



Viviane Anavi-Chaput
Jay Huie
Mike Tebolt

Dimensional Insight BI Tool for Linux on zSeries

This Redpaper provides an overview of the Dimensional Insight business intelligence (BI) tool using Linux on zSeries. We demonstrate that a BI solution developed for Linux on the Intel platform could be ported to Linux on zSeries. We then examine how effectively the Dimensional Insight tool runs on a Linux guest under z/VM with an average workload.

The following topics are discussed in this Redpaper:

- ▶ Dimensional Insight products overview
 - DI-Atlantis
 - DI-Integrator
 - DI-Builder
 - DI-ProDiver
 - DI-WebDiver
 - DI-ReportDiver
 - DI-DiveLine
- ▶ Dimensional Insight test scenario
 - Workload setup
 - Workload performance

Dimensional Insight products overview

The Dimensional Insight family of products provides a comprehensive BI solution that allows information to be retrieved throughout the enterprise with speed, flexibility, and accessibility. Dimensional Insight databases are referred to as a “sphere” of data. Users who wish to query information from this database do “dives” into the data space. Dives can be done from any starting point and users can navigate to any other item for on-the-fly ad hoc analysis, without the need for complicated SQL scripts or predefined access paths.

Dives can also be predetermined for routine analysis of data. These reports can be generated and saved for other users to access allowing them to view a prescribed subset of data without the need to spend time processing the data themselves.

Figure 1 shows the interactions of the Dimensional Insight products.

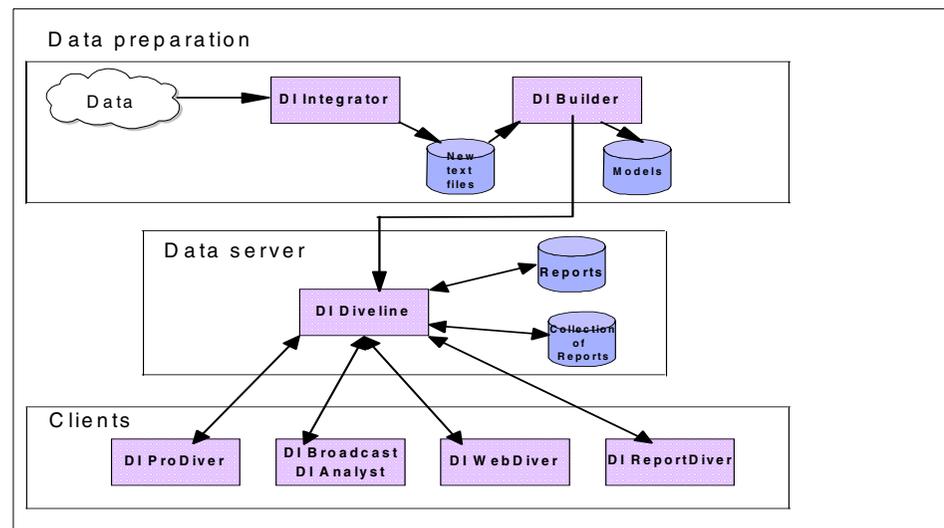


Figure 1 Dimensional Insight component layout

For the latest information on Dimensional Insight tools and solutions, direct your browser to:

<http://www.dimins.com>

The following sections give an abstract of the DI-Atlantis suite of tools.

DI-Atlantis

DI-Atlantis is the backbone of the Dimensional Insight solution. This suite of tools enables organizations to replace the traditional complicated reporting and data access methods of the past (SQL queries, RPG programming, etc.) with a much more integrated and streamlined solution for their BI needs. The DI-Atlantis tools make reports and data accessible to end users directly in a convenient and easy-to-use manner, eliminating the need for Information Services personnel to assist end users directly with data retrieval.

The DI-Atlantis suite of products is comprised of several modules which as a whole handle the extraction, transformation, processing and access to user data. Data is first translated into the Dimensional Insight sphere within which dives are done. The DI-Integrator and DI-Builder *packages* are responsible for creating the data sphere which is referred to as a *model*. Access to the model is then done using one of the many DI-Clients which connect to the Di-DiveLine server that retrieves data from the model. Dimensional Insight's DI-Diver, DI-ProDiver, DI-WebDiver, and DI-ReportDiver are all client interface options which allow end-user access to the data models through the network.

DI-Integrator

The Integrator module of DI-Atlantis acts as a preprocessor of the input data and can accept inputs from various sources, which it then combines to form flat files that are imported into the DI-Builder module. The DI-Integrator is controlled by a description file and specifies the source of the data, the operations to be performed upon it, and where to put the resulting output. Some operations the Integrator can perform are joins, table lookup, row selection and column selection. These and other functions are specified in a text file written in the Data Integrator's Object Language.

DI-Builder

The DI-Builder module accepts processed data from the DI-Integrator package and transforms this data into a model, or sphere, to be analyzed using one of Dimensional Insight's many clients. The model uses a patented data structure that the Dimensional Insight tools are optimized to use. The builder transforms the data by processing it into a more optimum structure. Most of this processing allows the DI-Builder to optimize the data layout, thereby reducing much of the burden of performing runtime analysis from the server and clients. This allows Dimensional Insight queries to be much more responsive and efficient in their access of the data.

DI-ProDiver

DI-ProDiver is a Windows-based program providing a full featured graphical interface. It allows for creating, viewing, analysing data and generating reports which can be made viewable to other Dimensional Insight users. DI-ProDiver can also be used to create models, from either text files, or an ODBC source. DI-ProDiver works with the DI-DiveLine server to perform data analysis and display. The data retrieved can be viewed and printed in many formats, including tabular format or more graphical representations.

DI-WebDiver

DI-WebDiver is a Java applet served through a Web server and accessed through a Java-capable browser. DI-WebDiver is similar in functionality to DI-ProDiver in that it allows end users to access models stored on a remote server. Users are given the freedom of accessing their data through the World Wide Web with the same intuitive interface as DI-ProDiver.

DI-ReportDiver

DI-ReportDiver is also a Java applet much like DI-WebDiver. However, DI-ReportDiver is designed to view only predefined reports. It contains the same diving engine as the other Diver Clients, but this engine is used only to display reports that were generated with ProDiver and WebDiver clients. Users access DiveBooks, which are collections of reports. These DiveBooks define how DI-ReportDiver should display the latest available data for the viewer. DI-ReportDiver supports a slim client who can display up-to-date information using a simplified GUI.

DI-DiveLine

DI-DiveLine is the data server software which enables the Dimensional Insight Business Intelligence solution to function as a cohesive whole. DI-DiveLine allows you to have a centrally controllable solution for BI data, which means there is no need to update other locations when new models or Reports are created. Likewise, neither do multiple refreshes need to be done when new data is added. As each client connects and queries the server, it will automatically access the latest data available. DI-DiveLine allows you to centrally store and control access to models, reports and specific views, making the necessary information readily available to those who need it, and protected from those who don't.

Porting and installing Dimensional Insight on Linux

In order to install the DI-Atlantis suite of tools on Linux on zSeries, we needed to build the Dimensional Insight tools which would run on Linux on zSeries.

First, we compiled the standard Dimensional Insight code with no prior modifications for our tests and produced the necessary distribution files. The Linux systems that we installed had all the necessary development tools and libraries. Most, if not all, of these tools are standard on almost all Linux installations regardless of platform or distribution, because many Linux tools are distributed in source form.

Then, we began the build process and encountered a problem only when faced with the question of byte ordering. zSeries machines are big-endian and that information needed to be set in the Dimensional Insight build configuration files with a standard C preprocessor define statement.

Once the configuration choice had been selected and the proper files modified, the build process proceeded without any further complication. It took only a short amount of time to build all the tools necessary to complete the DI-Atlantis suite. This demonstrates that for well-written code, a "port" to Linux is often just as simple as a recompile, provided the code has been written with platform considerations in mind, or is not dependent on hardware specifics.

These criteria are fairly common in the UNIX world. Code is often written to the POSIX standard, and programs which adhere to this programming specification meet with little challenge when being ported to a new platform that supports POSIX. This is especially the case with the cross-platform nature of Linux. Less than 1% of the stock Linux Kernel needed to be modified to allow it to run on zSeries hardware. This leaves very little possibility for inconsistencies in behavior between hardware platforms that are running an instance of the same Linux Kernel version that has been compiled for their hardware considerations.

From this point, the installation proceeded according to the Dimensional Insight documentation, except for a slight change in the inetd configuration. Since we used xinetd (a next generation replacement for inetd) on our Linux systems, we had to alter the installation slightly to account for this discrepancy. This was a minimal change and the procedure has been reported back to Dimensional Insight to be included in their Linux installation instructions.

Figure 2 shows the Dimensional Insight setup we did.

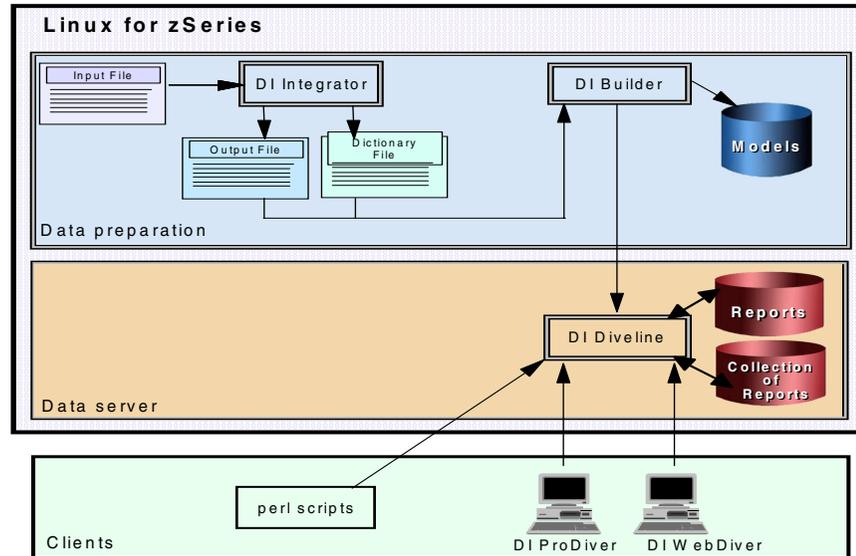


Figure 2 Dimensional Insight product setup

Because all Dimensional Insight clients use the same clear text protocol for communication, each instance of a Perl script which connects to the DI-DiveLine server represents a client connection.

The installation procedures used for these products are discussed in ,
 "DI-Atlantis installation" on page 12.

Setting up the Dimensional Insight scenario workload

It was decided that 100 active concurrent users was a reasonable amount of load for the test system to undergo. This number was selected based on the experience of the Dimensional Insight team with typical customer deployment configurations of their tools. We did not design our tests or our system configuration to maximize this number, nor did we attempt to measure optimal scalability in any empirical manner. Instead, we tried to focus on a typical middle-to-high-end range of load for the DI-DiveLine product and test the viability of running this as a guest under z/VM as a function of the stability of the setup and product.

For our testing, we did some Perl scripting around a Perl script provided by the vendor. Dimensional Insight's dives work on a clear text-based protocol, much like POP or HTTP. A graphical client will send commands in this language to the DI-DiveLine server, which performs the appropriate actions based on the commands issued. The results are sent back to the client performing the dive and can then be displayed in whatever manner the client and the user wish.

This separation of processing allows us to simulate many concurrent client requests simply by reproducing the clear text dialogue between the server and the client. The DI-ProDiver client can be used to produce reports that are saved on the server to be viewed by a client later. The DI-ProDiver GUI can also save transcripts of its actions and produce log files of the session. We used these log files as input to our Perl script in order to duplicate the actions a typical user would perform when doing dives.

Twelve different sets of typical user actions were generated and used as input to our testing. Some of these dives were simple dives only a few levels deep and some were much more complex. Reports were one of the following forms:

- Multi-crosstab: One vertical dimension, one horizontal dimension with more than one summary column
- Multi-level: Levels 2 and 5 deep are used, data joined from multiple dimensions
- Multi-tab: More than one vertical dimension, either 2 or 3 levels
- Multi-tab Multi-crosstab: Multi-tab first and then multi-crosstab, only 2 levels deep

For each of the 4 forms (6 types total) of input, we created 12 user actions which retrieved either 1 million or 10 million rows. A distribution was determined for the 12 user actions which we thought would adequately simulate a typical daily workload. A 100-entry table was created in which each activity appeared a number of times according to the distribution value we assigned to it.

Table 1 describes the 12 user actions, their selected distributions, and their approximate completion time.

Table 1 User actions with distribution and minimum completion time

User action No.	User action description	Distribution (%)	Minimum completion time
1	Multi-level, 2 level 1 M rows output	15%	10 sec
2	Multi -tab 2 summary column, 1 M rows output	15%	12 sec
3	Multi -level, 2 level 10 M rows output	10%	20sec
4	Multi-level, 5 level, 1 M rows output	10%	15 sec
5	Multi-level, 5 level, 10 M rows output	15%	21 sec
6	Multi-tab 2 summary column, 10 M rows output	15%	19 sec
7	Multi-tab 3 summary column, 1 M rows output	10%	22 sec
8	Multi-tab 3 summary column, 10 M rows output	3%	34 sec
9	Multi-cross tab, 1M rows output	5%	26 sec
10	Multi-cross tab, 10 M rows	2%	42 sec
11	Multi-tab Multicross tab, 1 M rows output	3%	53 sec
12	Multi-tab Multicross tab, 10 M rows output	2%	60 sec

Each user action was designed to simulate a user connection, which would include a recorded “wait time” of a few seconds. Thus the Dimensional Insight script would send the appropriate data, and after the response from the server, would then wait the recorded amount of time. If the server took longer during the test to send the data, then the overall time of the script would be longer; otherwise, the script would wait the predefined number of seconds.

An emulated client would make a random decision from this 100-entry table as to which test entry to run, simulating a distribution of work that would be typical of the daily activities performed by Dimensional Insight customers in their day-to-day business environment. For example, many employees might need to access a report on sales data for their work, so it was determined that this was a function which was more common than an employee who needed data for the whole year.

The Perl script was designed to moderate the number of concurrent connections seen by the server. Since the workload of the clients is light due to their lack of graphical data representation, all the connections could be run and controlled from one central system acting as multiple clients. Again, because of the separation between the DI-DiveLine server and the clients using the clear text protocol, the server performs the same amount of work whether or not the client chooses to display the results.

Workload performance

Figure 3 describes the methodology we used to generate the workload.

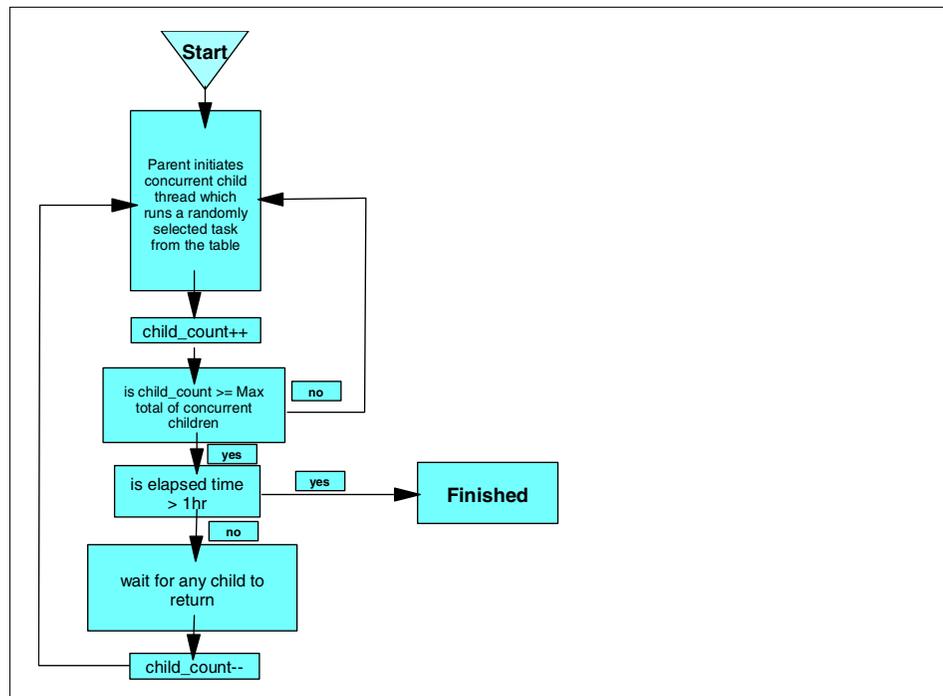


Figure 3 Methodology to generate the workload

Users were allowed to connect repeatedly over a 1-hour period for all of the tests. The rate of connections was determined by the number of active concurrent connections that were allowed, the time each dive took to complete, and the particular workload selection of the Perl scripts (which was determined by random selection).

We ran all tests with the same 100-entry table representing the predetermined distribution of work, while varying the number of simulated active concurrent clients, keeping in mind that each run's behavior was randomly determined. Table 2 describes the throughput achieved by concurrent active users.

Table 2 Active users and throughput

Active concurrent users	Approximate total reports accessed in 1 hour
60	14000
75	17500
100	24000

We measured the overall throughput with 60, 75, and 100 concurrent active users. The total throughput from each test was measured by a counter in the Perl script which was incremented after every simulated dive was started. Because the actual overall work done and the order of selections were completely random for each measurement, this provided us with approximate throughput measurements for each 1-hour period.

For example, measurements included dives that were started but not completed before the end of the hour time frame. At the completion of the hour of stress, the script was abruptly terminated. Each run was made against the Linux guest server which was running under z/VM and sharing 4 processors with the other two images, which were mostly idle. Because of the reliance on a random distribution of reports, versus a predefined order, it is virtually impossible to draw a performance-based conclusion from these results.

Figure 4 shows the projected throughput achievement of 100 active users

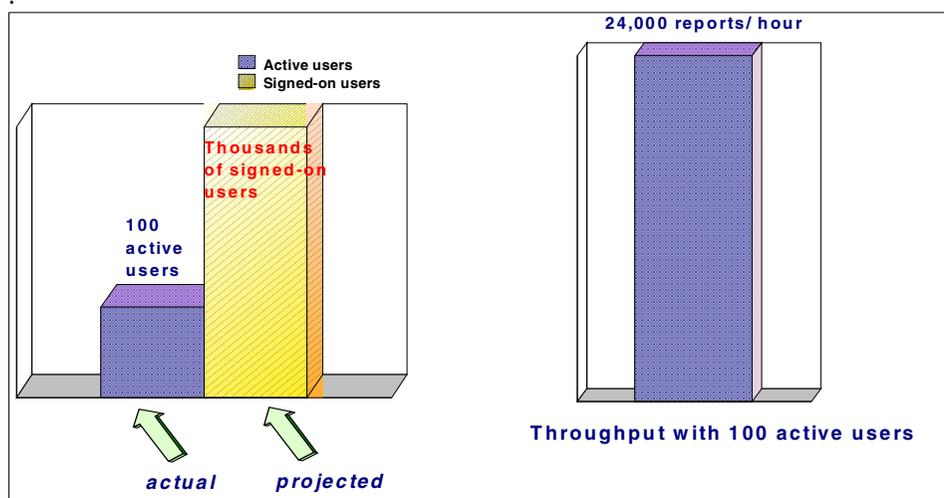


Figure 4 100 users projected throughput achievement

While not a performance-based measurement, the test shows that the DI-DiveLine server was quite effective at supporting a large number of active concurrent users on Linux on zSeries. Those users are connected to the server and are actively performing queries. In real life, it is uncommon for users to be so active in their queries. A typical user might make a few queries a minute and the connection might sit idle from the server's perspective, while the end user analyzes the data just retrieved - this is commonly referred to as "think time".

Our test scenario simulated a lower number of signed-on users (100) than would typically be logged on in a real-world scenario. Because each signed-on user was also an active user, our test cases generated more processing work in a given amount of time than is typically seen in a production environment. We assume that this situation translates into thousands of signed-on users in a typical production environment.

Linux on zSeries was quite capable of handling the workload with little or no tuning of the environment. The fact that this solution was more than capable of supporting a high rate of active user connections is an indication that Linux on zSeries is a viable business platform on which one can rapidly deploy a stable and robust BI solution.

In spite of the approximate number of active user connections over a 1-hour period, this test is a useful demonstration of the stability of a BI application which was ported from the Linux on Intel platform to the Linux on zSeries platform with a simple recompile.

DI-Atlantis installation

To install the DI-Atlantis package, follow these steps.

1. Unpack DI_Atlantis-pkg.tar:

```
[diuser@biserverbiserver DI_Atlantis]$ ls -l
/home/diuser/unpack-files/DI_Atlantis/DI_Atlantis-pkg.tar -rw-rw-r-- 1
diuser diuser 63354880 Oct 2 21:07
/home/diuser/unpack-files/DI_Atlantis/DI_Atlantis-pkg.tar
[diuser@biserver DI_Atlantis]$ tar -xf
/home/diuser/unpack-files/DI_Atlantis/DI_Atlantis-pkg.tar
[diuser@biserver DI_Atlantis]$ ls
DI-Demos      MapData      ReleaseNotes  Utilities  shared
Documentation OldVersions  Software      bin
```

2. Copy files to /home/DI_Atlantis/bin:

```
[diuser@biserver DI_Atlantis]$ cd bin
[diuser@biserver bin]$ cp ../Software/integ-2.1.5b-linux integ
[diuser@biserver bin]$ cp ../Software/builder-3.0.17 builder
[diuser@biserver bin]$ ls
README builder integ
```

3. Add /home/DI_Atlantis/bin to PATH:

```
diuser@biserver DI_Atlantis]$ export PATH=$PATH:/home/DI_Atlantis/bin
```

Installing DI-DiveLine

To install DI-DiveLine, follow these steps.

1. Unpack the tar file with the DI-DiveLine package:

```
[diuser@biserver DI_Atlantis]$ ls Software/di-diveline-4.3.8-linux.tar
Software/di-diveline-4.3.8-linux.tar
[diuser@biserver DI_Atlantis]$ tar -xf Software/di-diveline-4.3.8-linux.tar
[diuser@biserver DI_Atlantis]$ ls
DI-Demos      MapData      ReleaseNotes  Utilities  diveline
Documentation OldVersions  Software      bin        shared
[diuser@biserver DI_Atlantis]$ ls diveline
INSTALL README bin cgi-bin docs html install-di-diveline samples
webdir
```

2. Add /home/DI_Atlantis/bin to PATH:

```
[diuser@biserver DI_Atlantis]$ export PATH=$PATH:/home/DI_Atlantis/bin
[diuser@biserver DI_Atlantis]$ cd diveline
[diuser@biserver diveline]$ ls
INSTALL README bin cgi-bin docs html install-di-diveline samples
webdir
[diuser@biserver diveline]$ ls
```

```

INSTALL README bin cgi-bin docs html install-di-diveline samples
webdir
[diuser@biserver diveline]$ ls -l
total 40
-rw-rw-r-- 1 diuser diuser 3955 May 2 2000 INSTALL
-rw-rw-r-- 1 diuser diuser 735 Sep 28 05:57 README
drwxrwxr-x 2 diuser diuser 4096 Oct 2 21:15 bin
drwxrwxr-x 2 diuser diuser 4096 Oct 2 21:15 cgi-bin
drwxrwxr-x 2 diuser diuser 4096 Oct 2 21:15 docs
drwxrwxr-x 2 diuser diuser 4096 Oct 2 21:15 html
-rw-rw-r-- 1 diuser diuser 4407 Aug 17 00:14 install-di-diveline
drwxrwxr-x 2 diuser diuser 4096 Oct 2 21:15 samples
drwxrwxr-x 2 diuser diuser 4096 Oct 2 21:15 webdir

```

3. Install DI-DiveLine:

```

[diuser@biserver diveline]$ chmod 775 install-di-diveline
[diuser@biserver diveline]$ ./install-di-diveline
Welcome to the DI-DiveLine installation script.
This install script will first ask you a series of questions.
At the end, it'll ask you if you want to go ahead and do the install.
Nothing will be modified until after that point.
Defaults for questions will appear in square brackets []. Hitting
carriage return will automatically choose those defaults. If you
don't know what to do, trust the defaults.
To abort the installation, hit Control-C.
Verifying current directory...
Found installation files.
Where should DI-DiveLine store configuration files and temporary files?
[/usr/local/lib/diveline] /home/DI_Atlantis/diveline
DI-DiveLine can run at 3 security levels:
  Level 0: No security checking (all users have access to all models
           and divebooks in the models directory).
  Level 1: Security checking based on web login.
           If a model is not listed in the config file, then ALL users
           have access to it.
  Level 2: Security checking based on web login.
           If a model is not listed in the acl file, then NO users
           have access to it.
What security level should DI-DiveLine run at? [0]
Install sample model and divebook? [y]

OK. Go ahead and install? [y]
Integer "security_level": 0
0
Max Users: 20
0
Authentication Type: own
0
Debug Level: 0

```

```

0
0
0
0
0
[diuser@biserver diveline]$

```

- As root, add the following line to /etc/services to specify which port DI-DiveLine will listen to:

```
di-diveline    2130/tcp                # DI-DiveLine
```

- Add file /etc/xinetd.d/di-diveline with contents to specify the di-diveline executable which runs at the di-diveline port, and the dataroot directory which DI-DiveLine uses for its configuration:

```

service di-diveline
{
    flags          = REUSE
    socket_type    = stream
    wait           = no
    user           = root
    server         = /home/DI_Atlantis/diveline/bin/di-diveline
    log_on_failure += USERID
    disable        = no
    server_args    = -dataroot /home/DI_Atlantis/diveline
}

```

- As root, edit /etc/xinetd.conf and change or set the “instances” variable to a value greater than the maximum number of concurrent connections allowed.
- Restart the xinetd daemon, so that the DI-DiveLine program will listen at the specified port in /etc/services file. Also set the instances number to how many users you want to allow to connect concurrently in the /etc/xinted.conf file:

```

[root]% killall -USR1 xinetd
[root]% exit

```

- Log back in as the non-admin user and connect to DI-DiveLine through a telnet session to test the connection:

```

[diuser@biserver diveline]$ telnet 13.2.22.191 2130
Trying 13.2.22.191...
Connected to bilin.pok.ibm.com (13.2.22.191).
Escape character is '^'.
version
diuser1
4.3 (8)
0
file_list
1
demo-divebook.dbk
1

```

```
demo-sales.mdl
1
demo-dairy/
0
quit
Connection closed by foreign host.
[diuser@biserver diveline]$
```

9. Add an administrator and a test user:

```
[diuser@biserver diveline]$ export DIDATA_DIR=/home/DI_Atlantis/diveline
[diuserdiuser@biserver diveline]$ cd bin
[diuser@biserver bin]$ ./dicfg add user -user admin -password admin
-admin_flag TRUE
0
[diuser@biserver bin]$ ./dicfg add user -user test -password test
-admin_flag FALSE 0
```

Installing Web components

Dimensional Insight has two Web components:

- ▶ DI-WebDiver
- ▶ DI-ReportDiver.

Installing DI-WebDiver & DI-ReportDiver

To install the components, follow these steps.

1. Unpack the WebDiver and ReportDiver packages:

```
[diuser@biserver DI_Atlantis]$ tar -xf Software/webdiver-4.3.5.tar
[diuser@biserver DI_Atlantis]$ tar -xf Software/reportdiver-4.3.5.tar
[diuser@biserver DI_Atlantis]$ ls
DI-Demos      MapData      ReleaseNotes Utilities  diveline    shared
Documentation OldVersions Software     bin       reportdiver webdiver
[diuser@biserver DI_Atlantis]$ pwd
/home/DI_Atlantis
```

2. Add the html directory to the webserver's path:

This involves either setting up an aliases in your webserver's configuration file or moving the html directory provided by Dimensional Insight to a path accessible to the server. For our test we simply moved /home/DI_Atlantis/webdiver/html to /var/www/webdiver where our apache server would then distribute the Dimensional Insight index.html file which provides a link to the java DI-WebDiver.

```
[diuser@biserver DI_Atlantis]$ mv ~/diuser/DI_Atlantic/webdiver/html
/var/vvv/html/webdiver
```

And the same procedure for ReportDiver;

```
[diuser@biserver DI_Atlantis]$ cp -R ~diuser/DI_Atlantic/reportdiver/html
/var/vvv/html/reportdiver1s
```

Creating data extracts

At the time of writing, DI-Atlantis does not support ODBC connectivity to relational databases on Linux. Only ASCII text files are supported at this time.

If data resides on relational DBMS, you need to extract this data and put it into ASCII text files. In our test scenarios we did not investigate data extraction issues. We used sample data provided by Dimensional Insight in flat files.

Building models

We used the sample models and sample data provided with the DI-Demos package. We followed these steps.

1. Unpack DI-Demos:

```
[diuser@biserver data]$ pwd
/home/DI_Atlantis/diveline/data
[diuser@biserver data]$ mkdir DI_Demos
[diuser@biserver data]$ cd !$
cd DI_Demos
[diuser@biserver DI_Demos]$ tar -xf
/home/diuser/unpack-files/DI_Demo-dist.tar
```

Alternatively, if these are unpacked in a different directory, then a link can be made to that directory in the /home/DI_Atlantis/diveline/data/DI_Demos directory.

2. Build a sample model:

Simple test models can be built using sample data provided in the DI-Demos package.

The sample DI-Demos package includes data files (source data and dictionary files), integrator scripts, a builder description file, and a shell script to run through these. This will produce a model, dist.mdl, which is a simple sales/distribution data model:

```
[diuser@biserver DI_Benchmark]$ ls
/home/diuser/unpack-files/DI_Benchmark/DI_Benchmark.tar
/home/diuser/unpack-files/DI_Benchmark/DI_Benchmark.tar
[diuser@biserver DI_Benchmark]$ tar -xf
/home/diuser/unpack-files/DI_Benchmark/DI_Benchmark.tar
[diuser@biserver DI_Benchmark]$ ls
```

```

Data Logs Models Programs Temp
[diuser@biserver DI_Benchmark]$ cd Programs/
[diuser@biserver Programs]$ ls -l
total 28
-rw-rw-r-- 1 diuser diuser 84 Sep 30 10:02 README
-rw-rw-r-- 1 diuser diuser 1671 Sep 30 09:26 customer.int
-rw-rw-r-- 1 diuser diuser 432 Sep 30 09:26 dist.dsc
-rw-rw-r-- 1 diuser diuser 2419 Sep 30 09:26 dist.int
-rwxrwxr-x 1 diuser diuser 497 Sep 30 09:26 makedist
-rw-rw-r-- 1 diuser diuser 414 Sep 30 09:26 makedist.bat
-rw-rw-r-- 1 diuser diuser 2381 Sep 30 09:26 product.int

[diuser@biserver DI_Benchmark]$ ls -R
.:
DI_Benchmark.tar Data Logs Models Programs Temp ./Data:
README company.dic product.dic salesperson.dic supplier.dic
company.dat product.dat salesperson.dat supplier.dat ./Logs:
README ./Models: README ./Programs:
README customer.int dist.dsc dist.int makedist makedist.bat
product.int ./Temp: README
Run with only the first 1000 rows generated for dist.txt
[diuser@biserver Programs]$ ./makedist -first 1000
DI-Atlantis Data Integrator v2.1 (5)
Copyright (C) 1991-2001 by Dimensional Insight, Inc
Processing customer.int...
Starting task customer...
30000 records written to file ../Temp/customer.txt.
DI-Atlantis Data Integrator v2.1 (5)
Copyright (C) 1991-2001 by Dimensional Insight, Inc
Processing product.int...
Starting task product...
3671 records written to file ../Temp/product.txt.
DI-Atlantis Data Integrator v2.1 (5)
Copyright (C) 1991-2001 by Dimensional Insight, Inc
Processing dist.int...
Starting task Distrib...
1000 records written to file ../Temp/dist.txt.
Building ../Models/dist.mdl from dist.dsc
Tue Oct 2 22:06:44 2001 Starting phase 1...
Tue Oct 2 22:06:44 2001 Starting phase 2...
Input information:
Num input records = 1000
Key information:
Key Company, num values = 3, max clump = (399)"Omega Brands"
Key Ship Month, num values = 12, max clump = (99)"November"
Key Customer Premise, num values = 2, max clump = (792)"off"
Key Customer, num values = 982, max clump = (2)"Allen Teri Pope"
Key Salesperson, num values = 143, max clump = (16)"House Account"
Key Supplier, num values = 132, max clump = (197)"DISCONTINUED (059)"

```

```

Key Product Class, num values = 6, max clump = (496)"WINE"
Key Brand, num values = 384, max clump = (25)"DEKUYPER"
Key Product, num values = 856, max clump = (5)"FARNESE TREBBIANO V - 1.5 LT
(01259)"
Tue Oct 2 22:06:44 2001 Starting phase 3...
Tue Oct 2 22:06:44 2001 Starting phase 4...
Detail information:
DtI row size = 28, bucket size = 1000

Tue Oct 2 22:06:44 2001 Starting phase 5...
Tue Oct 2 22:06:44 2001 Starting phase 6...
Tue Oct 2 22:06:44 2001 Phase 6 -- 100% done
Tue Oct 2 22:06:44 2001 Build done
[diuser@biserver Programs]$

```

3. Link the Models directory back to the DiveLine data tree:

```

[diuser@biserver Programs]$ cd /home/DI_Atlantis/diveline/data/
[diuser@biserver data]$ ln -s /home/DI_Benchmark/Models Benchmark

```

4. Using ProDiver, do **File->Open, Benchmark/dist.mdl**. Then reports can be created, and saved as marker files.

Creating reports

Using the DI-Diver, open the models that you created, by using **File->Open**. When you open the DI-Diver for the first time, the application will prompt for the DataServer IP address. Give IP address of the DI-Integrator data server. Select one of the Dives from the console window.

Now, from the model file, select the interrelated dimensions that you want to be included in the report. Once you have selected your dimensions, save the report through **File->Save Marker**. Save the report to the DI-DiveLine server as a .MRK file. These reports can be accessed by the clients connected through the LAN through DI-ProDiver.

To create reports that have to be accessed via a browser, a user has to follow the same steps as described above for the DI-WebDiver. Such reports can be accessed either through a browser or a DI-ReportDiver.

Users accessing reports

Users can access the reports either through DI-WebDiver or DI-ReportDiver. For example, we accessed the reports by pointing the browser to this URL:

<http://biserver/webdiver/index.html>
<http://biserver/reportdiver/index.html>

Users can also view the reports through DI-Diver by connecting to the DiveLine data server.

This document created or updated on January 25, 2002.



Comments may be addressed to:
 IBM Corporation, International Technical Support Organization
 Dept. HYJ Mail Station P099
 2455 South Road
 Poughkeepsie, NY 12601-5400

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

Redbooks (logo)[™]  IBM

Other company trademarks

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.