The Progress Electronic Magazine

In this issue:

Publisher's Statement:

!!!WANTED!!!
Amduus Information Works, Inc. is looking for consultants to resell access to our web based software called Service Express. We will need you to find companies who would want use of this software, to configure the software to their needs, and to support them in the use of the software. The software is rented out – no licenses are sold. Each month, you would receive a portion of the revenue, as well be able to bill for training and support – modeled like an insurance agency. Contact sauge@amduus.com for more information.

EXTRA! EXTRA! EXTRA! Your company has been cracked!

It has finally happened. Someone managed to use an operating system/programming flaw to log into your database and perform queries upon queries of sensitive data entrusted to you by your customers.

Sometimes it is as simple as hacking ODBC on MS' IIS. Worse someone has made it into the network and has pulled back-ups of your database for free browsing!

How can you protect yourself from someone logging into your database via an ODBC "password flinger" and having free reign of the data? How can you protect yourself from the collection of viruses that would like to make use of your Windows computer to go wandering around your data?

Read on!

Alas, I have finished up a contract with the Superior Court of California and am available once again to do work. If any of you have some work that can be done in the United States or via the wire, take a gander at my resume: http://www.amduus.com/Resumes/ScottAuge.html. I tend towards web based applications on UNIX/Linux operating systems.

Also, money is getting a wee-bit tight these days in this technology jobs depression. I would like to ask the readers of this publication to send in a donation of maybe $5.00, $10.00 or even $20.00 to aid in the continuation of this publication. Please send these donations to:

Scott Auge
1818 Briarwood
Flint MI 48507

or

Log into www.paypal.com and
send money to scott_auge@yahoo.com via your credit card!
Better yet – simply order the CD-ROM available on the last page… Thanks!

Coding Article: Encrypting Data In Your 4GL Application and Database

By Scott Auge

Your database is being accessed from the wilds. Usually that refers to the internet – where there are people working on connections to your machines to achieve something. Sometimes it is to place an order. Sometimes it is to pull all the credit card numbers out of your system!!!

Yikes! Let's hear some lore:

http://www.internetnews.com/dev-news/article.php/10_1013341

> *It also appears the e-mail addresses and full names of Midwest Express customers have been compromised with the screenshot, which one security expert said, "seemed legitimate, and not just a manipulated image map."*

http://news.com.com/2100-1002-994821.html?tag=fd_top

> *Online intruders broke into a server containing the credit card numbers of some 57,000 patrons of a Georgia Institute of Technology arts and theater program, a university official said Monday.*

http://www.bday.co.za/bday/content/direct/1,3523,1293972-6129-0,00.html

> *CONSUMER confidence in online credit card transactions may take a dip after hackers gained access to more than 8-million MasterCard and Visa account numbers last week. The hackers compromised computers run by a third party firm that processes credit card transactions on behalf of US banks. Early investigations showed that none of the cards had been used fraudulently.*

That's really gotta suck to tell your customers YOU just gave their credit card numbers away.

What we are trying to prevent:

There are so many ways to break into a computer. It is just plain sad.

A lot of people are confused by the "silver bullet" of a firewall. "But we have a firewall!" they exclaim. Well – that firewall doesn't do ditty for you when someone attacks the web server software it's self (as in Code Red for MS IIS) nor does it do ditty when it goes after the database

it's self (SQL Slammer for MS SQL Server.) Blocking the ports via firewall doesn't accomplish much – after all the whole idea is for people on the internet to interact with the system![1]

Lets say you have an ODBC connection hanging out there somewhere. Especially on a Windows computer – in order for it to interact with a database, it needs a Data Source. Guess what – crackers can list the names of Data Sources, write programs to access it, and start throwing userid's and passwords at it.

Progress uses a different mechanism, but a couple of quick looks on the desktop shortcuts will tell a trouble maker how to connect to a database.

JDBC pretty much falls in the same category as ODBC.

I don't want to get into a lengthy description of cracks. What I do want you to know is that your very own software can be used against you. If it thinks someone accessing it is who should be accessing it – it will do as they please! Especially report programs. People internal to the company may be accessing tables with sensitive data on them. Do you want that?

Most attacks happen on the underlying software of an application – the middleware and glue that puts things together. In other words, it's not your software, it's the data source your software uses. It's not your web pages, it is the ASP weaknesses used in MS IIS that are used. Keep up on those patches!

Types of data that should be encrypted:

[1] Sure for internal systems you can specify which IPs can interact with a database hidden away on an internal network – but the fact is that database is accessed by some kind of software – and you guessed it – it is that software the crackers go after!

Some types of data to be encrypted are obvious. As a general rule – anything that can be used to identify a customer in your system should be encrypted. This includes:

- Passwords
- Names (provided they are not searchable – see below)
- Address Information (provided it is not searchable – see below)

- Purchase Authorization Codes

- Credit Card Numbers
- Credit Card Expiration Date
- Credit Card Auth Code

Next might be sensitive materials – such as:

- Paycheck amounts
- Salary amounts
- Commission amounts

Followed by perhaps confidential information – active quotes, strategy, etc.

A possible solution:

There are a couple of strategies one can use to encrypt software. They do depend on the software that you have available.

Overview of GPG/PGP:

This article will be working with gpg – an open source variant of pgp. I like both of them – though the gpg one seems to travel on more operating systems (probably that open source thing.)

Both these packages are public key crypto-systems. What that means is there are two keys. One, called the public key, pretty much gives the software the information for encrypting something, but not for decrypting it. The other, called the private key, will decrypt what ever it's corresponding public key has encrypted.

Very convenient, because no passwords are needed. Passwords have a tendency to be simple-thereby open to dictionary attacks, and worse - easily spread by gossip. With a public key, you can share it with everyone, and not worry about any one passing it along to other people. It's sole purpose is to encrypt messages/data.

Here is how we created a key set:

```
[~]$ gpg --gen-key
gpg (GnuPG) 1.0.1; Copyright (C) 1999 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

Please select what kind of key you want:
   (1) DSA and ElGamal (default)
   (2) DSA (sign only)
   (4) ElGamal (sign and encrypt)
Your selection? 1
DSA keypair will have 1024 bits.
About to generate a new ELG-E keypair.
              minimum keysize is  768 bits
              default keysize is 1024 bits
    highest suggested keysize is 2048 bits
What keysize do you want? (1024)
Requested keysize is 1024 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0) 5y
Key expires at Sat Jul 26 20:38:26 2008 PDT
Is this correct (y/n)? y

You need a User-ID to identify your key; the software constructs the user id
from Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: Scott Auge
Email address: sauge@amduus.com
Comment: Test Key
You selected this USER-ID:
    "Scott Auge (Test Key) <sauge@amduus.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
You need a Passphrase to protect your secret key.

We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
.+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++..+++++.+++++++++++++++.++++++++++++++++++++.....>+++++.......
....+++++
public and secret key created and signed.
```

Here is an example of a public key:

```
[~]$ gpg --armour --export sauge@amduus.com

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.0.1 (GNU/Linux)
Comment: For info see http://www.gnupg.org
```

```
mQGiBDxGjX0RBADUsG65LNF3gSqSzNZRHL52J2UAKQfZuwHzvcA8BzqOg/qQilRd
Z63qzjKBgnlLYbXIgFWIZmEH6HRq9VAPXr6ftwJdne7btKi94hMFMh2/GWXsDObm
YKZgpPfoqD4jSlPnGOfMBuEl+xK/vnyBN+Gk+qbNVZPCOsAOsfx0jP3fnQCg/wR1
6AuohTh48ipEIZ33vp43visEAMMQt8sKCZ8zT9NuJea6PngATaVkZz6KssMpcghN
QIrT6gmpEUHF1O7o0PzyB6mkpEmH2GHdaqzJj3LNDdj/J9oUCSUgNLa//S8aKgJl
3XZNoJqWG4xAeo5JUm2v4UB7GLe7X7xraRcpJhx0oKG5xfrOxSKFKjn7v21MqEni
V/XBA/41yIKUvKUnDIUABQ8s5emoULQ5ZjTa9uTxGJQdSuwmjDD7J9uifMJHSsBb
ZMLirLQPmb2EmdhmF73GiqkGKoZAU9Y9krf5rptdlsBfS3U6hiFtyHjW+lRSV46R
Mcr3nOtN5ed7LC7cuar3MTvZdKT3LIeBkKWOUKayQRRGSG8PhLQdU2NvdHQgQXVn
ZSA8c2F1Z2VAYW1kdXVzLmNvbT6ITgQQEQIADgUCPEaNfQQLAwIBAhkBAAoJEC9r
f8xZo6kzQmMAoL7WPETdyG4lRvFsOaoGqvtethTtAKC3sJ2OCw6Q8hn1xKwOdWzQ
O4NHx7kCDQQ8Ro19EAgA9kJXtwh/CBdyorrWqULzBej5UxE5T7bxbrlLOCDaAadW
oxTpj0BV89AHxstDqZSt90xkhkn4DIO9ZekX1KHTUPj1WV/cdlJPPT2N286Z4VeS
Wc39uK50T8X8dryDxUcwYc58yWb/Ffm7/ZFexwGq01uejaClcjrUGvC/RgBYK+X0
iP1YTknbzSC0neSRBzZrM2w4DUUdD3yIsxx8Wy2O9vPJI8BD8KVbGI2Ou1WMuF04
0zT9fBdXQ6MdGGzeMyEstSr/POGxKUAYEY18hKcKctaGxAMZyAcpesqVDNmWn6vQ
ClCbAkbTCD1mpF1Bn5x8vYlLIhkmuquiXsNV6TILOwACAggA5sIEM+Sd7G9odYOc
ZLE7yYCWsIT1NUbWBSU2oANebsoApkIdIr95d83iURAuUc6dOetLhyoA8qo1dzlH
KMbfCUh/MNjcp0CMiLTtKfi/hD/G75CIL3KeTP/A1ymRLg1XE0zoDTWj3M4dDQln
nAs4lzTl6FFdbFjxNjrLShepDTJzIxuMs5YP00yQx0n1kUKiqQ1Z/9OpxJKe1bXQ
o0RPDFa8zzPnxhtz5qhBojOUZhFHk2Rc7A6cPzc5VHnlzfarfu/6wHZAmZ3liYrj
8pX1i032Qg0e3lH/r09EKX2qGXlObHrQiSA7UQd/r7ao9QKJO33N83xzuW7zV4Jb
/z55K4hGBBgRAgAGBQI8Ro19AAoJEC9rf8xZo6kz8OsAn2JG0pwLyz+tibiKvWfJ
XTCnXFIJAJ9PrZq44BQez04h0VANpMb94LY26A==
=bd32
-----END PGP PUBLIC KEY BLOCK-----
```

Your probably thinking “Wow – that is a lot of numbers and letters to remember.” So – how do we keep track of all these keys and stuff. The good news is that public key crypto-systems have “keyring” software included in them. By using commands, you can add, delete, export, and import files containing all the gobble gook found above.

Lets see how we go about encrypting something.

Lets say we want to encrypt – something that looks like a credit card number:

```
5555-987-7654
```

How do we do this on the command line?

First we place that data into a file called testdata. Then we issue the following command:

```
[~/code/progress/ezine35]$ gpg -a -r sauge@amduus.com --yes -e testdata
```

This creates a file called testdata.asc which reads as:

```
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.0.1 (GNU/Linux)
Comment: For info see http://www.gnupg.org

hQEOA2IWBurjayzyEAQAhEocpk8qFQRQrnnyF+fx4oDjtAOy/BARtV6GgZlksZaf
yIOypJJS5ajQPdtJHcVSZeH8XUOkbcxkXLYbRIEUOXHTdljj5fhjU3blVnnvjlR6
```

```
qSn6cMhEbdWgAhA2t4WOsTerlDvrWZE6a3rdvdyfRp6ckPkZuR4CqtI0a3jdcfIE
ALQ7WXUbqsPp2Z3PYmSN1EH79BL37iSWAus8bEd3UCahSMG9wmXJIFAIN2j6pHdr
lDK5WvY4/+H6y/wSr1Su4jhDqmjuzCCsTR7fIp6BemEzCrJXmFfOJv9HJS+zeb5u
bwBqmRcg2cRuJyOBSjHzuKxSrk978Q0nwjIane3PsrWKyTBfHMJqHt1vfjlLG2S+
I7tj6jTf0UYK6ZiIJd2Xj9eiI089hHyNTIA3lHnub3QBEpw=
=v4Ir
-----END PGP MESSAGE-----
```

As you can tell – encrypting your data is going to create a substantially larger piece of data to store.

How do we decrypt such a thing?

```
[~/code/progress/ezine35]$ gpg -a -r sauge@amduus.com -o testdata.plain --yes
-d testdata.asc

You need a passphrase to unlock the secret key for
user: "Scott Auge (Test Key) <sauge@amduus.com>"
1024-bit ELG-E key, ID E36B2CF2, created 2003-07-29 (main key ID 4D596895)

Enter passphrase:

[~/code/progress/ezine35]$
```

This creates a file called testdata.plain which contains:

```
[~/code/progress/ezine35]$ cat testdata.plain
5555-987-7654
```

Our original data we encrypted.

See Appendix B of the article for how to call gpg with stdin/stdout commands.

Be sure to read the conclusion of the article for things that should be done for implementation.

Use one key pair:

As you can see above – one can make a single key pair, perhaps name it GobblyGook, and use it to encrypt and decrypt data. Then just store the encrypted goods into the database.

It is very simple and pretty effective.

There is a need of a pass phrase to decrypt data. One can make it a parameter, or allow certain users to enter it to access that level of data. If they don't have the right pass phrase, all they will get on their screens is encrypted alphabet soup.

You can "compartmentalize" data by encrypting it with different pass phrases. Certain fields regarding credit card numbers will be encrypted with one key pair that only Bob and Cathy know the pass phrase to. Other fields involving salary information is encrypted with a different key pair that only John and Isis know the pass phrase to.

Should one of these people move on – the pass phrase can be changed in the key ring – and the new one redistributed to the people who should know it.

Use multiple key pairs:

It is possible to have multiple public keys be used to encrypt text. That way, the owners of the pass phrase to the private key can decrypt the message. This gets a little more complicated with key management, but I am writing this to let you know that it is possible.

Encrypt.p

Below is a program used to encrypt the contents of a progress variable or database field.

It takes four arguments:

Input The data to be encrypted
Input The key name
Output The encrypted data

```
/*
 * Removal of this header is illegal.
 * Written by Scott Auge scott_auge@yahoo.com sauge@amduus.com
 * Copyright (c) 2002 Amduus Information Works, Inc.  www.amduus.com
 *
 * 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 Information Works, Inc. 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.
 *
 */

DEF VAR RCSVersion AS CHARACTER INIT "$Header:
/home/sauge/code/progress/ezine35/RCS/Encrypt.p,v 1.2 2003/07/29 21:13:17 sauge Exp sauge
$" NO-UNDO.

/* Call out to gpg to do a decrypt of data */
/* WARNING:  Some of the varibles are backwards because this is a simple */
/* copy of Decrypt re-worked to encrypt.                                 */

DEF INPUT  PARAMETER cEncryptData AS CHARACTER NO-UNDO.
DEF INPUT  PARAMETER cKeyName     AS CHARACTER NO-UNDO.
DEF OUTPUT PARAMETER cPlainData   AS CHARACTER NO-UNDO.

DEF VAR cEncryptFileName AS CHARACTER NO-UNDO.
DEF VAR cPlainFileName AS CHARACTER NO-UNDO.

/* Our defaults - might want to make them else wehere */

ASSIGN cEncryptFileName = "/tmp/testdata.asc".
ASSIGN cPlainFileName = "/tmp/plain.txt".

/* Take varible containing encrypted text and place it into a file. */
/* I hate doing this, but it is mostly demo code - all this should  */
/* happen via stdin and stdout.                                     */

OUTPUT TO VALUE(cEncryptFileName).
PUT UNFORMATTED cEncryptData.
OUTPUT CLOSE.


/* Convert that file into a plaintext file */
/* Note we do some funky stuff to get the passphrase into the string via the --
passphrase-fd */
/* We don't want the pass phrase on the command line which can be seen with ps -aux */

OS-COMMAND  gpg -a --batch --no-tty -r VALUE(cKeyName) -o VALUE(cPlainFileName) --yes -e
VALUE(cEncryptFileName) .

/* Load the output varible with the file's contents */
```

```
RUN ReadTextFile.p (INPUT cPlainFileName, OUTPUT cPlainData).

/* Clean house */

OS-DELETE VALUE(cPlainFileName).
OS-DELETE VALUE(cEncryptFileName).
```

The testing program:

```
DEF VAR cEncryptData AS CHARACTER NO-UNDO.
DEF VAR cPlainData   AS CHARACTER NO-UNDO.

RUN ReadTextFile.p (INPUT "testdata", OUTPUT cEncryptData).

RUN Encrypt.p (
INPUT  cEncryptData ,
INPUT  "sauge@amduus.com",
OUTPUT cPlainData
).

OUTPUT TO /tmp/a.
PUT UNFORMATTED cPlainData.
OUTPUT CLOSE.
```

One can certainly set a variable to a value to be encrypted and then pass that into the Encrypt.p program.

Decrypt.p

This program uses the same arguments as Encrypt, only the plain text and the encrypted text are reversed.

```
/*
 * Removal of this header is illegal.
 * Written by Scott Auge scott_auge@yahoo.com sauge@amduus.com
 * Copyright (c) 2002 Amduus Information Works, Inc.  www.amduus.com
 *
 * 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 Information Works, Inc. 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.
 *
 */

DEF VAR RCSVersion AS CHARACTER INIT "$Header:
/home/sauge/code/progress/ezine35/RCS/Decrypt.p,v 1.1 2003/07/29 06:46:48 sauge Exp sauge
$" NO-UNDO.

/* Call out to gpg to do a decrypt of data */

DEF INPUT  PARAMETER cEncryptData AS CHARACTER NO-UNDO.
DEF INPUT  PARAMETER cKeyName     AS CHARACTER NO-UNDO.
DEF INPUT  PARAMETER cPassPhrase   AS CHARACTER NO-UNDO.
DEF OUTPUT PARAMETER cPlainData   AS CHARACTER NO-UNDO.

DEF VAR cEncryptFileName AS CHARACTER NO-UNDO.
DEF VAR cPlainFileName AS CHARACTER NO-UNDO.

/* Our defaults - might want to make them else where */

ASSIGN cEncryptFileName = "/tmp/testdata.asc".
ASSIGN cPlainFileName = "/tmp/plain.txt".

/* Take varible containing encrypted text and place it into a file. */

OUTPUT TO VALUE(cEncryptFileName).
PUT UNFORMATTED cEncryptData.
OUTPUT CLOSE.


/* Convert that file into a plaintext file */
```

```
/* Note we do some funky stuff to get the passphrase into the string via the --
passphrase-fd */

input-output thru  gpg -a --batch --no-tty --passphrase-fd 0 -r VALUE(cKeyName) -o
VALUE(cPlainFileName) --yes -d VALUE(cEncryptFileName) .
put UNFORMATTED cPassPhrase.
input-output close.

/* Load the output varible with the file's contents */

RUN ReadTextFile.p (INPUT cPlainFileName, OUTPUT cPlainData).

/* Clean house */

OS-DELETE VALUE(cPlainFileName).
OS-DELETE VALUE(cEncryptFileName).
```

And the program to test it with:

```
DEF VAR cEncryptData AS CHARACTER NO-UNDO.
DEF VAR cPlainData   AS CHARACTER NO-UNDO.

RUN ReadTextFile.p (INPUT "/tmp/a", OUTPUT cEncryptData).

RUN Decrypt.p (
INPUT  cEncryptData ,
INPUT  "sauge@amduus.com",
INPUT "progressrocks",
OUTPUT cPlainData
).

disp cPlainData format "x(50)".
```

ReadTextFile.p

The above programs need this helping code.

```

```

```
/*
 * Removal of this header is illegal.
 * Written by Scott Auge scott_auge@yahoo.com sauge@amduus.com
 * Copyright (c) 2002 Amduus Information Works, Inc.  www.amduus.com
 *
 * 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 Information Works, Inc. 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.
 *
 */

DEF VAR RCSVersion AS CHARACTER INIT "$Header:
/home/sauge/code/progress/ezine35/RCS/ReadTextFile.p,v 1.1 2003/07/29 07:13:07 sauge Exp
sauge $" NO-UNDO.

DEF INPUT PARAMETER cFileName         AS CHARACTER NO-UNDO.
DEF OUTPUT PARAMETER cFileContents    AS CHARACTER NO-UNDO.

DEF VAR vFileLine                     AS CHARACTER NO-UNDO.

INPUT FROM VALUE(cFileName).

REPEAT:

  IMPORT UNFORMATTED vFileLine.

  ASSIGN cFileContents = cFileContents
                       + vFileLine
                       + "~n".
```

```
END.

INPUT CLOSE.
```

Using the routines:

As you can imagine, your going to need to call Encrypt.p and Decrypt.p quite often to access data stored in the database. These are computationally heavy routines. It will slow down your program, and so encryption should be used for only the most important information.

Note that you cannot assume that the same data will yield the same cipher text. In other words, 555-657-0987 encrypted might yield one set of text, while 555-657-0987 encrypted later on will yield another set of text. This is because "salt" is added into the message so that it is practically different every time the Encrypt.p message is run. The cryptosystem will recognize there is salt in there, and decrypt the data to the same message.

You should not use encrypted data as keys to your relations. Your begging for trouble if you do. It is better to use foreign keys to relate the tables. For one thing, as mentioned above – the same plain text data does not guarantee the same cipher text data. One can find the creation of orphaned records mighty quickly. Also your index space will go bonkers, because as you know – encrypted data is much larger than the normal data.

To further re-enforce – a lot of times people will use ENCODE() to hash a password into gobbly gook and compare it to the previous gobbly gook. You can't do that with most public key encryption – because your not guaranteed to get the same cipher text. To use encryption for passwords, you will need to compare the entered password to the Decrypt.p version of the password stored and compare. People who do use a hash to hide passwords should read: http://news.com.com/2100-1009-5053063.html It shows how using large look-up tables (up to 1.6 GB) can be used to work hash's backwards, or at least to an equivalent starting point.

Searching on encrypted fields. This is going to be painful. There are two main methods to accomplish this. One is you will need to do full table scans on the field – basically read the data, decrypt, decide if it is the set you want… go from there.

Another way to aid might be "clue markers." These are a bits of data to aid in selecting rows from the table. For example, one of the searchable fields is FirstName – which is encrypted. Another field might be made called ClueFirstName. Just as a card catalog for a name might have the first two letters to the field, the Clue field will too. That is:

First Name (unencrypted)	ClueFirstName

Scott	Sc
Craig	Cr
Brent	Br

Under Progress, you might be able to search with better bracketing be elongating the clue fields. But of course, if you elongate the clue fields, you more or less put that information out into the open.

Implementation Notes:

As you can guess, obviously the crypto-system software and more importantly – the keys – are on the system implementing the Encrypt and Decrypt routines.

On GUI based applications, that is a lot of places for the keys to be picked up at and fiddled with if the code is run on the user's machine. You can control this by using MetaFrame or the like to send all the users to a single or set of machines to run the code from.

On CHUI you might be able to control the keys a little better by controlling the accounts via UNIX permissions. At least they may be on one server everyone telnets into/X windows out of.

The keys are a little more controlled with web applications – users need to gain access to the machine the transaction server is on to mess with the keys. This can be more difficult if the code is running on a machine not part of the web server hardware collection[2].

The best way to implement such a set of routines is via an AppServer or an Encryption Server. Like the Webspeed transaction server, an AppServer for GUI and CHUI apps may be running on a more secure system on another piece of hardware that is out of the user's reach[3].

To use an encryption server, one may want to make a SSL[4] connection to a port waiting for encryption and decryption services. (This is a good use for Web Services.) It would receive a set of data much like the parameters to Encrypt.p and Decrypt.p – and return the results of those routines to the calling program.

[2] Webspeed allows the web server to have a messenger which contacts a transaction server on another machine.

[3] It is not known how safe packets of data between an application and an AppServer are.

[4] You want to use SSL or TLS to make the connection between the program and the encryption server encrypted it's self. This way people can't sniff the network looking for your packets with the de-crypted data floating in them.

Appendix A: So you have a warning about using insecure memory

If your using GPG on a multi-user system, it will use insecure memory[5]. The way to get past this is to set the SUID to root on the executable.

First find the binary you are executing and be sure it is the one you think you are executing:

```
[~/code/progress/ezine35]$ which gpg
/usr/bin/gpg
```

List the permissions and ownership. You will probably see something like this:

```
[~/code/progress/ezine35]$ ls -l /usr/bin/gpg
-rwxr-xr-x    1 root     root       502400 Aug 22  2000 /usr/bin/gpg
```

If the executable is not owned by root, chown the executable to be owned by root.

Next chmod the executable to use SUID of root when executed.

```
[~/code/progress/ezine35]$ su
Password:
[/home/sauge/code/progress/ezine35]# chmod 4555 /usr/bin/gpg
[/home/sauge/code/progress/ezine35]# ls -l /usr/bin/gpg
-r-sr-xr-x    1 root     root       502400 Aug 22  2000 /usr/bin/gpg
```

Appendix B: Example calling gpg via stdin/stdout.

```
[~/code/progress/ezine35]$ echo "5555-987-7654" | gpg --armour -e --yes  -r
sauge@amduus.com
```

Will output on stdout:

```
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.0.1 (GNU/Linux)
Comment: For info see http://www.gnupg.org

hQIOA/aOys0zndXaEAgApa+Qk3P3SyKLjC+i3SxeJSZADQF2oO8OC+mHbbOmmgwc
UICDe9NJNdykntbFKqDOuGP4QkLf6aaTgSTlAOqrqpGPAgm02PuTFO+qeiwvO/rO
dUjv4j8grqGsfWidpLLdpUQ4NG4begZRCUgj/dqNAo4DpmVD7GPlnKhnxvXvT6y2
/G9koskUpVCWA7a4qu6ZfuvdLTvS2x8/yPGekSScFlcCdwFu9YP8B9fIksD+ckTd
YYAWFEECaxaJ90fzvYq9KJfAsbj1xbaOcWaqgSMwtsNBgpqEIXZUGW0vSY210h2T
e8RYPhFlo36fUOsQ/GxjkCo1km1R0hv+6qECUM8rRQf/Tacl4eewOZSS/fFogqeE
1SODFwVzp6I5975Y010jLEKbDN111PuSZRTiH67aFgX/NliABftUMTtcbW5K+gFe
7nB4opr/r7a3Ip1FqofYvg1kDLoxiMeuUw05OH8LwrLFwxP3ZqOsreFz/fxgDT50
VMy/8tKUgZGqEiV6BJLJzEEZbnkuhuGmdHQz6Phie+jbto4aGJcRDoKqko5ONhKM
```

[5] Insecure memory means the program's binary and data areas can be swapped out to disk. If the OS supports user's "locking" memory into RAM – this won't be a problem. Otherwise the program will need to be run with the permissions of a user that can lock memory to RAM – usually root.

```
sYoEuj85oiAA50zKiXuc+oGlxbWDIzBT2KvqtSw5Xp2wPbKCsl4phwu11Vxk7vwb
wmjx23TpIMTCk3MkZ5VXwXIzTLrA32QibkK8Dlc6LbCTRewcjwP8L6CiKVJBgGjW
9skjFgUXBoyhOhL0uo9gvAKxbY/6mVS9PCel2KJxn0N40g2sh6I=
=ZjLJ
-----END PGP MESSAGE-----
```

Appendix C: Signing a key

When you encrypt something to someone using a key, sometimes it will say it cannot find the trust calculation for that key and prompt you to answer yes or no to actually use it.

This is how you get around it. When you create a new key with a secret key, you can assign the trust to the key by signing it. Use –u with the same name as the –edit-key value.

```
[~/code/progress/ezine35]$ gpg -u sauge@amduus.com --edit-key sauge@amduus.com
gpg (GnuPG) 1.0.1; Copyright (C) 1999 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

Secret key is available.

pub  1024D/59A3A933  created: 2002-01-17 expires: never      trust: f/q
sub  2048g/339DD5DA  created: 2002-01-17 expires: never
(1)  Scott Auge <sauge@amduus.com>

Command> trust
pub  1024D/59A3A933  created: 2002-01-17 expires: never      trust: f/q
sub  2048g/339DD5DA  created: 2002-01-17 expires: never
(1)  Scott Auge <sauge@amduus.com>

Please decide how far you trust this user to correctly
verify other users' keys (by looking at passports,
checking fingerprints from different sources...)?

 1 = Don't know
 2 = I do NOT trust
 3 = I trust marginally
 4 = I trust fully
 s = please show me more information
 m = back to the main menu

Your decision? 4

pub  1024D/59A3A933  created: 2002-01-17 expires: never      trust: f/q
sub  2048g/339DD5DA  created: 2002-01-17 expires: never
(1)  Scott Auge <sauge@amduus.com>

Command> sign

pub  1024D/59A3A933  created: 2002-01-17 expires: never      trust: f/q
              Fingerprint: 1323 5DAA BE20 9BD8 039A  0637 2F6B 7FCC 59A3 A933

     Scott Auge <sauge@amduus.com>

Are you really sure that you want to sign this key
```

```
with your key: "Scott Auge (Test Key) <sauge@amduus.com>"

Really sign? yes

You need a passphrase to unlock the secret key for
user: "Scott Auge (Test Key) <sauge@amduus.com>"
1024-bit DSA key, ID 4D596895, created 2003-07-29


Command> save
[~/code/progress/ezine35]$
```

Where to find the source code:

http://www.amduus.com/OpenSrc/FreePublications/EZineArchive/ezine35.2003.210.14.42.zip

About the author: Scott Auge is the founder of Amduus Information Works, Inc. He has been programming in the Progress environment since 1994. His works have included E-Business initiatives and focuses on web applications on UNIX platforms.
sauge@amduus.com

New application available from Amduus!

If you liked the encryption article above, you may be interested in our ASP offering of an encryption server.

Encryption server is a box you plug into your network. Manufacturer may vary, but 1U server size will be common.

The machine will aid you to protect information within your Progress application. Fear less about troublemakers gaining access to your database full of credit card numbers, social security numbers, customer email addresses, etc. Even if they have the WHOLE database – the important portions will be encrypted and out of their reach!

What do I need to do to work it?

Your application will need to use our API's to reach the server. Application types can be GUI, CHUI, WWW or background processes. Your application should run on Version 9.

Your company will need to reside in the United States.

The application server would be shipped to you with instructions on how to operate it. Phone support would be available.

How much will it cost?

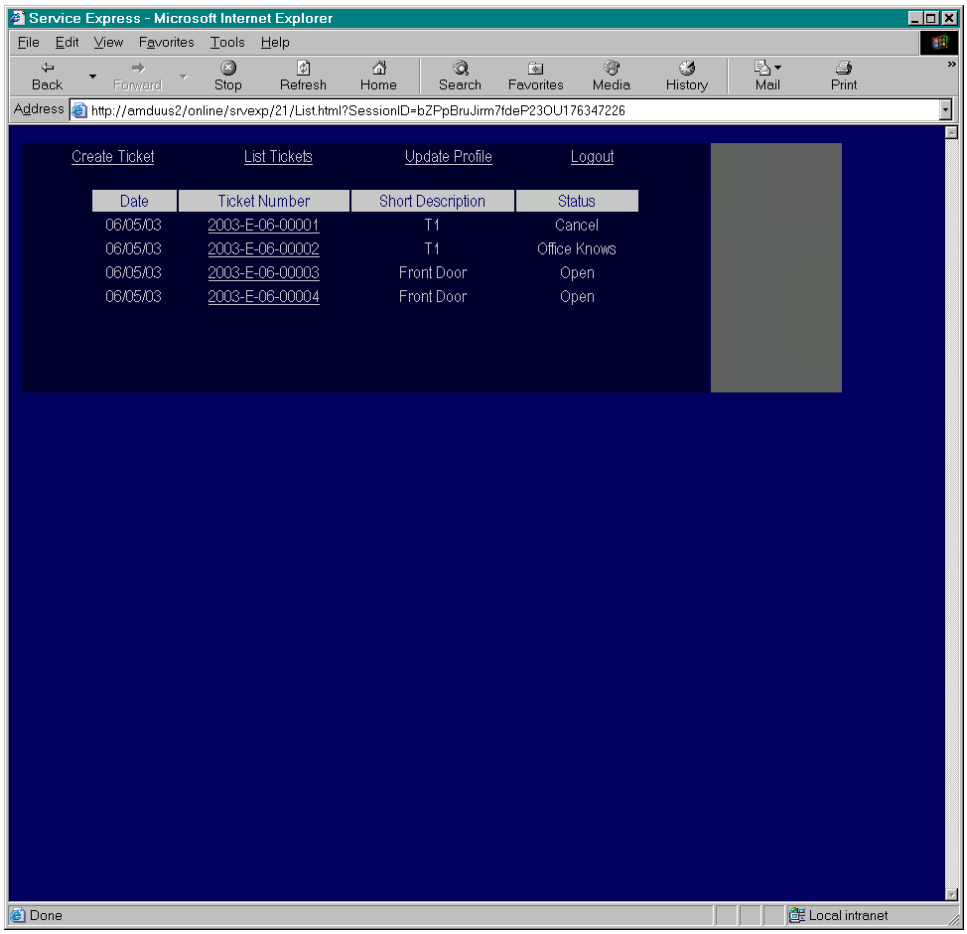
Set-up will be $1,400
Monthly rate of $100.00 per user or web agent accessing the machine

How can I learn more?

Contact Diane Fischer at dfischer@amduus.com

Service Express ASP application

Amduus Information Works, Inc. finally has a working version of the Service Express available!

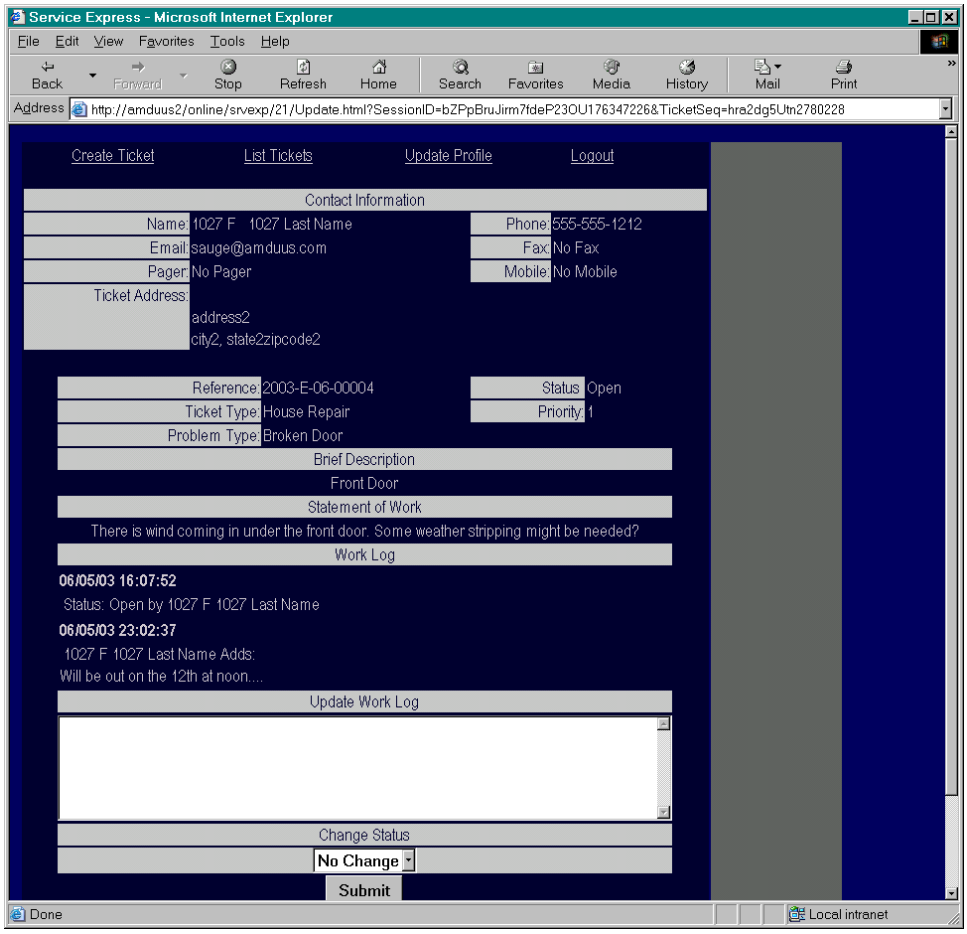


A listing of tickets belonging to a customer

What is it?

What is it? It is a work order/issue tracking system. It lets your customers (whether they be your co-workers, or the company's customers) create work orders (tickets) to solve problems. The

tickets can be categorized by types and problems; And a work flow can be developed to track the status of the ticket. All of this is configurable to match your industry or user base.



A customer's ticket information and work log

Who!

We would like to make this software available to you! It is completely web based. The only installation of software that would need to be done are web browsers in your company.

Configuration of the software can be performed by us or by you. We have work sheets to aid you in deciding the configuration. We also have templates already made for configuring the software for internal IT use, internal application development, etc.

Amduus Information Works, Inc. also provides documentation services! Scott Auge notes, "One of the things I have noticed throughout my contracting career is that companies with developed software always seem to be missing or weak on user documentation, administration documentation, and programmer documentation." Amduus can help you with this!

How!

You can reach the public portion of a demo for the application here:

https://www.amduus.com/cgi-bin/se0001pub/21/index.html

You would be acting as a customer of a company called "Demo" with relations to the organization running the web site. Your company name would be Demo, and to prove that you are indeed an employee/representative of that company, you would know that the authorization code is Demo.

Let me point out that you could be an internal customer, such as HR approaching IT about setting up a new employee; Or, you could be an external customer approaching the organization with a request for a service to be performed.

This could be used by a manufacturing company for repairs/over-hauls of their equipment. An apartment complex for handling property issues. Any company/organization that wishes to interact with it's customers in a manner that needs to be defined and tracked for a process of completion.

Be sure to see the accompanying Power Point presentations for more information.

Please contact Scott Auge at sauge@amduus.com if you have additional questions!

Publishing Information:

Scott Auge publishes this document. I can be reached at sauge@amduus.com.

Amduus Information Works, Inc. assists in the publication of this document by providing an internet connection and web site for redistribution:

Amduus Information Works, Inc.
1818 Briarwood
Flint, MI 48507
http://www.amduus.com

Other Progress Publications Available:

This document focuses on the programming of Progress applications. If you wish to read more business oriented articles about Progress, be sure to see the Profile's magazine put out by Progress software http://www.progress.com/profiles/

There are other documents/links available at http://www.peg.com .

There is a web ring of sites associated with Progress programming and consultants available at http://i.webring.com/hub?ring=prodev&id=38&hub .

Products/Services Available From Amduus:

Amduus Information Works, Inc. is a Progress reseller and ASPen partner. We primarily develop UNIX/Linux based applications with web interfaces for manufacturing, service, and law enforcement communities.

We also perform integration of Progress applications to non-Progress applications through such languages and tools as MQ Series, C, and C++.

Amduus provides support for the following applications: Blue Diamond, Denkh, Denkh HTML Reporter, Red Arrow Portal (CMS), Survey Express and other software.

Amduus is looking for consultants who might want to promote the use of our tools at user groups and companies they might work in. Send some information to sauge@amduus.com to let me know you are out there!

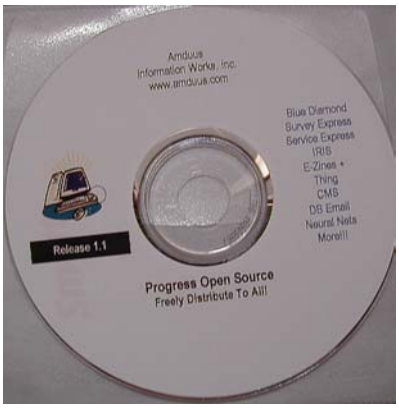
Article Submission Information:

> Please submit your article in Microsoft Word format or as text. Please include a little bit about yourself for the About the Author paragraph.
>
> Looking for technical articles, *marketing Progress* articles, articles about books relevant to programming/software industry, white papers, etc.
>
> Send your articles to sauge@amduus.com! Thanks!

Order Form for Progress Open Source CD-ROM

COUPON 001A

This is an offer for the CD-ROM at lower than list savings! This is a great way to support the E-Zine too!



Mail this form to:
Amduus Information Works, Inc.
1818 Briarwood
Flint, MI 48507

Please send _____ copies of the Open Source CD-ROM at $25.00 per disk to:

Name _____
Company _____
Address _____
City _____
State _____ Country _____
Zip _____

Please make your checks/money orders out to: Amduus Information Works, Inc. Cash works too!
This offer only valid in the United States of America and those countries with postal agreements with the United States Post Office.

The CD-ROM includes (all source code included):

- Blue Diamond/IRIS – Webspeed alternatives
- Survey Express – easily create text templates of surveys and then have the program generate the web pages automatically
- Service Express – Web based Help Desk.
- The Progress E-Zines, books on learning to program in Webspeed (PDF/Word/HTML)
- Denkh HTML Reporter – web based report writer
- CMS – a web content management system
- DB Email – Use pop3 to download emails into a Progress database
- Neural Networks – experiments in spam recognition and text message classification
- Denkh – create PDF file reports for Webspeed/UNIX CHUI!
- More!

Purchase Progress From Amduus Information Works!

Amduus Information Works, Inc. offers multiple Open Source applications. But some of these applications depend on software from Progress Software Corporation. This form is made available for you to purchase these licenses to run the applications made available by Amduus as well your own "home brewed" software.

Warning: You cannot REPLACE licenses from another VAR/ISV with licenses from Amduus Information Works, Inc. For example, if you purchased Progress licenses from QAD, Inc. for use with their application, you cannot replace those licenses with those from Amduus Information Works, Inc. ***YOU will be illegally using Progress licenses under such circumstances.***

Warning: You cannot INCREASE licenses from another VAR/ISV with licenses from Amduus Information Works, Inc. For example, if you purchased Progress licenses from QAD, Inc. for use with their application, you cannot increase user count with licenses with those from Amduus Information Works, Inc. ***YOU will be illegally using Progress licenses under such circumstances.***

Note: If you wish to develop/use Amduus software for use WITH another VAR/ISV's application, you CAN use licenses from Amduus Information Works, Inc. So all together, you would have licenses from Amduus AND licenses from that VAR/ISV.

Item	Description
Enterprise Level Database Server	Multi-user, relational database engine designed to support high volume, distributed, enterprise-level applications. Provides database services in both development and deployment environments. Application execution system for local application deployment. Includes ability to accept connections from remote clients, including ODBC client applications. Includes support for SQL-92, drivers for ODBC and JDBC access to SQL-92, and an embedded SQL-92 precompiler, enabling applications written in C to access Progress databases. Includes Win32 character client support.
Workgroup Level Database Server	Multi-user, relational database engine designed to support workgroup and departmental applications (less than 50 connections). Provides database services in both development and deployment environments. Application execution system for local application deployment. Includes ability to accept connections from remote clients, including ODBC client

	applications. Includes support for SQL-92, drivers for ODBC and JDBC access to SQL- 92, and an embedded SQL-92 precompiler, enabling applications written in C to access Progress databases. Includes Win32 character client support.
Progress Client Network	Provides networking support for 4GL and SQL-89 client access to a remote Progress RDBMS or Progress DataServer. Includes application execution system, SQL Client Access, and Win32 character client support.
Webspeed Transaction Server	Deployment engine for transaction-based Web applications. Includes networking for connections to remote Progress RDBMS servers, AppServers, or WebSpeed Transaction Servers. Includes Client Networking. Note: NameServer Load Balancer is included with licenses of 50 agents or higher. Usually includes Webtools for the given operationg system.
Progress Webspeed Workshop (For the Programmer/Analyst)	Integrated suite of development tools for building and testing transaction-based Web applications using HTML-based interfaces. Includes WebSpeed AppBuilder, SpeedScript (4GL) language, WebSpeed Messenger, Wizards, Pro*Tools, WebTools, Data Dictionary, and Data Administration tools. Also includes a WebSpeed Development Server.
4GL Development System (For the Programmer/Analyst)	Character-based development environment. Includes the Progress 4GL, Data Dictionary, Data Administration, Help Compiler, Procedure Editor, Application Compiler, Personal RDBMS, and Client Networking.
Progress Provision Plus (For the Programmer/Analyst)	An Integrated Development Environment (IDE) for 4GL, Web and Open Client application development and testing. Includes ProVision, WebSpeed Workshop, WebSpeed Messenger, and Open Client Toolkit products. Also includes a ProVision Plus Development Server. Note: ProVision Plus contains a coupon for a Roundtable Lite license at no charge. Note: This is a GUI/CHUI/WWW development tool combining the 4GL development system and the Webspeed Workshop development system.
Amduus Information Works, Inc. Open Source CD-	Blue Diamond/IRIS – Webspeed alternatives

ROM	Survey Express – easily create text templates of surveys and then have the program generate the web pages automatically Service Express – Web based Help Desk. The Progress E-Zines, books on learning to program in Webspeed (PDF/Word/HTML) Denkh HTML Reporter – web based report writer CMS – a web content management system DB Email – Use pop3 to download emails into a Progress database Neural Networks – experiments in spam recognition and text message classification Denkh – create PDF file reports for Webspeed/UNIX CHUI! More!

Direct any questions to Scott Auge at sauge@amduus.com

Order Form

Purchase Progress Licenses From Amduus Information Works, Inc.

Name _____
Company _____
Address _____
City _____
State _____ Country _____
Zip _____
E-Mail _____
Phone _____

Warning: You cannot REPLACE licenses from another VAR/ISV with licenses from Amduus Information Works, Inc. For example, if you purchased Progress licenses from QAD, Inc. for use with their application, you cannot replace those licenses with those from Amduus Information Works, Inc. *YOU will be illegally using Progress licenses under such circumstances.*

Warning: You cannot INCREASE licenses from another VAR/ISV with licenses from Amduus Information Works, Inc. For example, if you purchased Progress licenses from QAD, Inc. for use with their application, you cannot increase user count with licenses with those from Amduus Information Works, Inc. *YOU will be illegally using Progress licenses under such circumstances.*

Note: If you wish to develop/use Amduus software for use WITH another VAR/ISV's application, you CAN use licenses from Amduus Information Works, Inc. So all together, you would have licenses from Amduus AND licenses from that VAR/ISV.

Send your order to:

Amduus Information Works, Inc.
1818 Briarwood
Flint, MI 48507

Item	Price	Qty	Total
Enterprise Level Database Server Price per connection Circle your OS: • Tru64 UNIX 4.0F (4.0G, 5.0, 5.0A, 5.1, 5.1A, 5.1B) • HP-UX 32 Bit PA-RISC (Patches XSWGR1100+ PHSS_22543, 22868, 22478, HPUX 11i) • HP-UX 64 Bit PA_RISC (Patches XSWGR1100+ PHSS_22543, 22868, 22478, HPUX 11i) • IBM AIX (4.3.3 MUP Level 2) • Red Hat Linux Intel (6.2, 7.0, 7.1, 7.2, 7.3, 8.0, Advanced Server 2.1, Red Hat Enterprise Linux ES) • Sun Solaris SPARC 32Bit (Solaris SPARC 2.6 Solaris 7, 8, 9) • Sun Solaris SPARC 64 Bit (Solaris SPARC 2.8, Solaris 8, 9)	$1,022.00		
Workgroup Level Database Server Price per connection Circle your OS: • Tru64 UNIX 4.0F (4.0G, 5.0, 5.0A, 5.1, 5.1A, 5.1B) • HP-UX 32 Bit PA-RISC (Patches XSWGR1100+ PHSS_22543, 22868, 22478, HPUX 11i) • HP-UX 64 Bit PA_RISC (Patches XSWGR1100+ PHSS_22543, 22868, 22478, HPUX 11i) • IBM AIX (4.3.3 MUP Level 2) • Red Hat Linux Intel (6.2, 7.0, 7.1, 7.2, 7.3, 8.0, Advanced Server 2.1, Red Hat Enterprise Linux ES) • Sun Solaris SPARC 32Bit (Solaris SPARC 2.6 Solaris 7, 8, 9) • Sun Solaris SPARC 64 Bit (Solaris SPARC 2.8, Solaris 8, 9)	$350.00		
Progress Client Network Per User Circle your OS: • Tru64 UNIX 4.0F (4.0G, 5.0, 5.0A, 5.1, 5.1A, 5.1B)	$196.00		

• HP-UX 32 Bit PA-RISC (Patches XSWGR1100+ PHSS_22543, 22868, 22478, HPUX 11i) • HP-UX 64 Bit PA_RISC (Patches XSWGR1100+ PHSS_22543, 22868, 22478, HPUX 11i) • IBM AIX (4.3.3 MUP Level 2) • Red Hat Linux Intel (6.2, 7.0, 7.1, 7.2, 7.3, 8.0, Advanced Server 2.1, Red Hat Enterprise Linux ES) • Sun Solaris SPARC 32Bit (Solaris SPARC 2.6 Solaris 7, 8, 9) • Sun Solaris SPARC 64 Bit (Solaris SPARC 2.8, Solaris 8, 9) • MS Windows			
Webspeed Transaction Server 5 Agent Circle Your OS: • Tru64 UNIX 4.0F (4.0G, 5.0, 5.0A, 5.1, 5.1A, 5.1B) • HP-UX 32 Bit PA-RISC (Patches XSWGR1100+ PHSS_22543, 22868, 22478, HPUX 11i) • HP-UX 64 Bit PA_RISC (Patches XSWGR1100+ PHSS_22543, 22868, 22478, HPUX 11i) • IBM AIX (4.3.3 MUP Level 2) • Red Hat Linux Intel (6.2, 7.0, 7.1, 7.2, 7.3, 8.0, Advanced Server 2.1, Red Hat Enterprise Linux ES) • Sun Solaris SPARC 32Bit (Solaris SPARC 2.6 Solaris 7, 8, 9) • Sun Solaris SPARC 64 Bit (Solaris SPARC 2.8, Solaris 8, 9)	$7,560.00		
Webspeed Transaction Server 25 Agent Circle Your OS: • Tru64 UNIX 4.0F (4.0G, 5.0, 5.0A, 5.1, 5.1A, 5.1B) • HP-UX 32 Bit PA-RISC (Patches XSWGR1100+ PHSS_22543, 22868, 22478, HPUX 11i) • HP-UX 64 Bit PA_RISC (Patches XSWGR1100+ PHSS_22543, 22868, 22478, HPUX 11i) • IBM AIX (4.3.3 MUP Level 2)	$14,000.00		

• Red Hat Linux Intel (6.2, 7.0, 7.1, 7.2, 7.3, 8.0, Advanced Server 2.1, Red Hat Enterprise Linux ES) • Sun Solaris SPARC 32Bit (Solaris SPARC 2.6 Solaris 7, 8, 9) • Sun Solaris SPARC 64 Bit (Solaris SPARC 2.8, Solaris 8, 9)			
Webspeed Transaction Server 50 Agent Circle Your OS: • Tru64 UNIX 4.0F (4.0G, 5.0, 5.0A, 5.1, 5.1A, 5.1B) • HP-UX 32 Bit PA-RISC (Patches XSWGR1100+ PHSS_22543, 22868, 22478, HPUX 11i) • HP-UX 64 Bit PA_RISC (Patches XSWGR1100+ PHSS_22543, 22868, 22478, HPUX 11i) • IBM AIX (4.3.3 MUP Level 2) • Red Hat Linux Intel (6.2, 7.0, 7.1, 7.2, 7.3, 8.0, Advanced Server 2.1, Red Hat Enterprise Linux ES) • Sun Solaris SPARC 32Bit (Solaris SPARC 2.6 Solaris 7, 8, 9) • Sun Solaris SPARC 64 Bit (Solaris SPARC 2.8, Solaris 8, 9)	$25,200.00		
Progress Webspeed Workshop (For the Programmer/Analyst) Per User • MS Windows	$1,330.00		
4GL Development System (For the Programmer/Analyst) Per User Circle Your OS: • Tru64 UNIX 4.0F (4.0G, 5.0, 5.0A, 5.1, 5.1A, 5.1B) • HP-UX 32 Bit PA-RISC (Patches XSWGR1100+ PHSS_22543, 22868, 22478, HPUX 11i) • HP-UX 64 Bit PA_RISC (Patches XSWGR1100+ PHSS_22543, 22868, 22478, HPUX 11i) • IBM AIX (4.3.3 MUP Level 2) • Red Hat Linux Intel (6.2, 7.0, 7.1, 7.2, 7.3, 8.0, Advanced Server 2.1, Red Hat Enterprise Linux ES)	$3,920.00		

• Sun Solaris SPARC 32Bit (Solaris SPARC 2.6 Solaris 7, 8, 9) • Sun Solaris SPARC 64 Bit (Solaris SPARC 2.8, Solaris 8, 9)			
Progress Provision Plus (For the Programmer/Analyst) Per User Circle Your OS: • MS Windows	$6,300.00		
Amduus Open Source CD-ROM Per Organization	Free!		Free!
		Tax	
		Grand Total	