The Progress Electronic Magazine

In this issue:

Publisher's Statement:

In this issue, we example the reporting tool called Denkh HTML Reporter. This is a simple tool that allows a user to create a vast array of reports of varying complexity. Since it is open source, and runs on the web, one only needs a simply licensed version of Webspeed to make use of the software. This license could handle tens to hundreds of users at a low cost compared to other solutions out there. Once the reports are set up, they are very easy to use. The article included will show some basics of creating these reports, as well some tricks to improve their complexity and ease of use.

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. My skills include law enforcement, manufacturing, and service oriented companies and their problems. I also offer Webspeed training.

Coding Article: Tips and Tricks To Creating Reports With Denkh HTML Reporter

By Scott Auge

Denkh HTML Reporter (DHR), a free and open source reporting tool, allows report designers to create reports on Webspeed without being Webspeed programmers. All report designers need to be familiar with are:

- 4GL Syntax queries
- Tables available in the application
- Other Reports that can act as building blocks for other reports
- HTML
- and/or CSS
- and/or XHTML
- Denkh HTML Reporter Macros

Report designers can mix HTML/CSS/XHTML with macros. These macro's can ask questions of the user, embed reports in reports, reference a field in a result set, format data, call out to 4GL programs, etc. I won't go into details about the functionality of DHR, there is other documentation that covers such things.

This article is about tips one can use to build a reporting site with DHR that is robust and includes something's learned by other's implementing DHR.

Tip One: Report Naming Conventions

As it stands now, Denkh's Report Repository Manager does not categorize reports, so naming reports are very important. *One should never use a comma as part of the report name.* Some reports may be pieces of other reports, or used to act as basic building blocks and naming will aid in remembering what these do. Spaces can be used, but are not recommended.

One should call a set of reports by a given prefix. For example, if your using DHR as a content management system, you might want to label reports beginning with CMS:

```
CMS.Banner
CMS.General.Footer
CMS.General.Header
```

If your reports are oriented to Sales using it, you might want to prefix it with descriptions that aid in sorting them out.

```
Sales.LookUpOrderByNumber
Sales.LookUpOrderByCustomer
Sales.LookUpCreditWorthiness
```

Embedded reports that are meant to be building blocks controlling colors of other reports might be named:

Color.*YourColorName*

For example, some reports controlling color throughout the site might be:

```
Color.PageColor
Color.Text.Basic
Color.Text.Standout
```

The same could be said of fonts, images, etc.

```
Image.Company.SmallLogo
Image.Company.LargeLogo
Image.RedBar
```

Tip Two: Setting up an embedded report to control colors

Creating reports with Denkh HTML Reporter is much like setting up a web site. One will need to decide such items as the colors, fonts, images and layouts one should include on the site. When encoding information in a report, one generally uses a tag such as this to control color:

```
<body bgcolor="#FFFFFF">
```

But what happens when you have multiple reports using this color, and it is decided to change the color? An embedded report can aid in this. Instead of the above syntax, use this:

```
<body bgcolor="@emb(Color.PageColor,)">
```

But how to set up a report called Color.PageColor to only return a color code? Doesn't a report require a header, footer, query, etc? No, actually they do not. One can set up a report called Color.PageColor, and in the Page Header section of the report, simply enter the color code WITHOUT any carriage returns. If there are no page footer, query, etc. these items will not be returned.

Tip Three: Setting up an embedded report to control fonts

Just as one can use an embedded report to control the colors used in the site, one can do the same with fonts. Try setting up a report called Font.Text.General with the page header containing the following (no carriage returns):

```
<font face="Arial, Helvetica, sans-serif"
color="@emb(FontColor,)">@inp(1)</font>
```

Note that this report not only looks up the color the font should be, but takes some input too! In other reports, you can call this report to format the text sent to it:

```
<td>@emb(Font.Text.General,This should be in the color and font defined in the
report!)</td>
```

or by something a little more complicated:

```
@emb(Font.Text.General,@fmt(@fld(Article.ArticleText),x(80)))
```

This would take the contents of the Article.ArticleText field from a record, format it to 80 characters, then pass that along to the embedded report which will set the font and color of that text for output to the browser. (This should give you an idea of the power Denkh HTML Reporter has.)

Should the font need to be changed, one only needs to change the report Font.Text.General to use different font's in the tags and all reports using that report for text formatting will be instantly changed.

Tip Four: Setting up embedded reports to act as the header and footer

Often when reports are put into an intranet portal, they have the same look and feel across the board. The company logo, or common hyperlinks to other locations in the portal may be desired.

Instead of tediously coding this into each and every report, one may want to combine elements into a header and footer embedded reports.

For example, one may have a body color set, a banner for the company name and logo, a section for the motto. By creating a report called PageHeader.General with this information in it's Page Header:

```
<html>
@emb(BaseBody,)
@emb(Banner.Amduus,)
<center>@emb(Motto,)</center>
```

you can change multiple reports header sections by including in it's Page Header:

```
@emb(PageHeader.General,)
```

So as example, should you change the Motto report, that will automatically be changed across the set of reports using PageHeader.General because it refers to the embedded report Motto.

For the page footer, you may want to create a report called PageFooter.General containing the following:

```
<center>@emb(AskSearch,)</center>
</body></html>
```

As you can see, it finishes off the <html> and <body> tags found in the PageHeader.General report. It also includes a little report that generates a prompt for an across site search.

Lets start exploring how to create some input boxes that are a bit more robust than the general text prompts provided by the base report engine!

Tip Five: Creating drop down boxes, check boxes, and radio boxes as embedded reports

As will be seen below, you can include forms in your reports to call out to other pages (or DHR reports!) But instead of using simple input boxes, how can one create a drop down box?

One can create a special embedded report that will return the HTML needed for a drop down box. For example, if you create a report called Input.DropDown.Category with the following:

Name: Input.DropDown.Category
Query: for each category by name
Page Header: <select name="@inp(1)">
Record Definition: <option value="@fld(category.name)">@fld(Category.Name)</option>
Page Footer: </select>

This embedded report would allow you to create a drop down box on the category table for use in your reports. For example, to test this, in the Page Header/Footer/Record/Summary of another report embed the report (shown in red):

```
<html><body>
<form method="post" action="HTMLRpt.html?RptTemplate1ID=1zWgV3eI781981084">
Category: @emb(Input.DropDown.Category,1)
<input type="submit">
<input type="hidden" name="NumberInputs" value="2">
</form></body>
```

This will create a form with a select box named “1” with the contents of the category table listed.

Lets explore some more about how to make forms in your reports that call out to other reports.

Tip Six: Embedding forms into your reports that call other reports

Lets say you want to be able to setup an search box on your web site. This is a very common need. Below is an example of how to accomplish this. It is in a report entitled AskSearch and this HTML code is stored in the "Page Header" section of the template:

```
<form method="post" action="HTMLRpt.html?RptTemplate1ID=1zWgV3eI781981084">
<input type="hidden" name="NumberInputs" value="2">
<p>@emb(BaseFont,Search Amduus:) <input type="text" name="2">
<input type="submit" value="Submit">
</form>
```

To do this, first you would need to identify the report's ID which can be found in the report editing screen.

Next create a set of form tags to create a form in your report. The Action should always post to HTMLRpt.html where the value in the NVP RptTemplate1ID is the ID of the report. This NVP could also be a hidden input tag also.

Note that there is a hidden field called NumberInputs. This should match the number of input's your report needs (@inp() or @plc() macros).

Next name your input widgets, 1, 2, 3, … n where each input widget matches the order @plc() appears in queries, and the associated number for the @inp() macros throughout the target report.

Tip Seven: How to pronounce Denkh and Amduus?

Denkh is a simple one. Pronounce it like this: "Den-kah."

Amduus is pronounced "Ah-m-dus" where dus is like the dus in dust but without the t. ☺ It is not pronounced "Am-do-s." Bad Bad Bad pronounciation! ☹

Tip Eight: Creating report "modules"

This is the main power of Denkh HTML Reporter and where true sweetness sets in. A lot of report generator's require the user's to know and understand the tables of the database the generator is working with. This is not necessarily required with Denkh HTML Reporter.

Once one has set up base reports that do all the querying and formatting of basic information, all that report designers need to know are these basic reports. They can act as building blocks for other reports and allow report designer's to focus on these instead of the application's tables.

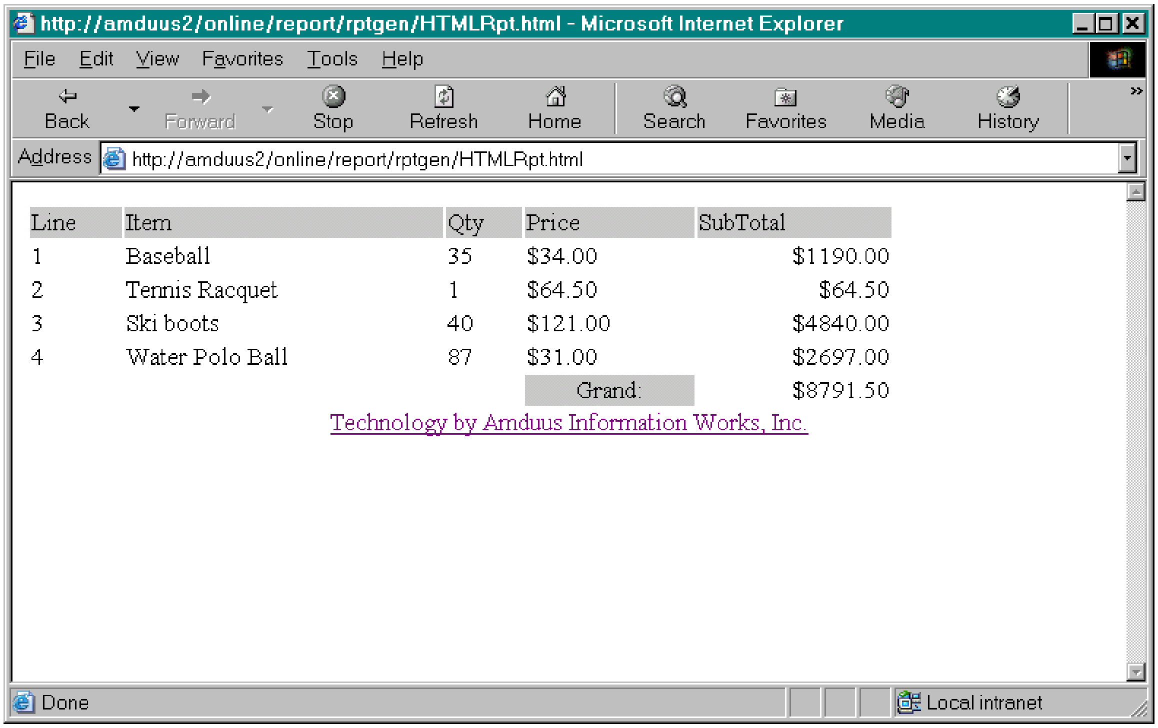
For example, a report designer might need to create a new report that includes some order header information. With a pre-made report that provides this, the designer only needs to know that the order number is the second argument to @emb(Sales.Order.Header,100) in their new report and all the table joins, query, and formatting are done for them.

One can abstract away from database tables by using embedded reports and drilled reports where the report designer only needs to know the names of pre-made reports and their basic input values.

Lets explore this. The summary of an an order's lines might be commonly used in other reports. So we would create a report called Emb.OrderLine. It would have the following set up for it's template:

Name:	Emb.OrderLine
Query:	FOR EACH OrderLine WHERE OrderLine.OrderNum = @plc(Enter Order Number:), FIRST Item OF OrderLine
Page Header:	<table width="80%" bgcolor="">
Column:	Line,Item,Qty,Price,SubTotal
Record:	<tr> <td>@fld(OrderLine.LineNum)</td> <td>@fld(Item.ItemName)</td> <td>@fld(OrderLine.Qty)</td> <td>@fmt(@fld(OrderLine.Price),$>>>9.99)</td> <td align="right">@calc(@fld(OrderLine.Qty) * @fld(OrderLine.Price),$>>999.99) @trksubtotal(Grand,@calc(@fld(OrderLine.Qty) * @fld(OrderLine.Price),>>999.99))</td> </tr>
Summary:	<tr> <td></td> <td></td> <td></td> <td align="center" bgcolor="#CCCCCC">Grand:</td> <td align="right">@rptsubtotal(Grand,$>>999.99)</td> </tr>
Page Footer:	</table>

This will create a report similar to this when given an order number:

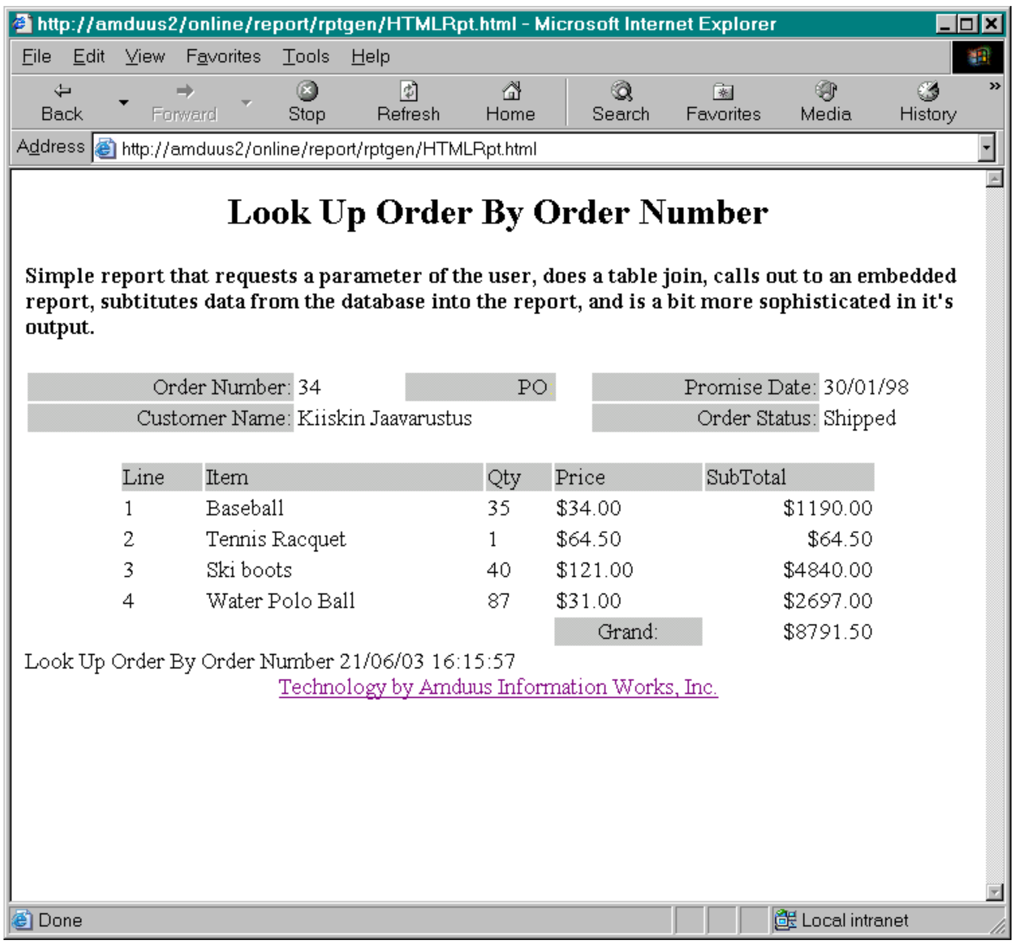


Now lets include this into a new report that will include some additional information. Note that the line items are listed via a @emb() macro in the record section (see blue highlighted area):

Name:	`Look Up Order By Order Number`
Query:	`FOR EACH Order WHERE Order.OrderNum = @plc(Enter Order Number), FIRST Customer OF Order`
Page Header:	<center> Look Up Order By Order Number </center> Simple report that requests a parameter of the user, does a table join, calls out to an embedded report, subtitutes data from the database into the report, and is a bit more sophisticated in it's output.
Column:	
Record:	<table width="100%" border="0"> <tr> <td bgcolor="#CCCCCC"> Order Number: </td> <td>@fld(Order.OrderNum)</td> <td bgcolor="#CCCCCC"> PO: </td> <td>@fld(Order.PO)</td> <td bgcolor="#CCCCCC"> Promise Date: </td> <td>@fld(Order.PromiseDate)</td> </tr>

	<tr> <td bgcolor="#CCCCCC"> Customer Name: </td> <td colspan="3">@fld(Customer.Name)</td> <td bgcolor="#CCCCCC"> Order Status: </td> <td>@fld(Order.OrderStatus)</td> </tr> </table> **<center>@emb(Emb.OrderLin,@fld(Order.OrderNum))</center>**
Summary:	
Page Footer:	</table> @rptname() @date() @time()

When this report is run, it will result with:



Can you see how the report designer, with one simple line of code and without knowledge of the database tables associated with an order's line items has created a report containing the line items of an order?!?!

One could certainly create an Emb.OrderHeader report with the top level information associate with it, and then create reports based on that.

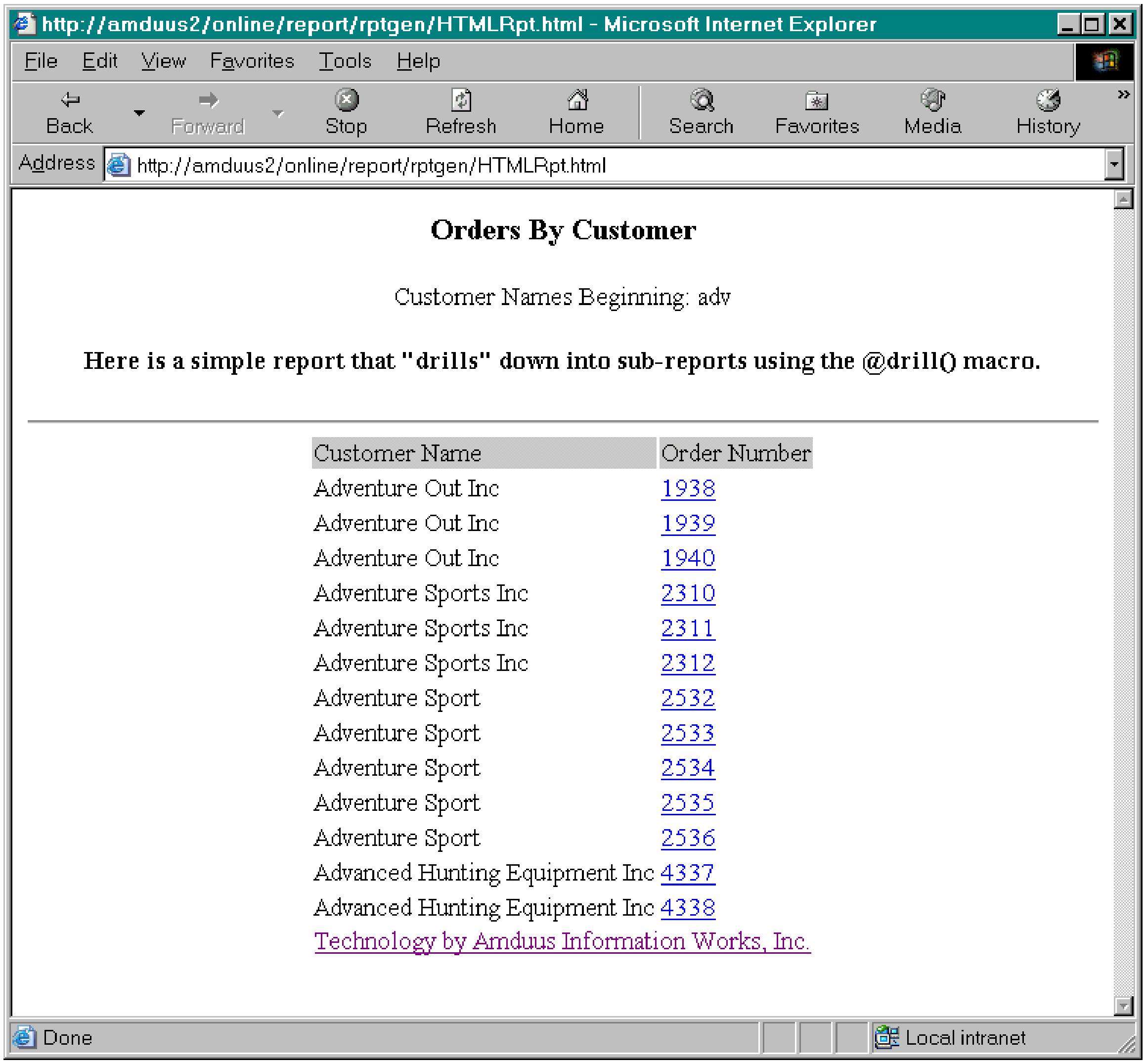
Tip Nine: Creating reports that drill into other reports

So now we have a set of reports that provide some header and line information about a customer's order. Now the request has come in to look up a set of customers by their name, and to be able to examine their orders.

Lets create the following report that makes use of the @drill() macro that creates a hyperlink into a report for orders – you guessed it – by requiring the report designer to only know the order number! Those called and embedded reports will take care of the rest.

Name:	Orders By Customer
Query:	for each customer where customer.name begins "@plc(Name Begins:)", each order of customer by order.ordernum
Page Header:	<center> Orders By Customer </center> <html> Customer Names Beginning: @inp(1) Here is a simple report that "drills" down into sub-reports using the @drill() macro. <hr> <table>
Column:	Customer Name,Order Number
Record:	<tr> <td> @fld(Customer.Name) </td> <td> **@drill(Look Up Order By Order Number,@fld(Order.OrderNum),_blank,@fld(Order.OrderNum))** </td> </tr>
Summary:	
Page Footer:	</table> </html>

Creates a report that appears as follows:

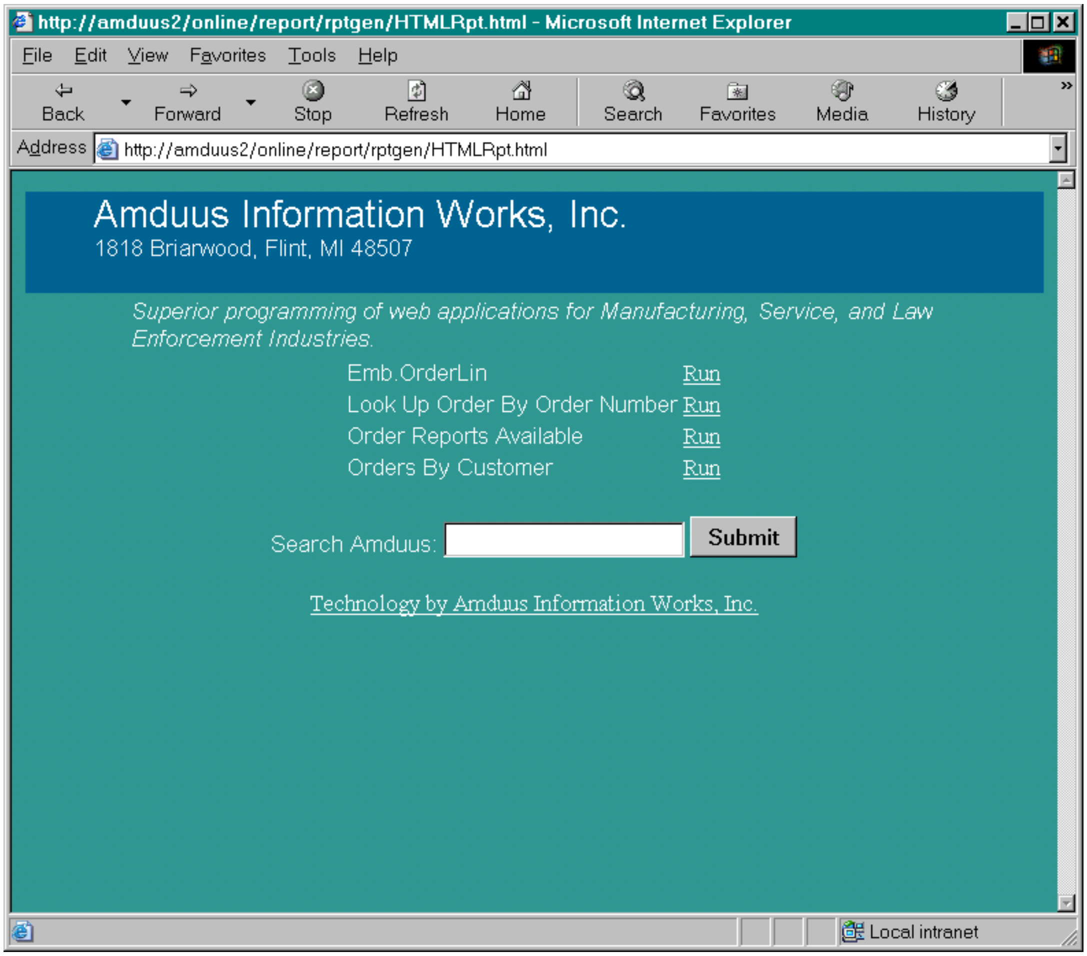


When the hyperlink is clicked, the Order is presented as in previous screens.

Tip Ten: Creating your own report listing screen

Sometimes you may want to aid the user's by listing only reports that might be of interest to them – or to include the report's listing in a format more congruent to your web environment. The base report manager that comes with Denkh is very bare (for reason – you have the code – you can update it to appear as you wish.)

Below is a definition using the formatting rules of the web environment for Amduus, as well listing reports oriented towards Order lookup.



This is accomplished with the following report definition:

Name:	Order Reports Available
Query:	for each rpttemplate1 where index(rpttemplate1.name, "order") > 0 by rpttemplate1.name
Page Header:	<html> @emb(BaseBody,) @emb(Banner.Amduus,) <center>@emb(Motto,)</center> <table align="center">
Column:	
Record:	<tr> <td>@emb(BaseFont,@fld(RptTemplate1.Name)) </td> <td> Run</td>

	`</tr>`
Summary:	

Page Footer:	`</table>`
	`<center>@emb(AskSearch,)</center>`
	`</body></html>`

Note how if a report needs to prompt for input, you should call RunHTMLRpt.html (unless you have a report that creates prompts for a report already.)

Where is Denkh HTML Reporter going?

As you can probably tell – Denkh HTML Reporter is practically a replacement for those parts of a Webspeed application program that inquire and lookup information in database tables. One can certainly stitch together two or more web programs with DHR. By using <forms> and hyperlinks in your reports, you can send information to other pages outside DHR. Those other parts doing database updates might even have a different application infrastructure below it!

I am sure as more uses are found for it, additional macro's will be needed. Some that have been requested include @urlencode() and @htmlencode().

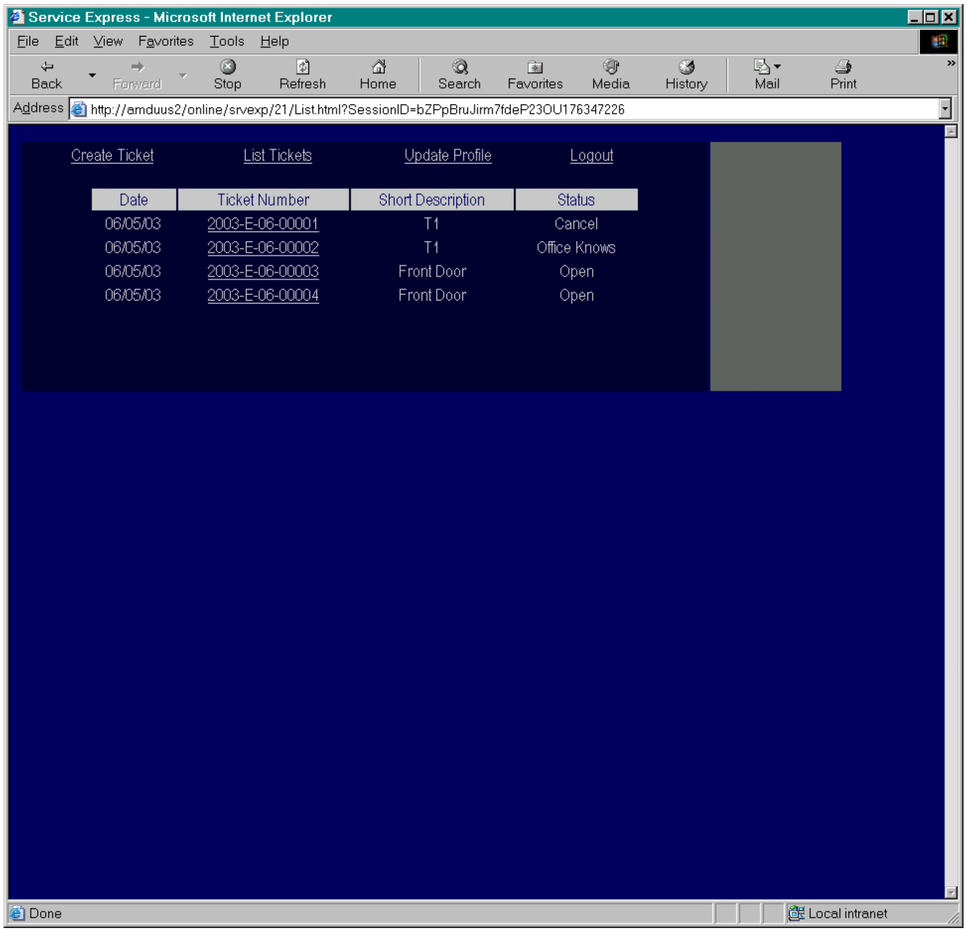
The most desired next steps would be some built in reports for various Progress oriented applications, such as QAD's Mfg/Pro, some of the accounting programs out there, Symix, etc. If you would like to pay me to aid in the creation of these reports – I am certainly game for that. If you have some made from your own installation, please be sharing with the community – you never know what you might get back!

One of the next steps under consideration will be to create macros that actually create and update records in a database. Of course, one can already use the @prg() macro to call 4GL procedures to do such a thing already.

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

Reference Partners Wanted

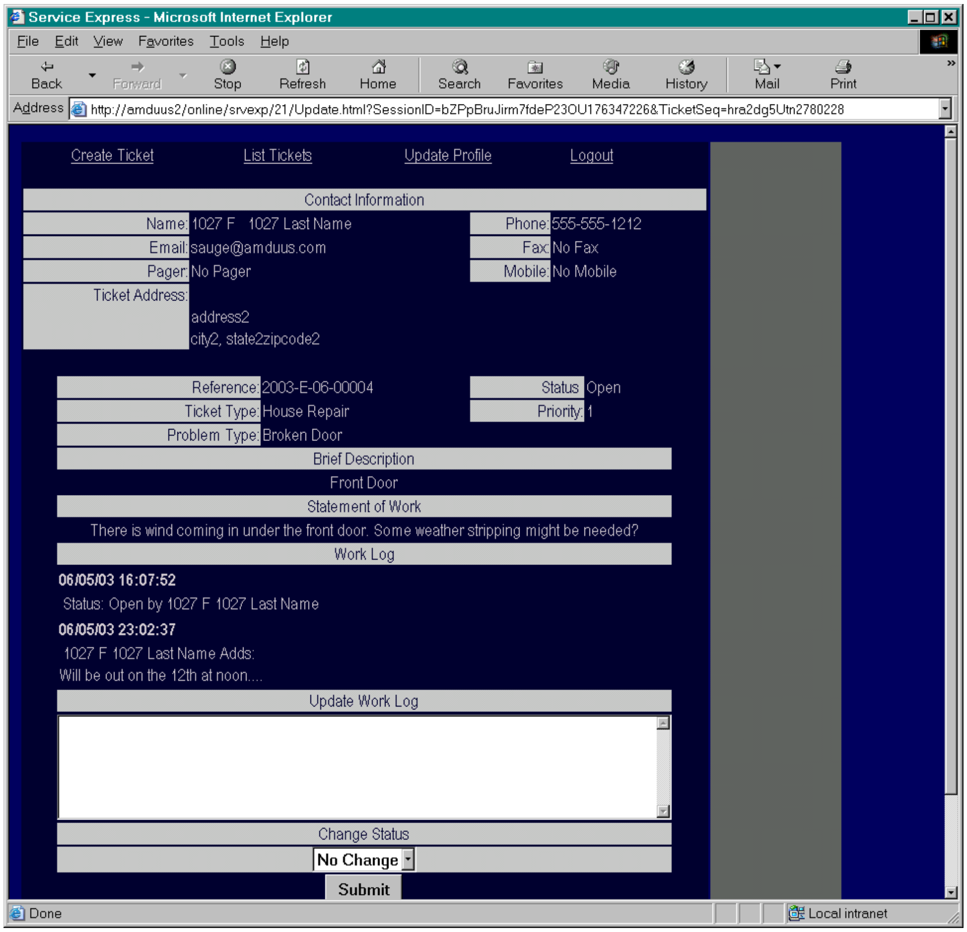
Amduus Information Works, Inc. finally has a working version of the Service Express available!



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 it available to five companies/organizations that might be willing to offer suggestions for improvement and are willing to be referenced as user's of the software.

The company/organization can be governmental, non-profit, private, or a corporation.

Qualification Questions For Free Access To Service Express	
Market Served By Your Company?	
Is your organization non-profit?	Yes No
Is your organization a corporation?	Yes No
Would external customers use your site?	Yes No
Would internal customers use your site?	Yes No
Would the site be integrated into an existing site?	Yes No
Types of problems the site would aid in managing?	
Contact Person's Name	
Contact Person's Phone	
Contact Person's E-Mail Address	
Would your company be willing to act as a reference for the software?	
Would you be able to spend a day or so to learn how to and to perform configurations for the software?	
Would you be willing to pay a nominal fee to have Amduus configure the software?	
Estimated number of tickets created per day?	
Country of Company/Organization? *Currently Service Express is available only in English*	

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:

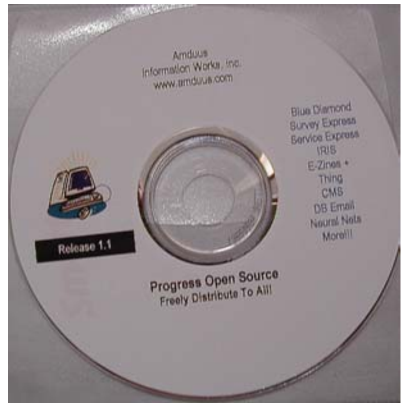
Please submit your article in Microsoft Word format or as text. Please include a little bit about yourself for the About the Author paragraph.

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!