

## The Progress Electronic Magazine

**In this issue:**

**Publisher’s Statement:** ..... 2

**Coding Article: Update Service for your Progress Application**..... 3

    What your still going to need to do ..... 3

    Theory of Operation ..... 4

**OneStep Charge**..... 4

    Premier Credit Card Processing for the 4GL..... 4

    Database Tables ..... 4

    Determining the current version on the local disk ..... 10

    Determining the current version in the repository ..... 11

    Performing an upgrade ..... 12

    Loading the repository ..... 13

    Just for fun! ..... 14

    Download code and schema ..... 14

**Reference Partners Wanted**..... 17

    What is it? ..... 17

    Who! ..... 18

    How! ..... 19

**Publishing Information:**..... 20

**Other Progress Publications Available:** ..... 20

**Products/Services Available From Amduus:** ..... 21

**Article Submission Information:**..... 21

**Order Form for Progress Open Source CD-ROM**..... 22

*This document may be freely shared with others without modification. Subscribe for free here: <http://www.amduus.com/online/dev/ezine/EZineHome.html>*

*Though intended for users of the software tools provided by Progress Software Corporation, this document is NOT a product of Progress Software Corporation.*

© Scott Auge and Amduus Information Works, Inc. 2003

**Publisher's Statement:**

In this issue we explore a tool that eases the updating of applications on remote clients. Here is the problem – you have a program on a slurry of PC's across a network – perhaps even across a state or country. You need to update the r-code available on that application to the latest and greatest!

**!!!WANTED!!!**

Amduus Information Works, Inc. is looking for consultants to resell access to up-coming ASP web based software. 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](mailto:sauge@amduus.com) for more information.

How can you do this conveniently? 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](http://www.paypal.com) and  
send money to [scott\\_auge@yahoo.com](mailto:scott_auge@yahoo.com) via your credit card!*

*Better yet – simply order the CD-ROM available on the last page... Thanks!*

**Coding Article: Update Service for your Progress Application***By Scott Auge*

More and more we are seeing companies providing upgrade services for user's of their applications. Red Hat even makes money on keeping their Linux distribution tip top with a simple click of an icon/command line.

I have seen large organizations attempting to keep multiple dispersed machines loaded with progress applications up to date via a mirage of strategies.

You can do a couple of things to accomplish upgrading the r-code of an application:

- Make a visit out to all the computers and upgrade the r-code available on the system. This plainly sucks if you have machines across town or in multiple buildings.
- Put the r-code on mapped drives and run the code from there. This has been shown to be incredibly slow for more complex apps – especially when using SmartObjects. Large amounts of .r files coming across the network slow application start up; And here and there the app will just “freeze” for a few moments while it loads up a .r from far away....
- Use a tool to upgrade the r-code local to a machine!

As you can guess, this approach uses the last in this list – a tool that will upgrade the r-code on a user's local file system. Clean, fast, and automatic!

***What your still going to need to do***

Your going to still need to install Progress on the local machine. The repository doesn't install Progress binaries, DLLs, .so's, or make registry entries.

Your still going to need to set up the PROPATH on the local machine. This is because the repository code doesn't check for disk space or arbitrarily decide where to put the code downloaded.

You may also need to create an environmental variable that contains the directory path to where the code should be deposited upon an upgrade. It depends on your implementation of the code below.

**Reach nearly one thousand  
programmers and companies.**

**Your ad could be here!**

**Advertise in the E-Zine for  
\$10.00 per issue!**

### *Theory of Operation*

Since Progress can store BLOBs (Binary Large Objects) in the database<sup>1</sup>, we have a nice place accessible to the program that it can pull the files from since it is already connected to a database<sup>2</sup>.

#### **OneStep Charge**

##### ***Premier Credit Card Processing for the 4GL***

- Integration in 10 minutes
- Realtime authorizations in 2 seconds
- Pure Progress
- Only Requires V9 or higher
- Fully-documented API
- NO drop files
- NO plain-text hazards
- Certified with all major processors, such as VITAL, Nova, Paymentech, NDC, FHMS
- Tri-8-sponsored merchant accounts (optional) can save literally thousands per month

**<http://OneStepCharge.com>  
**[oscinfo@onestepcharge.com](mailto:oscinfo@onestepcharge.com)  
**405.377.3888******



What we need is:

1. a method to determine the current version loaded on the user's computer,
2. a method to determine the current version available in the database repository,
3. the method to download that information into the user's computer,
4. and the method to load the database with the repository files.

Let's get started!

### ***Database Tables***

Here is a convenient reference of the database tables needed. We can read the tables and have a pretty good idea what is going to happen in the code.

Why do we put the code in a database? I have thought about a copying function – problems include:

- How to do a massive copy across multiple operating systems? The commands are different (UNIX cp, DOS copy.)

---

<sup>1</sup> Version 9.1 or better needed

<sup>2</sup> In fact, you can store user preferences in the database. This might include information like the location of the code for that user's local computer on their hard drive.

- Sometimes the protocols are different between machines (Samba or FTP?)
- How to tell the local computer if it has the correct version number or not? Have it look for some little file out there amongst the gigabytes of upgrade space? And there is that protocol problem again.

Keeping the goodies in the database means that it will be naturally available to all clients connecting to that database, and that we keep the coding “in the 4GL house” so we don’t need to code around OS differences and protocol differences.

The Upgrade table contains one record, and that record will state which upgrade loaded into the repository is the latest and greatest.

That table relates into the XUpgradeOSFile table. This table has an entry for each repository of code placed into it.

This table then relates into the XUpgradeOSFileList table. This table lists all the files that are part of that repository.

This table relates into OSFile which acts as a header for the file information.

The OSFile relates into OSFileData which contains the actual file data information.

It does not matter the type of file that is stored in the database. It will store JPGs, .r’s, .p’s, HTML, etc.

Below is a listing of the tables needed. The schema is available in the source download referenced later in this article.

```
20/07/03 21:58:34          PROGRESS Report
Database: amduus (PROGRESS)
```

```
=====
===== Table: XUpgradeOSFile =====
```

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

Table Name	Dump Name	Table Field Flags	Count	Index Count	Table Label
XUpgradeOSFile	xupgrade		3	3	XUpgradeOSFile

Storage Area: Schema Area

===== FIELD SUMMARY =====  
===== Table: XUpgradeOSFile =====

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

Order	Field Name	Data Type	Flags	Format	Initial
10	XUpgradeOSFileID	char	i	x(8)	
20	Version	char	i	x(8)	
30	ModuleName	char	i	x(8)	

Field Name	Label	Column Label
XUpgradeOSFileID	XUpgradeOSFileID	XUpgradeOSFileID
Version	Version	Version
ModuleName	ModuleName	ModuleName

===== INDEX SUMMARY =====  
===== Table: XUpgradeOSFile =====

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

Flags	Index Name	Cnt	Field Name
	key1	1	+ ModuleName
	key2	1	+ Version
pu	pukey	1	+ XUpgradeOSFileID

\*\* Index Name: key1  
Storage Area: Schema Area  
\*\* Index Name: key2  
Storage Area: Schema Area  
\*\* Index Name: pukey  
Storage Area: Schema Area

===== FIELD DETAILS =====  
===== Table: XUpgradeOSFile =====

20/07/03 21:58:43 PROGRESS Report  
Database: amduus (PROGRESS)

===== Table: XUpgradeOSFileList =====

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

Table Name	Dump Name	Table Field Count	Index Count	Table Label
XUpgradeOSFileList	xuplist	2	2	XUpgradeOSFileList

Storage Area: Schema Area

===== FIELD SUMMARY =====  
===== Table: XUpgradeOSFileList =====

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

Order	Field Name	Data Type	Flags	Format	Initial
10	OSFileID	char	i	x(8)	
20	XUpgradeOSFileID	char	i	x(8)	

Field Name	Label	Column Label
OSFileID	OSFileID	OSFileID
XUpgradeOSFileID	XUpgradeOSFileID	XUpgradeOSFileID

===== INDEX SUMMARY =====  
===== Table: XUpgradeOSFileList =====

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

Flags	Index Name	Cnt	Field Name
p	key1	1	+ OSFileID
	key2	1	+ XUpgradeOSFileID

\*\* Index Name: key1  
Storage Area: Schema Area  
\*\* Index Name: key2  
Storage Area: Schema Area

===== FIELD DETAILS =====  
===== Table: XUpgradeOSFileList =====

20/07/03 21:57:41 PROGRESS Report  
Database: amduus (PROGRESS)

===== Table: OSFile =====

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

Table Name	Dump Name	Table Field Flags	Field Count	Index Count	Table Label
OSFile	osfile		7	2	OSFile

Storage Area: Schema Area

===== FIELD SUMMARY =====  
===== Table: OSFile =====

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

Order	Field Name	Data Type	Flags	Format	Initial
10	FileName	char	i	x(8)	
40	OSFileType	char		x(8)	
50	OSFileID	char	i	x(8)	
60	CreateDate	date	i	99/99/99	?
70	CreateTime	inte	i	->, >>>, >>9	0

Field Name	Label	Column Label
FileName	FileName	FileName
OSFileType	OSFileType	OSFileType
OSFileID	OSFileID	OSFileID
CreateDate	CreateDate	CreateDate
CreateTime	CreateTime	CreateTime

===== INDEX SUMMARY =====  
 ===== Table: OSFile =====

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

Flags	Index Name	Cnt	Field Name
	key1	3	+ FileName + CreateDate + CreateTime
pu	pukey2	1	+ OSFileID

\*\* Index Name: key1  
 Storage Area: Schema Area  
 \*\* Index Name: pukey2  
 Storage Area: Schema Area

===== FIELD DETAILS =====  
 ===== Table: OSFile =====

20/07/03 21:57:53 PROGRESS Report  
 Database: amduus (PROGRESS)

===== Table: OSFileData =====

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

Table Name	Dump Name	Table Field Flags	Field Count	Index Count	Table Label
OSFileData	osfileda		4	1	OSFileData

Storage Area: Schema Area

===== FIELD SUMMARY =====  
===== Table: OSFileData =====

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

Order	Field Name	Data Type	Flags	Format	Initial
20	Sequence	inte	i	->, >>>, >>9	0
50	OSFileID	char	i	x(8)	
60	Data	raw		x(8)	

Field Name	Label	Column Label
Sequence	Sequence	Sequence
OSFileID	OSFileID	OSFileID
Data	Data	Data

===== INDEX SUMMARY =====  
===== Table: OSFileData =====

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

Flags	Index Name	Cnt	Field Name
p	pukey1	2	+ OSFileID + Sequence

\*\* Index Name: pukey1  
Storage Area: Schema Area

===== FIELD DETAILS =====  
===== Table: OSFileData =====

20/07/03 21:58:16 PROGRESS Report  
Database: amduus (PROGRESS)

===== Table: Upgrade =====

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

Table Name	Dump Name	Table Field Flags	Table Count	Index Count	Table Label
Upgrade	upgrade		4	3	Upgrade

Storage Area: Schema Area

===== FIELD SUMMARY =====  
===== Table: Upgrade =====

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

Order	Field Name	Data Type	Flags	Format	Initial
10	CurrentVersion	char	i	x(8)	
20	UpgradeID	char	i	x(8)	
30	State	char		x(8)	
40	ModuleName	char	i	x(8)	

Field Name	Label	Column Label
CurrentVersion	CurrentVersion	CurrentVersion
UpgradeID	UpgradeID	UpgradeID
State	State	State
ModuleName	ModuleName	ModuleName

===== INDEX SUMMARY =====  
 ===== Table: Upgrade =====

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

Flags	Index Name	Cnt	Field Name
	key1	1	+ ModuleName
	key2	1	+ CurrentVersion
pu	pukey	1	+ UpgradeID

\*\* Index Name: key1  
 Storage Area: Schema Area  
 \*\* Index Name: key2  
 Storage Area: Schema Area  
 \*\* Index Name: pukey  
 Storage Area: Schema Area

===== FIELD DETAILS =====  
 ===== Table: Upgrade =====

***Determining the current version on the local disk***

The best way to determine the version on a user’s local computer is a global variable containing the version name loaded. To keep it simple, it travels with the r-code when it is moved around<sup>3</sup>, lots of benefits!

Version naming can be varied. Some organizations might want to use the old number approach: Version 4.1. Other companies may be inclined to use a “build number” such as

---

<sup>3</sup> Be sure to upgrade your version number BEFORE deploying it!

2003.07.14.13.34. And yet another way is to to use version names such as “Red 345” or “Tango 12.”

As such, the version number should be a character variable – as that type will handle numbers and characters.

Just in case the version control is different – we wrap the question in a function that can be altered easily and yet be accessible with the rest of the code methodology.

```
FUNCTION MyVersion RETURNS CHARACTER:
  DEF SHARED VAR LoaderVersionNumber AS CHARACTER NO-UNDO.
  RETURN LoaderVersionNumber.
END. /* FUNCTION MyVersion () */
```

A global variable should be made available in the application start-up module so that any use of the MyVersion function will have it available.

```
DEF NEW SHARED VAR LoaderVersionNumber AS CHARACTER NO-UNDO.
ASSIGN LoaderVersionNumber = "Tango 12".
```

You might want a module of code that upon execution of the application, checks the version number. Another option might be a “Check For Update” button on the application menu.

I recommend checking upon start up as the database schema may have changed and there is always disasters on r-code touching tables that have changed.

### ***Determining the current version in the repository***

Once we have the version of code available on the local machine, we need to see the latest version of the code that is available on the repository side. Since there maybe other ways of doing this than shown below, we wrap the question in a function so that it plays well with the others.

```
FUNCTION RepositoryVersion RETURNS CHARACTER (INPUT cModuleName AS CHARACTER):
  /* Find the record that has information about the version number in the */
  /* Repository on the database. */
  FIND Upgrade NO-LOCK
  WHERE Upgrade.ModuleName = cModuleName
```

```
NO-ERROR.  
  
  /* Error out if not available. */  
  IF NOT AVAILABLE Upgrade THEN RETURN ?.  
  
  /* Error out if the repository is being loaded at this moment */  
  IF Upgrade.State <> "READY" THEN RETURN ?.  
  
  /* All is good, return the version number available in the repository. */  
  RETURN Upgrade.CurrentVersion.  
  
END. /* FUNCTION RepositoryVersion () */
```

The function comes in with a module name. In case an application is modularized upon features that are sold, there might be a need to categorize your r-code.

As you may have noticed, the code above checks for the existence of the Upgrade record. If it is not available the function will return a ?.

If the Upgrade.State is not equal to READY it will return a ?. This is to handle the situation where the r-code is being loaded into the database at the SAME TIME as the user is attempting to do an upgrade<sup>4</sup>. If not all the r-code is loaded into the database, there is the distinct possibility the user will not receive ALL the r-code for that version. When the database is being loaded with the given code, the Upgrade.State field will be populated with LOADING and will be changed to READY upon completion of the load.

Note that the Upgrade table will only have ONE record – detailing the latest version of the repository loaded into the database.

### ***Performing an upgrade***

This code is the basics of getting the upgrade going. You may want a pop-up dialogue box that states an upgrade is being performed in a GUI environment. If your in a character environment, you may want to put up a frame – or a web environment you may want to put up a message page about the system being upgraded.

```
{RepositoryVersion.i}  
{MyVersion.i}  
  
/* Check if the current version matches the repository version. If */
```

---

<sup>4</sup> I have seen some very large applications with megabytes of r-code in it. It might take some time to load it into the database, and so we don't want clients downloading it before it is fully available!

```

/* not, begin pulling down the repository version. */
IF RepositoryVersion() <> MyVersion() THEN
  RUN PullRepository.p ... .

```

A call to the PullRepository.p code is exemplified below. Note you place the base directory as the first argument, and that it must have a trailing /, and then the module name followed by the version. There is a fourth argument that will pass back messages – you are looking for NO\_ERROR. See the source for the other messages.

Note that the database tables allow for multiple versions of an application to go into the repository. So, if something goes wrong with an upgrade, it can be backed out by calling PullRepository.p with a previous version name.<sup>5</sup>

```

run PullRepository.p
(input "/tmp/test1/", /* base directory */
 input "test", /* module */
 input "2", /* version */
 output c /* message */
).

```

### ***Loading the repository***

In order to use the above code – one has to load up the database tables. This is done via the LoadRepository.p program as exemplified below.

Note that the base directory must have a trailing /. All files and subdirectories in the base directory will be loaded into the database. They will be named as if the base directory were part of the PROPATH. In other words,

True Path	Loaded Path
/tmp/test1/e.txt	e.txt
/tmp/test1/test2/a.txt	test2/a.txt
/tmp/test1/test2/test3/test.txt	test2/test3/test.txt

---

<sup>5</sup> You may want to create a menu item on your application that allows a user to pull out a given version of the application. This way, you can tell the users what to type into a dialog box to uninstall a version. Backing out a version can also be done by changing the Upgrade.CurrentVersion to a previous version stored in the repository.

```
RUN LoadRepository.p
(INPUT "/tmp/test1/", /* Base Directory */
 INPUT "test", /* Module Name */
 INPUT "2" /* Version */
).
```

The module name is given next and then the version number.

Very simple. The available source code has some tools to load up a repository via a UNIX command line. People with GUI experience might opt to create a GUI screen to aid in this (send it along to [sauge@amduus.com](mailto:sauge@amduus.com)!) Of course, you can do this right from the procedure editor and webspool script lab also.

### ***Just for fun!***

We have the basics for making an online back up system for PCs into a Progress based database.

One could use the LoadRepository.p program to walk the directories of a PC storing files into the database under a module name of the “user/directory set” and the version being a date and time string.

Then to retrieve all the files, use the PullRepository.p routines to pull the files back out of the database onto the local PC. Since this is awkward, it might be exciting to create a program that will list the files in a back up (repository) and then use RetrieveFile.p to pull that file back into the PC.

Even more fun! Since we can store and retrieve files from the database with a version and module name – we have the beginnings of a document control system! See StoreFile.p for how to load a file from the PC into the database.

Hopefully people will work this code into some open source tools for these purposes! ☺ Please send them to me at [sauge@amduus.com](mailto:sauge@amduus.com)!

### ***Download code and schema***

You can download the code and schema here:

<http://www.amduus.com/OpenSrc/FreePublications/EZineArchive/ezine34.2003.202.01.24.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](mailto:sauge@amduus.com)

## XCheck

Logfile analyzing  
system checking

- usable with Windows or UNIX
- check activity of NS/DB/WS/httpd
- analyze logfiles of NS/DB/WS
- check drive space, space in DB
- execute self defined scripts
- analyze self defined logfiles
- get notified by e-mail, http
- or screen output

## Viper

Visual Printing  
and Enhanced Reports

- uses Windows printer drivers
- data processing with 4GL
- incl. layout designer (VFD)
- stores layouts DB or file based
- no runtime licence cost
- supports bmp/jpg/wmf images
- embedding rtf-texts (font,...)
- generates xml output (xslfo)
- generates pdf-files (email)
- supports WebSpeed /-Client

## PCase

CASE-Extension for the  
Data Dictionary

- view Progress-DB structures
- create/update DBs directly
- reengineer Progress-DBs
- read/write Progress df-files
- compare/maintain versions
- incl. DB Content Viewer
- incl. Open Report Interface
- autogenerates references
- print resizable ER-Diagrams
- report-, structure- or ERD view

## QComp

Project management  
compiling, analyzing

- compiles project file lists
- includes compiler server
- also compiles in char-mode
- uses different Progress vers.
- compiles for different OS
- contains xref-analyze frontend
- shows db structure & content
- keeps track of project errors



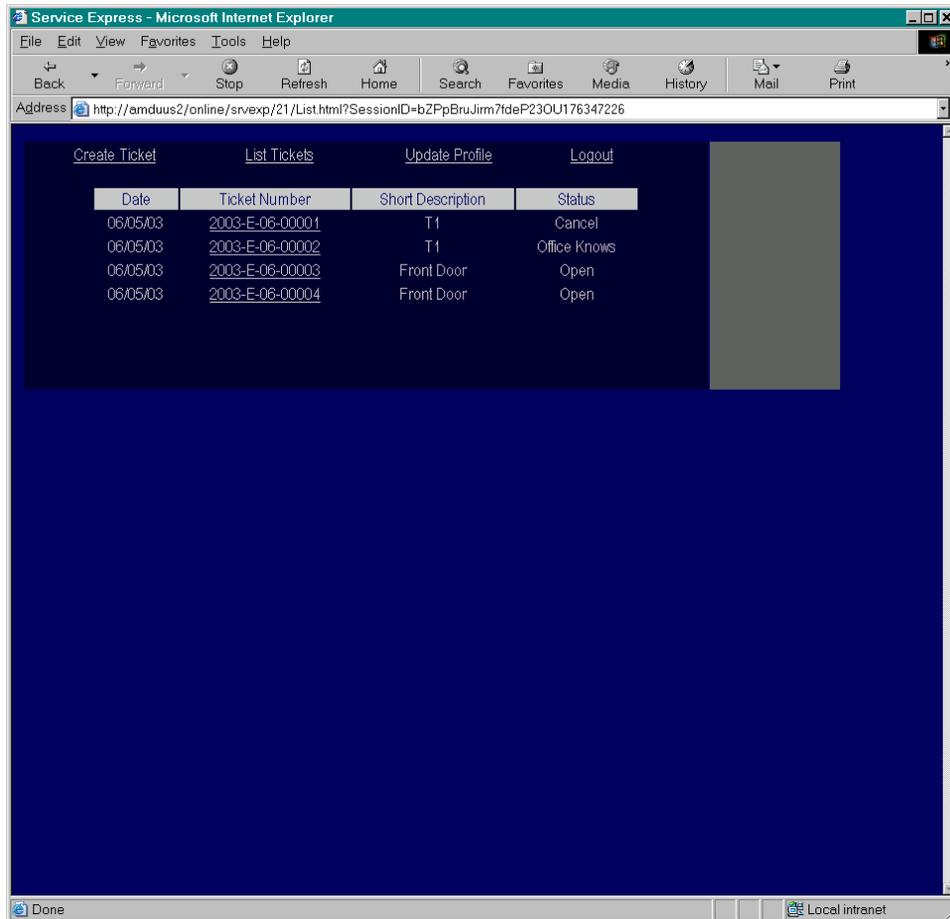
IAP GmbH • Moerkenstrasse 9 • D-22767 Hamburg • Germany  
Tel. +49 40 30 68 03 - 0 • Fax +49 40 30 68 03 - 10  
email: [info@tools4progress.com](mailto:info@tools4progress.com)



Information and free testversions at [www.tools4progress.com](http://www.tools4progress.com)

## Reference Partners Wanted

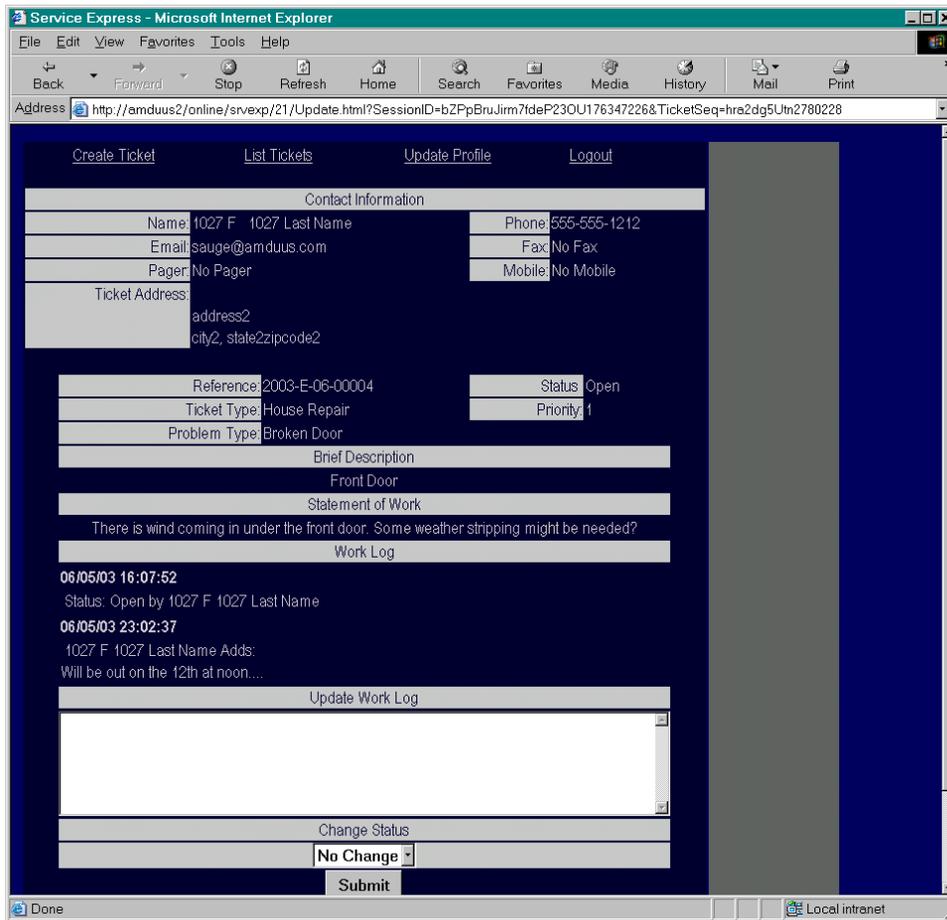
Amduus Information Works, Inc. finally has a working version of the Service Express available!



*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! Your welcome to use the software on your Webspeed/Blue Diamond installation or we can run the software for you. You can request instructions for downloading the software by writing to sauge@amduus.com.

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](mailto:sauge@amduus.com) if you have additional questions!

**Publishing Information:**

Scott Auge publishes this document. I can be reached at [sauge@amduus.com](mailto: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](mailto: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](mailto: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 operating 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 ProVition, WebSpeed Workshop, WebSpeed Messenger, and Open Client Toolkit products. Also includes a ProVition Plus Development Server. Note: ProVition 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 Websppeed (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 Websppeed/UNIX CHUI! More!
-----	---

Direct any questions to Scott Auge at [sauge@amduus.com](mailto: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
<p><b>Enterprise Level Database Server</b> Price per connection Circle your OS:</p> <ul style="list-style-type: none"> <li>• Tru64 UNIX 4.0F (4.0G, 5.0, 5.0A, 5.1, 5.1A, 5.1B)</li> <li>• HP-UX 32 Bit PA-RISC (Patches XSWG1100+ PHSS_22543, 22868, 22478, HPUX 11i)</li> <li>• HP-UX 64 Bit PA_RISC (Patches XSWG1100+ PHSS_22543, 22868, 22478, HPUX 11i)</li> <li>• IBM AIX (4.3.3 MUP Level 2 )</li> <li>• 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)</li> <li>• Sun Solaris SPARC 32Bit (Solaris SPARC 2.6 Solaris 7, 8, 9)</li> <li>• Sun Solaris SPARC 64 Bit (Solaris SPARC 2.8, Solaris 8, 9)</li> </ul>	\$1,022.00		
<p><b>Workgroup Level Database Server</b> Price per connection Circle your OS:</p> <ul style="list-style-type: none"> <li>• Tru64 UNIX 4.0F (4.0G, 5.0, 5.0A, 5.1, 5.1A, 5.1B)</li> <li>• HP-UX 32 Bit PA-RISC (Patches XSWG1100+ PHSS_22543, 22868, 22478, HPUX 11i)</li> <li>• HP-UX 64 Bit PA_RISC (Patches XSWG1100+ PHSS_22543, 22868, 22478, HPUX 11i)</li> <li>• IBM AIX (4.3.3 MUP Level 2 )</li> <li>• 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)</li> <li>• Sun Solaris SPARC 32Bit (Solaris SPARC 2.6 Solaris 7, 8, 9)</li> <li>• Sun Solaris SPARC 64 Bit (Solaris SPARC 2.8, Solaris 8, 9)</li> </ul>	\$350.00		
<p><b>Progress Client Network</b> Per User Circle your OS:</p> <ul style="list-style-type: none"> <li>• Tru64 UNIX 4.0F (4.0G, 5.0, 5.0A, 5.1, 5.1A, 5.1B)</li> </ul>	\$196.00		

<ul style="list-style-type: none"> <li>• HP-UX 32 Bit PA-RISC (Patches XSWGR1100+ PHSS_22543, 22868, 22478, HPUX 11i)</li> <li>• HP-UX 64 Bit PA_RISC (Patches XSWGR1100+ PHSS_22543, 22868, 22478, HPUX 11i)</li> <li>• IBM AIX (4.3.3 MUP Level 2 )</li> <li>• 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)</li> <li>• Sun Solaris SPARC 32Bit (Solaris SPARC 2.6 Solaris 7, 8, 9)</li> <li>• Sun Solaris SPARC 64 Bit (Solaris SPARC 2.8, Solaris 8, 9)</li> <li>• MS Windows</li> </ul>			
<p><b>Webspeed Transaction Server 5 Agent</b></p> <p>Circle Your OS:</p> <ul style="list-style-type: none"> <li>• Tru64 UNIX 4.0F (4.0G, 5.0, 5.0A, 5.1, 5.1A, 5.1B)</li> <li>• HP-UX 32 Bit PA-RISC (Patches XSWGR1100+ PHSS_22543, 22868, 22478, HPUX 11i)</li> <li>• HP-UX 64 Bit PA_RISC (Patches XSWGR1100+ PHSS_22543, 22868, 22478, HPUX 11i)</li> <li>• IBM AIX (4.3.3 MUP Level 2 )</li> <li>• 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)</li> <li>• Sun Solaris SPARC 32Bit (Solaris SPARC 2.6 Solaris 7, 8, 9)</li> <li>• Sun Solaris SPARC 64 Bit (Solaris SPARC 2.8, Solaris 8, 9)</li> </ul>	<p>\$7,560.00</p>		
<p><b>Webspeed Transaction Server 25 Agent</b></p> <p>Circle Your OS:</p> <ul style="list-style-type: none"> <li>• Tru64 UNIX 4.0F (4.0G, 5.0, 5.0A, 5.1, 5.1A, 5.1B)</li> <li>• HP-UX 32 Bit PA-RISC (Patches XSWGR1100+ PHSS_22543, 22868, 22478, HPUX 11i)</li> <li>• HP-UX 64 Bit PA_RISC (Patches XSWGR1100+ PHSS_22543, 22868, 22478, HPUX 11i)</li> <li>• IBM AIX (4.3.3 MUP Level 2 )</li> </ul>	<p>\$14,000.00</p>		

<ul style="list-style-type: none"> <li>• 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)</li> <li>• Sun Solaris SPARC 32Bit (Solaris SPARC 2.6 Solaris 7, 8, 9)</li> <li>• Sun Solaris SPARC 64 Bit (Solaris SPARC 2.8, Solaris 8, 9)</li> </ul>			
<p><b>Webspeed Transaction Server 50 Agent</b> Circle Your OS:</p> <ul style="list-style-type: none"> <li>• Tru64 UNIX 4.0F (4.0G, 5.0, 5.0A, 5.1, 5.1A, 5.1B)</li> <li>• HP-UX 32 Bit PA-RISC (Patches XSWGR1100+ PHSS_22543, 22868, 22478, HPUX 11i)</li> <li>• HP-UX 64 Bit PA_RISC (Patches XSWGR1100+ PHSS_22543, 22868, 22478, HPUX 11i)</li> <li>• IBM AIX (4.3.3 MUP Level 2 )</li> <li>• 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)</li> <li>• Sun Solaris SPARC 32Bit (Solaris SPARC 2.6 Solaris 7, 8, 9)</li> <li>• Sun Solaris SPARC 64 Bit (Solaris SPARC 2.8, Solaris 8, 9)</li> </ul>	\$25,200.00		
<p><b>Progress Webspeed Workshop (For the Programmer/Analyst)</b> Per User</p> <ul style="list-style-type: none"> <li>• MS Windows</li> </ul>	\$1,330.00		
<p><b>4GL Development System (For the Programmer/Analyst) Per User</b> Circle Your OS:</p> <ul style="list-style-type: none"> <li>• Tru64 UNIX 4.0F (4.0G, 5.0, 5.0A, 5.1, 5.1A, 5.1B)</li> <li>• HP-UX 32 Bit PA-RISC (Patches XSWGR1100+ PHSS_22543, 22868, 22478, HPUX 11i)</li> <li>• HP-UX 64 Bit PA_RISC (Patches XSWGR1100+ PHSS_22543, 22868, 22478, HPUX 11i)</li> <li>• IBM AIX (4.3.3 MUP Level 2 )</li> <li>• 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)</li> </ul>	\$3,920.00		

<ul style="list-style-type: none"> <li>• Sun Solaris SPARC 32Bit (Solaris SPARC 2.6 Solaris 7, 8, 9)</li> <li>• Sun Solaris SPARC 64 Bit (Solaris SPARC 2.8, Solaris 8, 9)</li> </ul>			
<b>Progress Provision Plus (For the Programmer/Analyst) Per User</b> Circle Your OS: <ul style="list-style-type: none"> <li>• MS Windows</li> </ul>	\$6,300.00		
Amduus Open Source CD-ROM Per Organization	Free!		Free!
			Tax
			Grand Total