp;</p>~n'.
PUT STREAM StdOut UNFORMATTED `</body>~n'.
PUT STREAM StdOut UNFORMATTED `</html>~n'.
```

**Note: There is now a tool by Steven Jellin that will concatenate PUT STREAM StdOut UNFORMATTED lines into a single statement. It is still experimental, but is found in the BlueDiamond/plusscript directory. This will help reduce your R code size and performs some optimizations.**

---

## How do I convert my current Webspeed E4GL programs to Blue Diamond?

You must have the source code to your applications. Blue Diamond does not execute .r's created with Workshop.

Blue Diamond works only on E4GL based programs. If your application is written using the HTML-Mapped method, Blue Diamond will not be able to execute it.

If your E4GL based program makes use of any of the ADM functionality or any other Progress code based architecture, Blue Diamond will not function. Blue Diamond does not contain any code derived from the works of Progress Software Corporation.

Blue Diamond supports the following:

- The use of `<WSS` and `-->` tags around 4GL segments. It is best if these are on separate lines.

- The use of `<script language="speedscript"></script>` tags around 4GL segments. It is best if these are on separate lines.

- The `GET-VALUE ()` function.

- The `GET-FIELD ()` function.

- The `GET-COOKIE ()` function.

- The `GET-CGI ()` function.

- The `SET-COOKIE ()` function (expires upon close of the browser, no "sticky" cookies yet)

- The `HTML-ENCODE ()` function.

- The `URL-ENCODE ()` function.

- The use of ``` around CHAR based expressions embedded within HTML.

- The `OUTPUT-HEADERS` program.

All the 4GL statements supported by the Progress Character Client (excluding display based statements such as `DISPLAY` or `FRAME`). HTML is used to output information, not these kinds of statements.

Blue Diamond was written to use existing E4GL programs to the best of its ability. This is because the author found himself with a bunch of E4GL that became really expensive to execute with Progress' change in Webspeed Pricing policy. It was much easier to write a transaction server than to re-write all the code into something else.

---

## How do I convert my Blue Diamond programs to Webspeed?

Currently, Blue Diamond handles a subset of Webspeed functionality. You do not need to make any changes to your source code to run your Blue Diamond programs under Webspeed.

You must however recompile your programs using Webspeed Workshop in order for them to execute under Webspeed.

## How can I change the content-type in a Blue Diamond program?

Currently Blue Diamond assumes a page of type text/html is being sent out. The content type can be changed by editing the line `RUN SetContent("text/html")` in the `sga_util.i` program. Understand this will effect ALL programs converted with `cnv4gl` or `mcnv4gl` as they prefix the generated code with this include.

To have more control over this, one can remove the `RUN SetContent("text/html")` line from `sga_util.i`, and place it in your `.html .htm` file. This will be a departure will make the page executable on Blue Diamond only, as it removes the source from Webspeed/Workshop compliance.

## Supported WSMETA tags

```
<!--WSMETA NAME="wsoptions" CONTENT="include" -->
```

When this tag is a line or a portion of a line in a file, the output file will become an include file (\*.i) instead of a .p file. The entire tag syntax must be on one line.

This is convenient to make .html or .htm files into include files.



---

## Blue Diamond Routines

Blue Diamond has routines pre-made that are available to the programmer. These routines perform tasks such as:

- Handling configurations for your application
- Handling session information for your application's users
- Handling login, logout, counting, and limiting the number of users of your application
- Handling security in your application
- Sending mail from the application

***NOTE THAT WEBSPEED DOES NOT HAVE ALL OF THESE ROUTINES. IF YOU WRITE A PROGRAM USING THESE ROUTINES, YOU WILL NEED TO PORT THEM OVER TO WEBSPEED (PRETTY MUCH JUST COPY THE FUNCTIONS INTO .i FILES AND INCLUDE IN YOUR PROGRAM.) TO INSURE YOUR PROGRAM IS COMPATIBLE WITH WEBSPEED, BE SURE TO USE WEBSPEED WORKSHOP USING ROUTINES WITH COMMON NAMES.***

### ***Blue Diamond HTML and support Routines***

These routines are found in the sga\_util.i function which is included in every Blue Diamond generated source file.

#### IntToHex ()

FUNCTION IntToHex RETURNS CHARACTER (INPUT n AS INTEGER):

#### URL-ENCODE()

Inside prgsrc/Config.i, if ORGURLENCODE is defined as YES, then this

FUNCTION URL-ENCODE RETURNS CHARACTER (INPUT t AS CHARACTER):

otherwise the Progress compliant version here:

FUNCTION URL-ENCODE RETURNS CHARACTER (INPUT t AS CHARACTER, INPUT s AS CHARACTER):

By default, the Progress compliant version is used.

#### HexToInt()

FUNCTION HexToInt RETURNS INTEGER (INPUT Hex AS CHAR):

---

### URL-DECODE()

FUNCTION URL-DECODE RETURNS CHARACTER (INPUT r AS CHARACTER):

### SetContent

PROCEDURE SetContent:

### GET-FIELD()

FUNCTION Get-Field RETURNS CHARACTER (INPUT T AS CHARACTER):

### GET-CGI()

FUNCTION GET-CGI RETURNS CHARACTER (INPUT pText AS CHARACTER):

### GET-COOKIE()

FUNCTION Get-Cookie RETURNS CHARACTER (INPUT T AS CHARACTER):

### GET-VALUE()

FUNCTION Get-Value RETURNS CHARACTER (INPUT T AS CHARACTER):

### SET-COOKIE()

FUNCTION Set-Cookie RETURNS CHARACTER  
(INPUT pName AS CHARACTER,  
INPUT pValue AS CHARACTER,  
INPUT pDate AS DATE,  
INPUT pTime AS INTEGER,  
INPUT pPath AS CHARACTER,  
INPUT pDomain AS CHARACTER,  
INPUT pOptions AS CHARACTER):

### DELETE-COOKIE()

FUNCTION Delete-Cookie RETURNS CHARACTER (INPUT CookieName AS  
CHARACTER, INPUT A  
AS CHARACTER, INPUT B AS CHARACTER):

### HTML-ENCODE()

FUNCTION HTML-Encode RETURNS CHARACTER (INPUT pText AS CHARACTER):

---

### IntToHex()

FUNCTION IntToHex RETURNS CHARACTER (INPUT n AS INTEGER):

### URL-ENCODE()

FUNCTION URL-ENCODE RETURNS CHARACTER (INPUT t AS CHARACTER):

or

FUNCTION URL-ENCODE RETURNS CHARACTER (INPUT A AS CHARACTER , INPUT B AS CHARACTER ):

If &ORGURLENCODE=Yes then the first definition is used. If not, then the second definition is used. The second definition is Webspeed 3.1 compliant. You can set ORGURLENCODE in prgsrsrc/sga\_util.i

By default the second definition is used.

### HexToInt()

FUNCTION HexToInt RETURNS INTEGER (INPUT Hex AS CHAR):

### URL-DECODE()

FUNCTION URL-DECODE RETURNS CHARACTER (INPUT r AS CHARACTER):

## ***Application Configuration Routines***

### DeleteParm.p

```
DEF INPUT PARAMETER pApplication AS CHARACTER NO-UNDO.  
DEF INPUT PARAMETER pGroupName  AS CHARACTER NO-UNDO.  
DEF INPUT PARAMETER pParmName   AS CHARACTER NO-UNDO.  
DEF OUTPUT PARAMETER pError     AS CHARACTER NO-UNDO.
```

### ReadParm.p

```
DEF INPUT PARAMETER pApplication AS CHARACTER NO-UNDO.  
DEF INPUT PARAMETER pGroupName  AS CHARACTER NO-UNDO.  
DEF INPUT PARAMETER pParmName   AS CHARACTER NO-UNDO.  
DEF OUTPUT PARAMETER pParmValue AS CHARACTER NO-UNDO.  
DEF OUTPUT PARAMETER pError     AS CHARACTER NO-UNDO.
```

### SetConf.p

```
DEF INPUT PARAMETER pFileName AS CHARACTER NO-UNDO.
```

---

```
DEF INPUT PARAMETER pApplication AS CHARACTER NO-UNDO.
```

### WriteParm.p

```
DEF INPUT PARAMETER pApplication AS CHARACTER NO-UNDO.  
DEF INPUT PARAMETER pGroupName AS CHARACTER NO-UNDO.  
DEF INPUT PARAMETER pParmName AS CHARACTER NO-UNDO.  
DEF INPUT PARAMETER pParmValue AS CHARACTER NO-UNDO.  
DEF OUTPUT PARAMETER pError AS CHARACTER NO-UNDO.
```

### inird.p

```
DEF INPUT PARAMETER pFileName AS CHARACTER NO-UNDO.  
DEF INPUT PARAMETER pStanza AS CHARACTER NO-UNDO.  
DEF INPUT PARAMETER pParm AS CHARACTER NO-UNDO.  
DEF OUTPUT PARAMETER pValue AS CHARACTER NO-UNDO.
```

## ***Application User Session Saving Routines***

### ClearState.p

```
DEF INPUT PARAMETER pSessionID AS CHARACTER NO-UNDO.  
DEF OUTPUT PARAMETER pError AS CHARACTER NO-UNDO.
```

### ReadState.p

```
DEF INPUT PARAMETER pSessionID AS CHARACTER NO-UNDO.  
DEF INPUT PARAMETER pCategory AS CHARACTER NO-UNDO.  
DEF INPUT PARAMETER pName AS CHARACTER NO-UNDO.  
DEF OUTPUT PARAMETER pValue AS CHARACTER NO-UNDO.  
DEF OUTPUT PARAMETER pError AS CHARACTER NO-UNDO.
```

### WriteState.p

```
DEF INPUT PARAMETER pSessionID AS CHARACTER NO-UNDO.  
DEF INPUT PARAMETER pCategory AS CHARACTER NO-UNDO.  
DEF INPUT PARAMETER pName AS CHARACTER NO-UNDO.  
DEF INPUT PARAMETER pValue AS CHARACTER NO-UNDO.  
DEF OUTPUT PARAMETER pError AS CHARACTER NO-UNDO.
```

## ***Miscellaneous Routines***

These routines are not available in Webspeed base product.

---

### RandomString.p

Found in MakeID3.i

Creates a random string.

### MakeID3()

Found in MakeID3.i

Creates a random based identification string.

### Error.i

Used for the possible error returns from the various Blue Diamond routines.

### PostFix()

FUNCTION PostFix RETURNS CHARACTER (INPUT cFileName AS CHARACTER )

PositFix.i

Returns the Postfix of the input.

### SetPostFix()

FUNCTION SetPostFix RETURNS CHAR (INPUT cFileName AS CHARACTER,  
INPUT cPostFix AS CHARACTER):

SetPostFix.i

Sets the cFileName to have the Postfix cPostFix. If an existing postfix is on the filename, it is replaced with the new postfix.

### BaseName()

FUNCTION BaseName RETURNS CHARACTER (INPUT cFileName AS CHARACTER ):

BaseName.i

Returns the path and name of file without the postfix.

---

### FilePath()

FUNCTION FilePath RETURNS CHARACTER ( INPUT a AS CHARACTER ):

Returns the path portion of a path/filename combination.

Path.i

### FilePath()

FUNCTION FileName RETURNS CHARACTER ( INPUT a AS CHARACTER ):

Returns the filename portion of a path/filename combination.

Path.i

## ***Application User Accounting Routines***

### LoginUser.p

```
DEF INPUT PARAMETER cUserID      AS CHARACTER NO-UNDO.  
DEF INPUT PARAMETER cPassword    AS CHARACTER NO-UNDO.  
DEF OUTPUT PARAMETER cMessage    AS CHARACTER NO-UNDO.
```

### LogoutUser.p

```
DEF INPUT PARAMETER cCMSID AS CHARACTER NO-UNDO.
```

## ***Handling Mail With The Application***

### smtplib.p

```
DEF INPUT PARAMETER mailhub      as char no-undo.  
DEF INPUT PARAMETER EmailTo      AS CHAR NO-UNDO.  
DEF INPUT PARAMETER EmailFrom    AS CHAR NO-UNDO.  
DEF INPUT PARAMETER EmailCC      AS CHAR NO-UNDO.  
DEF INPUT PARAMETER Attachments  AS CHAR NO-UNDO.  
DEF INPUT PARAMETER LocalFiles   AS CHAR NO-UNDO.  
DEF INPUT PARAMETER Subject      AS CHAR NO-UNDO.  
DEF INPUT PARAMETER Body         AS CHAR NO-UNDO.  
DEF INPUT PARAMETER MIMEHeader   AS CHAR NO-UNDO.  
DEF INPUT PARAMETER BodyType     as char no-undo.  
  
DEF OUTPUT PARAMETER oSuccessful AS LOGICAL NO-UNDO.  
DEF OUTPUT PARAMETER vMessage    AS CHAR NO-UNDO.
```

---

The following are found in pop3api.i, it requires the same source to have pop3.i also.

**PROCEDURE ConnectToServer:**

```
DEF INPUT PARAMETER MailServer AS CHARACTER NO-UNDO.  
DEF OUTPUT PARAMETER hHandle AS HANDLE NO-UNDO.  
DEF OUTPUT PARAMETER Result AS INTEGER NO-UNDO.
```

**PROCEDURE DisconnectFromServer:**

```
DEF INPUT PARAMETER hSocket AS HANDLE NO-UNDO.
```

**PROCEDURE Login:**

```
DEF INPUT PARAMETER hSocket AS HANDLE NO-UNDO.  
DEF INPUT PARAMETER pUserID AS CHARACTER NO-UNDO.  
DEF INPUT PARAMETER pPassword AS CHARACTER NO-UNDO.  
DEF OUTPUT PARAMETER pMessageText AS CHARACTER NO-UNDO.
```

**PROCEDURE ListMessages:**

```
DEF INPUT PARAMETER hSocket AS HANDLE NO-UNDO.  
DEF OUTPUT PARAMETER pMessageText AS CHARACTER NO-UNDO.
```

**PROCEDURE RetrieveMessage:**

```
DEF INPUT PARAMETER hSocket as HANDLE NO-UNDO.  
DEF INPUT PARAMETER pMessageID AS CHARACTER NO-UNDO.  
DEF OUTPUT PARAMETER pMessageText AS CHARACTER NO-UNDO.
```

**PROCEDURE DeleteMessage:**

```
DEF INPUT PARAMETER hSocket as HANDLE NO-UNDO.  
DEF INPUT PARAMETER pMessageID AS CHARACTER NO-UNDO.  
DEF OUTPUT PARAMETER lServerResponse AS CHARACTER NO-UNDO.
```

**PROCEDURE Logout:**

```
DEF INPUT PARAMETER hSocket as HANDLE NO-UNDO.  
DEF OUTPUT PARAMETER lServerResponse AS CHARACTER NO-UNDO.
```

---

## *Application Security Routines*

### Loggedin.i

Identifies a user that is logged in by bringing up the SecUser record for that user. Else will output an error message.



---

## Database Tables For Blue Diamond

04/01/02 16:07:01      PROGRESS Report  
Database: bd (PROGRESS)

Table Name	Description
AccessCount	
Parms	Parameters
Person	Individuals within the system.
SecUser	Security information for a system user
WebState	Used to store session information about a web based log in.

=====  
===== Table: AccessCount =====

Table Flags: "f" = frozen, "s" = a SQL table

Table Name	Dump Name	Table Field Count	Index Count	Table Label
AccessCount	accessco	3	2	AccessCount

Storage Area: Data Area

=====  
===== FIELD SUMMARY =====  
===== Table: AccessCount =====

Flags: <c>ase sensitive, <i>ndex component, <m>andatory, <v>iew component

Order	Field Name	Data Type	Flags	Format	Initial
10	AccessCountID	char	i	x(8)	
20	Name	char	i	x(8)	
30	Counter	inte		->, >>>, >>9	0

Field Name	Label	Column Label
AccessCountID	AccessCountID	AccessCountID
Name	Name	Name
Counter	Counter	Counter

=====  
===== INDEX SUMMARY =====  
===== Table: AccessCount =====

---

Flags: <p>primary, <u>nique, <w>ord, <a>bbreviated, <i>nactive, + asc, - desc

Flags	Index Name	Cnt	Field Name
pu	pukey	1	+ AccessCountID
u	pukey2	1	+ Name

\*\* Index Name: pukey  
Storage Area: Data Area  
\*\* Index Name: pukey2  
Storage Area: Data Area

===== FIELD DETAILS =====  
===== Table: AccessCount =====

=====   
===== Table: Parm s =====

Table Flags: "f" = frozen, "s" = a SQL table

Table Name	Dump Name	Table Field Flags	Index Count	Table Count	Label
Parms	parms		4	1	Parameters

Description: Parameters  
Storage Area: Data Area

===== FIELD SUMMARY =====  
===== Table: Parm s =====

Flags: <c>ase sensitive, <i>ndex component, <m>andatory, <v>iew component

Order	Field Name	Data Type	Flags	Format	Initial
20	Application	char	i	x(8)	
30	GroupName	char	i	x(35)	
40	ParmName	char	i	X(35)	
50	ParmValue	char		x(60)	

Field Name	Label	Column Label
Application	Application	Application
GroupName	Group Name	Group Name

---

ParmName	Parm Name	Parm Name
ParmValue	Parm Value	Parm Value

```
===== INDEX SUMMARY =====
===== Table: Pams =====
```

Flags: <p>primary, <u>nique, <w>ord, <a>bbreviated, <i>nactive, + asc, - desc

Flags	Index Name	Cnt	Field Name
-----	-----	-----	-----
pu	ParmsKey	3	+ Application + GroupName + ParmName

\*\* Index Name: PamsKey  
Storage Area: Data Area

```
===== FIELD DETAILS =====
===== Table: Pams =====
```

\*\* Field Name: Application  
Description: Application Name  
Help: Enter Application Name

\*\* Field Name: GroupName  
Description: Used For Grouping Various Parameters Together  
Help: Enter Group Name

\*\* Field Name: ParmName  
Description: Parameter Name  
Help: Enter Parameter Name

\*\* Field Name: ParmValue  
Description: Parameter Value; Used like a Name/Value Pair  
Help: Enter Parameter Value

```
=====
===== Table: Person =====
```

Table Flags: "f" = frozen, "s" = a SQL table

Table Name	Dump Name	Table Field Count	Index Count	Table Label
-----	-----	-----	-----	-----
Person	person	5	3	Person

---

Description: Individuals within the system.  
Storage Area: Data Area

===== FIELD SUMMARY =====  
===== Table: Person =====

Flags: <c>ase sensitive, <i>ndex component, <m>andatory, <v>iew component

Order	Field Name	Data Type	Flags	Format	Initial
10	PersonID	char	i	x(8)	
20	FirstName	char		x(8)	
30	LastName	char		x(8)	
40	AddressID	char	i	x(8)	
50	ContactID	char	i	x(8)	

Field Name	Label	Column Label
PersonID	PersonID	PersonID
FirstName	FirstName	FirstName
LastName	LastName	LastName
AddressID	AddressID	AddressID
ContactID	ContactID	ContactID

===== INDEX SUMMARY =====  
===== Table: Person =====

Flags: <p>primary, <u>nique, <w>ord, <a>bbreviated, <i>nactive, + asc, - desc

Flags	Index Name	Cnt	Field Name
pu	key1	1	+ PersonID
	key2	1	+ AddressID
	key3	1	+ ContactID

\*\* Index Name: key1  
Storage Area: Data Area  
\*\* Index Name: key2  
Storage Area: Data Area  
\*\* Index Name: key3  
Storage Area: Data Area

===== FIELD DETAILS =====  
===== Table: Person =====

```
=====
===== Table: SecUser =====
```

Table Flags: "f" = frozen, "s" = a SQL table

Table Name	Dump Name	Table Flags	Field Count	Index Count	Table Label
SecUser	secuser		6	3	SecUser

Description: Security information for a system user  
Storage Area: Data Area

```
===== FIELD SUMMARY =====
===== Table: SecUser =====
```

Flags: <c>ase sensitive, <i>ndex component, <m>andatory, <v>iew component

Order	Field Name	Data Type	Flags	Format	Initial
10	SecUserID	char	i	x(8)	
20	PersonID	char	i	x(8)	
30	Login	char		x(8)	
40	Password	char		x(8)	
50	SecProfileID	char	i	x(8)	
60	Confirmed	logi		yes/no	no

Field Name	Label	Column Label
SecUserID	SecUserID	SecUserID
PersonID	PersonID	PersonID
Login	Login	Login
Password	Password	Password
SecProfileID	SecProfileID	SecProfileID
Confirmed	Confirmed	Confirmed

```
===== INDEX SUMMARY =====
===== Table: SecUser =====
```

Flags: <p>primary, <u>nique, <w>ord, <a>bbreviated, <i>nactive, + asc, - desc

Flags	Index Name	Cnt	Field Name
pu	key1	1	+ SecUserID
	key2	1	+ PersonID

```

key3
1 + SecProfileID

** Index Name: key1
Storage Area: Data Area
** Index Name: key2
Storage Area: Data Area
** Index Name: key3
Storage Area: Data Area

===== FIELD DETAILS =====
===== Table: SecUser =====

** Field Name: Confirmed
Description: User confirmed to be who they say they are
Help: User confirmed to be who they say they are

=====
===== Table: WebState =====

Table Flags: "f" = frozen, "s" = a SQL table

Table
Name
-----
WebState
Dump
Name
webstate
Table
Flags
5
Field
Count
2
Index
Count
WebState
Table
Label

Description: Used to store session information about a web based log in.
Storage Area: Data Area

===== FIELD SUMMARY =====
===== Table: WebState =====

Flags: <c>ase sensitive, <i>ndex component, <m>andatory, <v>iew component

Order
Field Name
Data Type
Flags
Format
Initial
-----
10 SessionID
char
i
x(8)
20 Category
char
i
x(8)
30 Name
char
i
x(8)
40 Data
char
x(8)
50 CreateDate
date
99/99/99
TODAY

Field Name
Label
Column Label
-----
SessionID
SessionID
SessionID
Category
Category
Category

```

---

Name	Name	Name
Data	Data	Data
CreateDate	CreateDate	CreateDate

===== INDEX SUMMARY =====  
===== Table: WebState =====

Flags: <p>primary, <u>nique, <w>ord, <a>bbreviated, <i>nactive, + asc, - desc

Flags	Index Name	Cnt	Field Name
	key1	1	+ SessionID
pu	pukey	3	+ SessionID + Category + Name

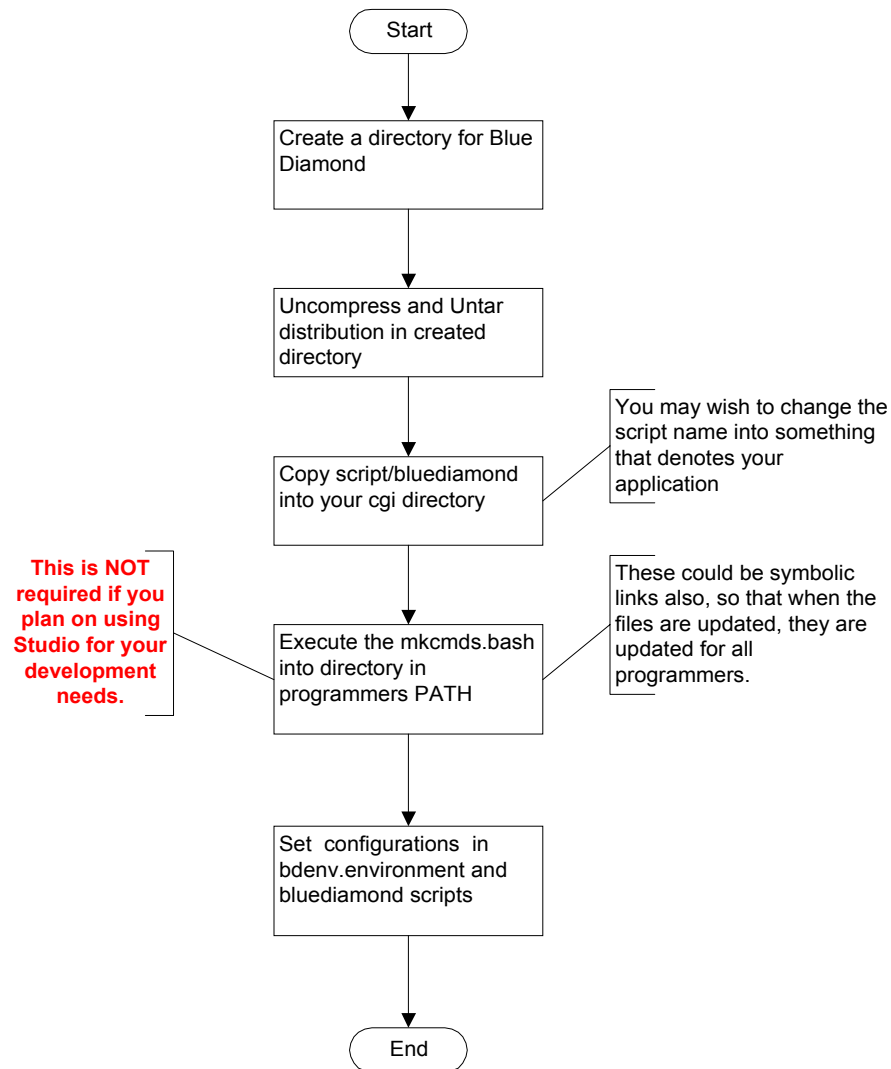
\*\* Index Name: key1  
Storage Area: Data Area  
\*\* Index Name: pukey  
Storage Area: Data Area

===== FIELD DETAILS =====  
===== Table: WebState =====

===== SEQUENCES =====

---

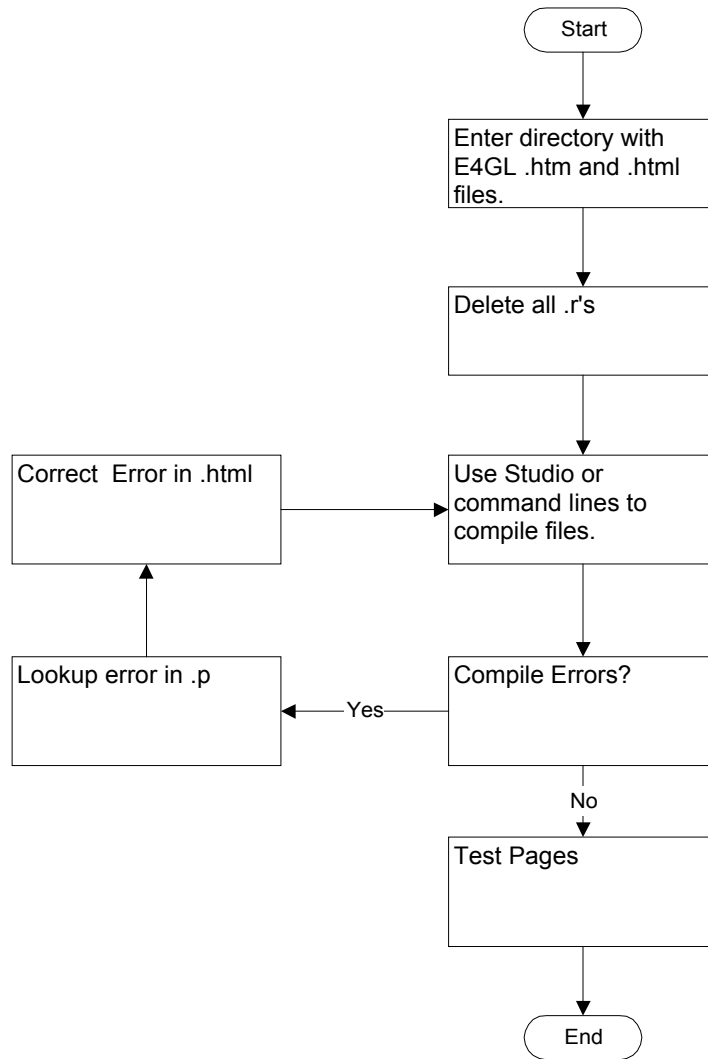
## Appendix A: General flow of steps to installing Blue Diamond





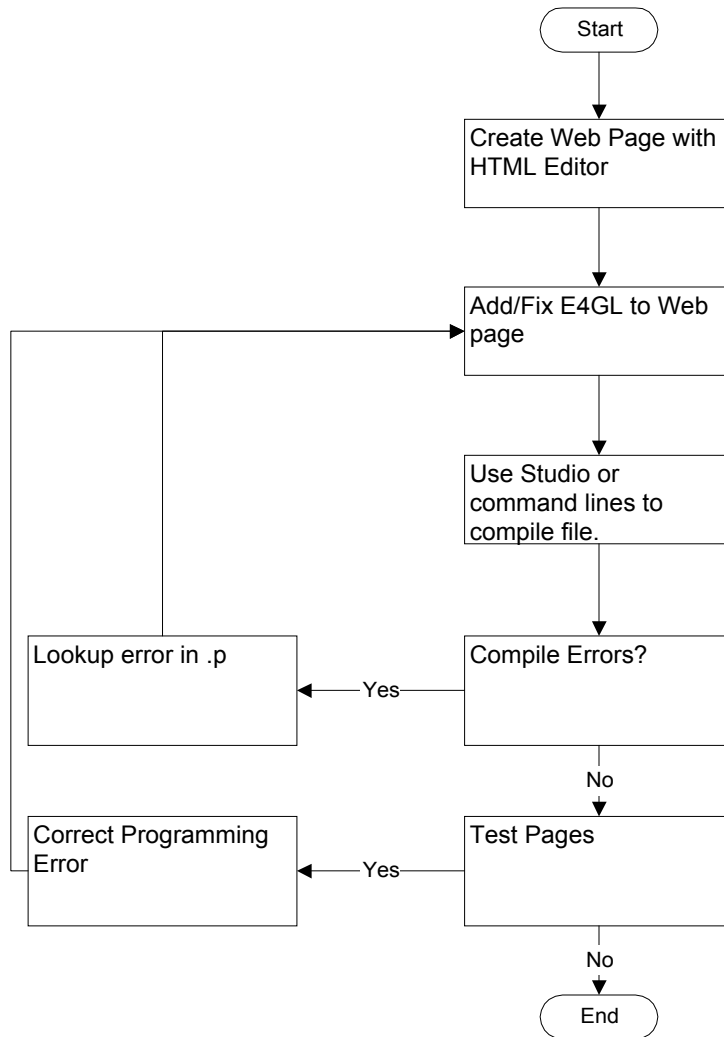
---

## Appendix B: Converting your Webspeed E4GL program to work with Blue Diamond



---

## Appendix C: Creating an E4GL program for Webspeed or Blue Diamond



---

## Appendix D: Problem Resolution Strategies

At times there will be frustrations with the application. Here are some strategies and tips to work these out.

Sometimes the application will not be able to write to a directory or file. Remember that the program is running under the User ID of the web server, and not the user. So either the web server needs to have it's UserID option changed<sup>5</sup>, or the directory permissions and file permissions need to be opened up for the web server to access them.

Sometimes a screen will appear and be totally blank. A lot of times, in the Page Source there is a Progress error of somekind showing up. Be sure to use the Page Source option on your browser to search for Progress errors on that page.

If there are progress errors that occur during rendering of the page – those are pretty obvious.

Sometimes nothing will happen – this means an error has occurred before there was any output to the CGI interface. Be sure to check the web server's error log file. A lot of times, this occurs with Progress statements failing before there is an HTML tag or the start up parameters in the bluediamond script are incorrect.

To help determine the variable settings, there is a set.ksh script in the script directory<sup>6</sup>. This script will pass back to the browser the environmental variables associated with your CGI.

---

<sup>5</sup> Remember that you can start up another instance on a different port and use that for the development environment.

<sup>6</sup> This will need to be copied over into your cgi-bin directory to be used.

---

## Appendix E: Change Log

April 20, 2002

Scott Auge : Changed the PostFix.i so it won't blow up under a non . delimited file name  
Created ProPathCmp.html that compiles everything in the environmental variable APPL\_PROPATH with results to the screen.  
Updated Studio.html to have a link to ProPathCmp.html and to line up the links a little nicer.

April 16, 2002

Scott Auge : Converted htldict tool into online dictionary viewer. Made one for application tables and one for system tables.

April 11, 2002

Mark Newnham: Ported to Windows NT. See doc/README.WINDOWS for more info.  
Paul Keary : Tables.html changed to filter out system tables. Suggest making a SysTables.html for those - put it into the TODO list.  
Scott Auge : Fixed the RUN button on the Studio to run what is clicked. Only one html can be run though!

April 1, 2002

Scott Auge: UNIX Symbolic link files are considered directories instead of normal files.

March 26, 2002

Scott Auge: Updated Error.i, as well as ClearState.p, WriteState.p, and ReadState.p to be pure Blue Diamond. ReadSession.i and WriteSession.i should work but are currently untested.  
Wrapped .i files with preprocessor checks if they have been included already (except for the \*Session.i files as they are allowed to be included multiple times.)  
Updated pop3.i to use a smaller buffer space. Tends to behave better that way.

March 13, 2002

Scott Auge: Some javascript functions for manipulating cookies. Found in javascript/cookies.js

March 12, 2002

Scott Auge: Made a switch in prgsrc/Config.i that will cause the trx svr to use the original URL-ENCODE() with 1 argument, or the progress compliant URL-ENCODE() that has 2 arguments. Set up &IF's

---

through out the code to catch these in the base programs and call appropriately.  
Also updated the MakeURL.i program so that if included more than one time, the preprocessor would not re-include it. (Bug found by Daryl Daly [daryld@norco.com](mailto:daryld@norco.com))  
Updated Documentation  
Neal Rhodes <neal@dexter.mnopltd.com> contrib1 directory. Includes code for determining if the databases are connected or programs are available.

February 27, 2002

Scott Auge: Add a default directory for Studio.  
Took out the call to a var setting script in bluediamond script for ease of set-up.

February 26, 2002

Scott Auge: Created WriteSession.i and ReadSession.i in plussrc  
Added the SelfURL variable to sga\_util.i

February 25, 2002

Paul Keary: Added LastModified to help with proxy cache'ing of dynamic web pages.  
Scott Auge: Added {&DISPLAY} for use in programs.

February 24, 2002

Scott Auge: Corrected problem in api\_e4glprs.p in which " and ' are not "tildefied" after the first set of `` are found in a line of HTML.  
Was screwing up javascript something fierce!

February 22, 2002

Scott Auge: Fixed RUN on CompileFile.html as well changes in output format.  
Scott Auge: Created MakeURL.i  
Paul Keary: Filter Files of Type on Studio.html  
Paul Keary: Default to first directory of STUDIOPATH  
Scott Auge: Propath.html and link to it on Studio.html  
Scott Auge: This is a big one. Pulled SetContent out of sga\_util.i into it's own file SetContent.i. Changed PrePendUtil.p to include this after sga\_util.i in generated files. The file sga\_util.i can be included in files that need the functions in there, but do not need the content type output automatically. One may need to regenerate all application files after install of this change. See source comments for more information. SourceLab.html puts sga\_util.i in the generated file so all functions/vars can be reached.

February 20, 2002

Changed SourceLab.html to use XMP tag instead of PRE tag for output format.  
Changed SourceLab.html so clear applies focus to source area.

---

Added a Jump Directory input to Studio.html  
Added a SetPostFix.i to the list of available functions  
When encounters WSMETA for include, will make the .p an include file.  
Changed CompileFile.html to read more clearly  
Updated Documentation

February 19, 2002

Added GET-CGI() function.  
Added CLEAR button to Source Lab  
Added RESET button to Source Lab  
Updated documentation

February 14, 2002

Paul Keary

Added STUDIOPATH variable to script controls what directories the Studio can traverse.  
Cleaned up internal directory management code.

Scott Auge

Added data dictionary browsing tools.  
Cleaned up the bluediamond script  
Updated documentation  
Added the Refresh Button to the Studio File Operations page

February 13, 2002

Scott Auge

Changed Studio so .r filtering will not squelch on 1 character file names.

Paul Keary

Defined width for table cells to keep consistent look.  
Add feature to display directory listing with multiple entries per row.  
Although I need to make it another table within the cell to keep the alignments.  
Added some <HR>'s to define the userinterface with more structure.  
Filtered out the r-code from the file listing. - TODO add check to turn this feature on/off.  
If no files are found then display No Files Found in the Selection list.  
If user hits compile, run, edit, etc with no files selected then do not perform the openwindow.

Steve Jellin

---

Added mksmall.pl to plusscript - Strips out the PUT STREAM StdOuts on consecutive lines and makes the whole set of lines one string. Shrinks .r code size for some optimizations.

February 12, 2002

Added some problem resolution methods to the documentation  
Added set.ksh to the script directory  
Created ns\_runner.p for netscape servers on OpenServer machines  
Created apache\_runner.p for apache web server  
Rewrote runner.p to determine a call to apache\_runner.p or ns\_runner.p  
Corrected problem in SourceLab.html - important code was commented out!  
Allow creation of a new file from the Studio.html page.

February 5, 2002

Started using build numbers  
Updated the documentation so that people can decipher the build number:  
        yyyy.ddd.hh.mm  
Changed the name of this file from UPDATES.TXT CHANGE\_LOG

2.3

Added Line Numbers to the ViewFile.html page. Helps in determining errors.  
Updated documentation for an Apache Web Server Install  
Updated documentation for clearer instructions on installing command line utilities  
Updated documentation for clearer installation of the CGI script

2.2.1

Added bluediamond script to the BlueDiamond/script directory. This is the CGI script used by the web server to reach the Blue Diamond transaction server.  
Removed the c directory. Blue Diamond no longer needs C.  
Removed the lib directory. Blue Diamond is Progress contained.

2.2

Changed GET-FIELD() {sga\_util.i} so that names and values don't mess up. That is, T=H&N2=TN, seems T's value is screwed by N2 having the symbol T in it.  
Changed GET-FIELD() {sga\_util.i} so that multiple pick lists will return , seperated values. Unknown if this is how Webspeed does it.  
Added studio directory. By changing PROPATH, studio can be turned on or off.  
Added Studio.html for web based file operations.  
Added CompileFile.html to compile files from the web.  
Added EditFile.html so as to edit files from the web.  
Created api\_e4glprs.p and changed e4glprs.p to call api\_e4glprs.p  
Created api\_cmp4gl.p and changed cmp4gl.p to call api\_cmp4gl.p  
Added ViewFile.html so one can read files from the web.

1.7

WARNING: Webspeed does not have the following features built in

---

---

Added ini\*.p files so that programs can have .conf files.  
Added Parm\*.p files so that programs can have database record based params.  
Added SMTP support - allows mailings from Blue Diamond Programs  
Added POP3 support - allows pulling in mail from a POP3 account  
Added State saving support.  
Added BASE64 Mime encoding support.  
Added some logged in user management tools.

1.6.6

Added mxref4gl  
Updated mkcmds.bash  
SourceLab.html fixed  
Updated Manual (seperate item from distribution)

1.6.5

Added xref4gl  
Updated mkcmds.bash



---

## Appendix F: Using Blue Diamond on Windows

Subject: RE: Blue Diamond supports Windows now  
Date: Fri, 12 Apr 2002 17:27:03 -0600  
From: Mark Newnham <mnewnham@globalbeveragegroup.com>  
To: webspeed@peg.com

The installation is designed to test your intuitivity, but heres some pointers.

1. You need perl 5 up and running on your NT box (try [www.activestate.com](http://www.activestate.com))
2. You need apache up and running. Make sure it knows where the perl installation is in httpd.conf
3. Move the bluediamondwindows.pl script to cgi-bin, edit to match your specification.
4. Read the readme.windows document in the BD distribution.
5. Take special note of the registry changes required for propath.
6. Cut the text out of the readme.windows file at the relevant point and create a perl script "cat.pl" - put it somewhere accessible.

*Editor's Note: There is a cat.pl in the script directory already*

7. Modify the runner.p program at the bottom, change it to point at your perl installation and cat.pl script.
8. Try something like  
"http://localhost/cgi-bin/bluewindowsdiamond.pl/StudioHome.html"

Start debugging...

9. The command line utilities don't work at all.....

Regards

Mark.

---

## What does support include?

Understand this support for Blue Diamond may be separate from your Progress support.

Support includes the following:

- Blue Diamond Support Bulletin
- Answers to questions about Blue Diamond problems via Email/Call back
- Immediate access to updates, the open source version will be updated once a quarter
- Immediate access to bug fixes, you will be alerted via email where to find the build
- CD-ROM of the latest build of Blue Diamond once a quarter
- Printed and bound documentation for Blue Diamond (including support bulletin)
- List of current bugs and plan to fix them
- Product Plans
- Send your enhancement requests!
- Extra source code for functionality not in the base open source product

Support is available in the United States only.

If you wish to be a support provider in another country, please contact Scott Auge at [sauge@amduus.com](mailto:sauge@amduus.com).

If you wish to be a support provider for Amduus Information Works, Inc. please contact Scott Auge at [sauge@amduus.com](mailto:sauge@amduus.com).

Contact [sauge@amduus.com](mailto:sauge@amduus.com) for pricing.

Progress licenses for use with Blue Diamond are available from Amduus Information Works, Inc. Contact Scott Auge at [sauge@amduus.com](mailto:sauge@amduus.com) for more information.

**Please register Blue Diamond! This tells me how many people are interested in it, and so then I will be interested in it! Send to [sauge@amduus.com](mailto:sauge@amduus.com):**

**Your Name**

**Your Email Address**

**Your Company Name (if applicable)**

**Thank You!**

All source sent to Amduus for Blue Diamond becomes the property of Amduus Information Works, Inc. I do this because at some point I may wish to cut CD-Roms and ask for money for

---

them and don't want a bunch of people hassling me about it. Blue Diamond will likely remain in the open source realm forever.