

The Progress Electronic Magazine

In this issue:

| | |
|--|-----------|
| Publisher's Statement: | 2 |
| Coding Article: Natural Language User Interface and Adventure Game Part III | 3 |
| Theory | 3 |
| TakeItem.p..... | 8 |
| DropItem.p | 10 |
| Inventory.p | 13 |
| Additional Words and Verbs | 14 |
| RPGItem Database Table | 14 |
| Coding Article: Web Based Parameter Maintenance Screen | 15 |
| ParameterMaintenance.html..... | 19 |
| NewParameter.html | 23 |
| HTMLSelect.i..... | 24 |
| Coding Article: Poor Man's AppServer | 25 |
| Architecture | 26 |
| Using the Client API to request a PDF..... | 26 |
| Administration of the Denkh PDFSvr process | 28 |
| PDFSvr.ksh..... | 28 |
| PDFSvr.p | 28 |
| PDFSvrPut.p..... | 30 |
| PDFSvrWatch.p..... | 31 |
| PDFSvrGet.p | 32 |
| Publishing Information: | 33 |
| Other Progress Publications Available: | 33 |
| Products Available From Amduus: | 33 |
| Article Submission Information: | 33 |
| Order Form for Progress Open Source CD-ROM | 35 |

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.

© Amduus Information Works, Inc. 2003

Publisher's Statement:

This issue we finally get to the web based parameter system. It is an example of how to set up the values in one set of fields, based on the user's choice in another field. It is useful for web or background processes to set up dynamic and changeable parameters of operation and behavior from your application. If used with say, smtpmail.p, such things as where to find mail servers, from address, etc. can be set up using a parameter; Instead of being hard coded or pre-processor defined. It also will make it somewhat easier for an "application administrator" to modify the behavior of an application. Add it to your application today!

In this issue we further explore natural language interfacing to Progress applications. We do this in a fun way by adding to our adventure game. Continuing using "baby steps" we look at how to allow the player to use natural language to interact with items in the game. Such things as picking up items, dropping items, and noting the inventory the player has will be explored.

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

Your ad could be here!

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

We also look at a "Poor Man's Data Server." We explore how can we get operations to work on another server based on a request from a remote client. Often this is useful for reports or operations that are simply too long to run in a web window, or can be run asynchronously with the user's program. We use example code from the Denkh PDF tool set that accepts messages from various client types (GUI, CHUI, and WWW) and performs some work on another computer. Modification of the code to run different kinds of operations asynchronous to the client can be made easily in the code. One could set up a message type to a module of code, and have the server back ground process run sets of code based on the message sent. In this example, it is the execution of programs available on Linux that are not readily available on Windows.

!!!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 for more information.

To your success,

Scott Auge

Founder, Amduus Information Works, Inc. sauge@amduus.com

Coding Article: Natural Language User Interface and Adventure Game Part III

Written by Scott Auge sauge@amduus.com

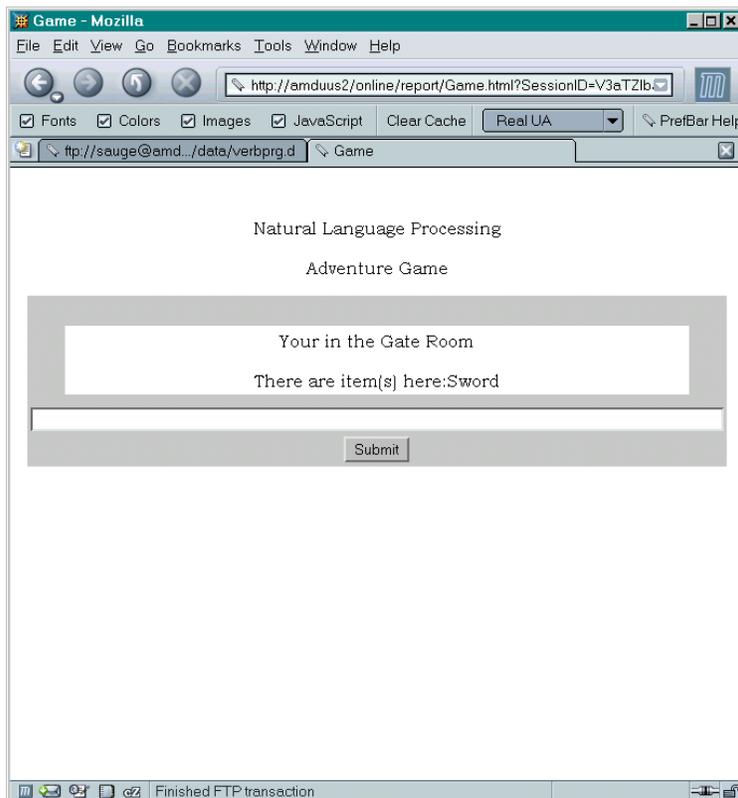
Once again we explore the use of a natural language interface for a user to interact with a program. In this section we explore the basic manipulation of items, that is the taking and dropping of these items. We also implement a command to remind the player what inventory they have available.

Theory

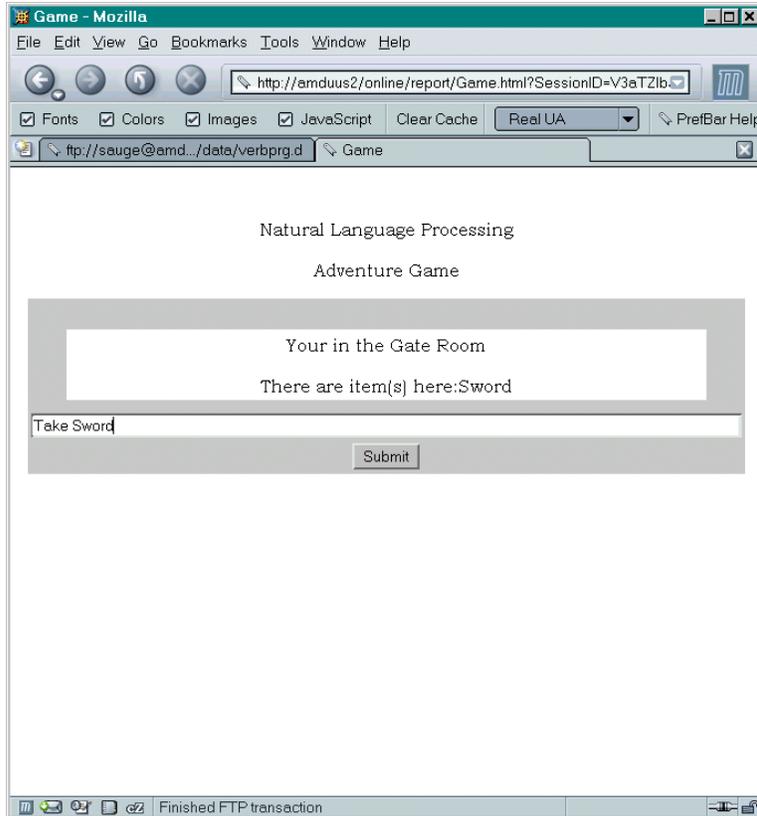
We pretty much do as we have with other commands. We identify some verbs and the programming associated with them. We introduce some nouns that the programs will work with in respect to the command the user gave.

Going through the steps

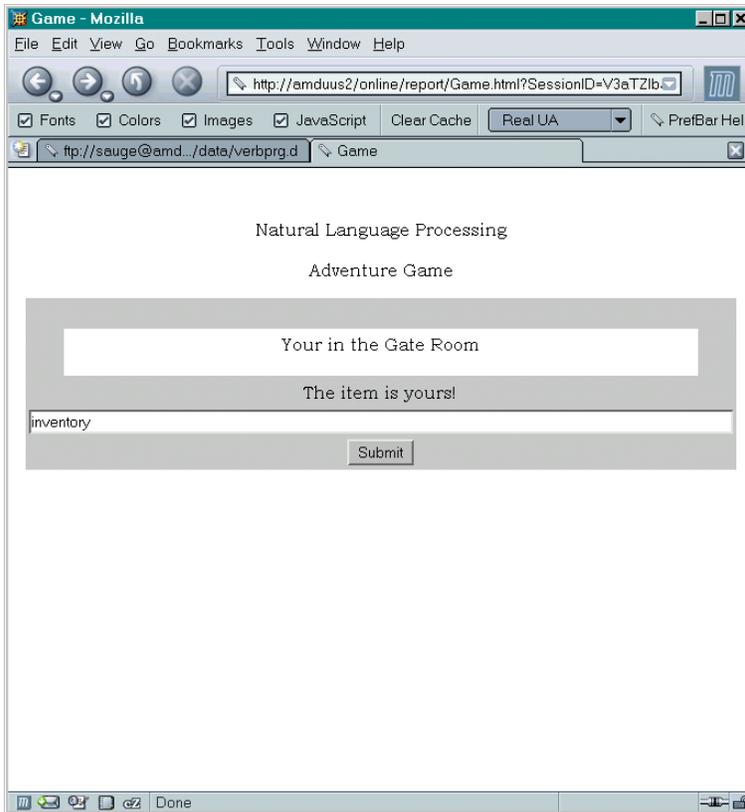
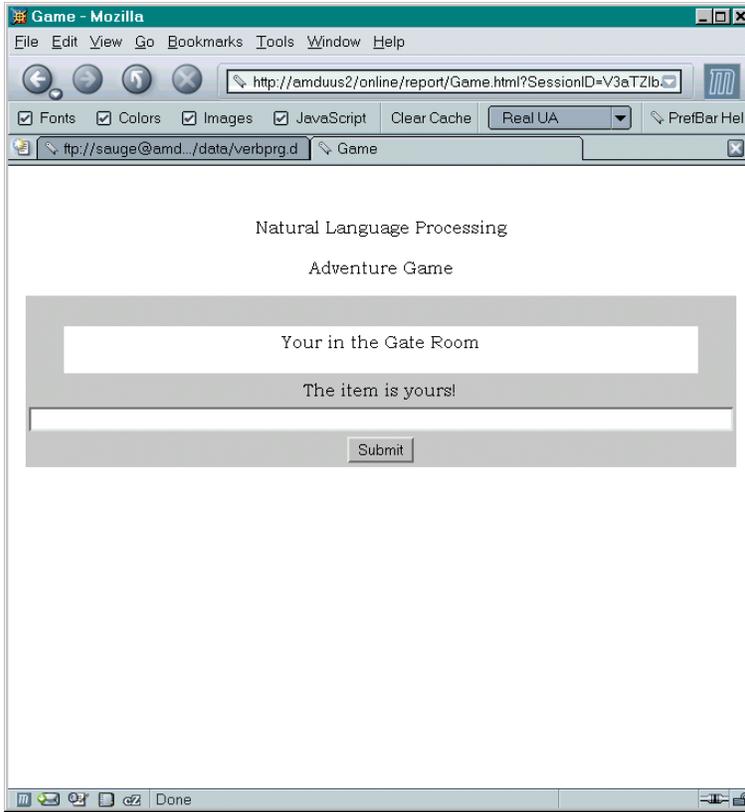
Here are some screen prints of the application in action with regards to working with items.

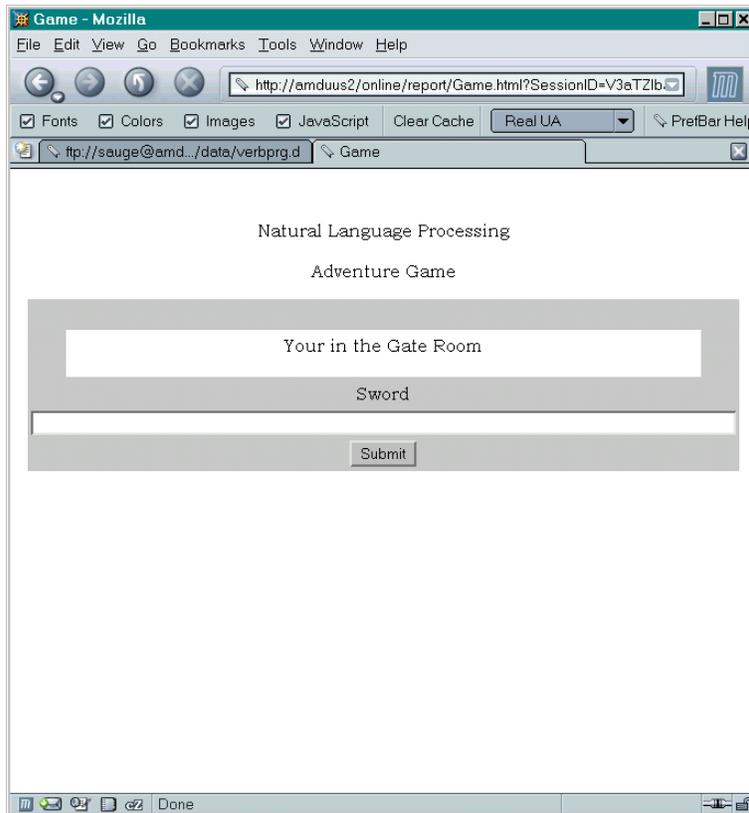


When the player enters the room, and there are items they can manipulate, they will appear. Some items may not be manipulated – but more investigation can be made of them. (Not implemented yet.) That way items such as plaques and the like can be read or observed.

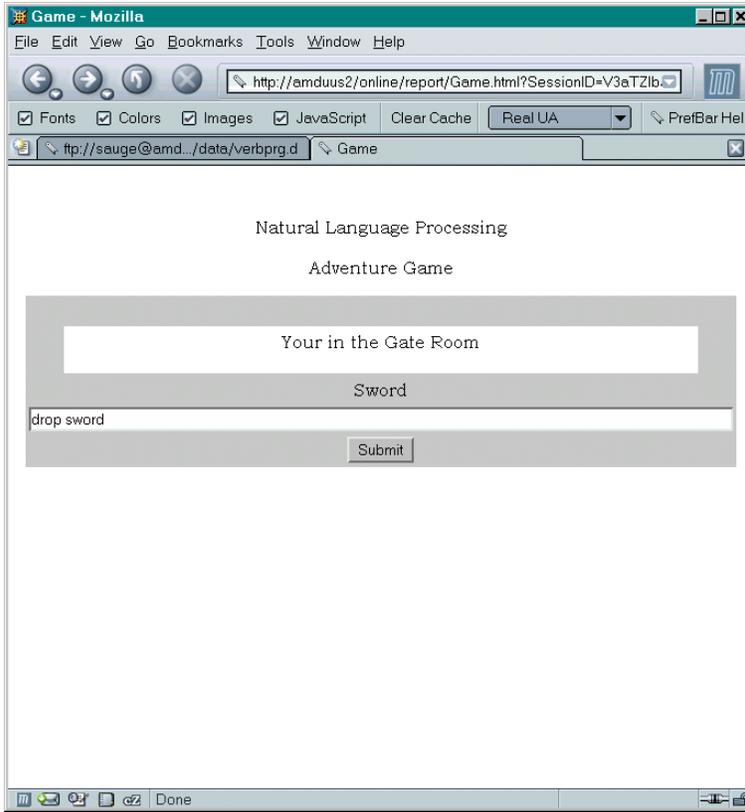


Here we introduced the verb “Take” – one could also include “get,” “obtain,” etc. as synonyms by simply updating the Verb table.





Note that the inventory command puts the player's inventory in the messages section.



TakeItem.p

```
/*
 * 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/appl/rpg/src/verb/TakeItem.p,v 1.1 2003/02/10 05:14:37 sauge Exp sauge $"
NO-UNDO.

DEF INPUT PARAMETER cSessionID AS CHARACTER NO-UNDO.

DEF VAR iCurrentRoom AS INTEGER NO-UNDO.
DEF VAR cDescription AS CHARACTER NO-UNDO.
DEF VAR cCommand AS CHARACTER NO-UNDO.
DEF VAR cError AS CHARACTER NO-UNDO.

/* Find the current room the user is in */

FIND WebState NO-LOCK
WHERE WebState.SessionID = cSessionID
AND WebState.Category = "Map"
AND WebState.Name = "CurrentRoom"
NO-ERROR.

ASSIGN iCurrentRoom = INT(WebState.Data).

/* Pull up what the user sent to us. */

FIND WebState NO-LOCK
WHERE WebState.SessionID = cSessionID
```

```
AND WebState.Category = "FeedBack"
AND WebState.Name     = "Command"
NO-ERROR.

ASSIGN cCommand = WebState.Data.

/* Lets see if we can figure out what RPGItem the user is speaking of. */
/* We do this by looping through all words that are type "NOUN" and */
/* see if they match anything in our cCommand. Note that not all */
/* NOUN words need to be RPGItems, but if they are, the need cor- */
/* responding RPGItem records. */

FOR EACH WordType NO-LOCK
WHERE WordType.WordType = "NOUN":

    IF INDEX(cCommand, WordType.Word) <> 0 THEN DO:

        ASSIGN cDescription = WordType.Word.
        LEAVE.

    END. /* IF INDEX(cCommand, WordType.Word) <> 0 */

END. /* FOR EACH WordType */

IF cDescription = "" THEN DO:

    RUN WriteState.p (INPUT cSessionID,
                     INPUT "FeedBack",
                     INPUT "Message",
                     INPUT "Not sure what you speak of....",
                     OUTPUT cError).

    RETURN.

END. /* IF cDescription = "" */

/* Find the corresponding RPGItem record for the NOUN */

FIND RPGItem EXCLUSIVE-LOCK
WHERE RPGItem.Description = cDescription
NO-ERROR.

IF NOT AVAILABLE RPGItem THEN DO:

    RUN WriteState.p (INPUT cSessionID,
                     INPUT "FeedBack",
                     INPUT "Message",
                     INPUT "Not sure what you speak of....",
                     OUTPUT cError).

    RETURN.

END. /* IF NOT AVAILABLE RPGItem */

/* Determine if the RPGItem is actually in the room */

IF RPGItem.RoomNumber <> iCurrentRoom THEN DO:

    RUN WriteState.p (INPUT cSessionID,
                     INPUT "FeedBack",
                     INPUT "Message",
                     INPUT "Item is not in this room!",
                     OUTPUT cError).
```

```
RETURN.
END. /* IF RPGItem.RoomNumber <> iCurrentRoom */
/* Determine if the RPGItem is takeable */
IF NOT RPGItem.Takeable THEN DO:
    RUN WriteState.p (INPUT cSessionID,
                     INPUT "FeedBack",
                     INPUT "Message",
                     INPUT "You can't take that!",
                     OUTPUT cError).
RETURN.
END. /* IF NOT RPGItem.Takeable */
/* Move the RPGItem into the user's possession */
FIND WebState NO-LOCK
WHERE WebState.SessionID = cSessionID
    AND WebState.Category = "Player"
    AND WebState.Name = "Login"
NO-ERROR.
ASSIGN
RPGItem.OnPerson = YES
RPGItem.RoomNumber = -1
RPGItem.PersonID = WebState.Data.
/* Reply back to the user they have it. */
RUN WriteState.p (INPUT cSessionID,
                 INPUT "FeedBack",
                 INPUT "Message",
                 INPUT "The item is yours!",
                 OUTPUT cError).
```

DropItem.p

```
/*
 * 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/appl/rpg/src/verb/DropItem.p,v 1.1 2003/02/10 05:14:37 sauge Exp sauge $"
NO-UNDO.

DEF INPUT PARAMETER cSessionID AS CHARACTER NO-UNDO.

DEF VAR iCurrentRoom AS INTEGER NO-UNDO.
DEF VAR cCommand AS CHARACTER NO-UNDO.
DEF VAR cDescription AS CHARACTER NO-UNDO.
DEF VAR cError AS CHARACTER NO-UNDO.

/* Pull up what the user sent to us. */

FIND WebState NO-LOCK
WHERE WebState.SessionID = cSessionID
AND WebState.Category = "FeedBack"
AND WebState.Name = "Command"
NO-ERROR.

ASSIGN cCommand = WebState.Data.

/* Lets see if we can figure out what RPGItem the user is speaking of. */
/* We do this by looping through all words that are type "NOUN" and */
/* see if they match anything in our cCommand. Note that not all */
/* NOUN words need to be RPGItems, but if they are, the need cor- */
/* responding RPGItem records. */

FOR EACH WordType NO-LOCK
WHERE WordType.WordType = "NOUN":

    IF INDEX(cCommand, WordType.Word) <> 0 THEN DO:

        ASSIGN cDescription = WordType.Word.
        LEAVE.

    END. /* IF INDEX(cCommand, WordType.Word) <> 0 */

END. /* FOR EACH WordType */

IF cDescription = "" THEN DO:

    RUN WriteState.p (INPUT cSessionID,
                     INPUT "FeedBack",
                     INPUT "Message",
                     INPUT "Not sure what you speak of..."),

```

```
        OUTPUT cError).

RETURN.

END. /* IF cDescription = "" */

FIND RPGItem EXCLUSIVE-LOCK
WHERE RPGItem.Description = cDescription
NO-ERROR.

IF NOT AVAILABLE RPGItem THEN DO:

    RUN WriteState.p (INPUT cSessionID,
                     INPUT "FeedBack",
                     INPUT "Message",
                     INPUT "Error: Cannot find the item record! " +
cDescription,
                     OUTPUT cError).

RETURN.

END. /* IF NOT AVAILABLE RPGItem */

/* Find out if the user has the item */

FIND WebState NO-LOCK
WHERE WebState.SessionID = cSessionID
    AND WebState.Category = "Player"
    AND WebState.Name = "Login"
NO-ERROR.

IF RPGItem.PersonID <> WebState.Data THEN DO:

    RUN WriteState.p (INPUT cSessionID,
                     INPUT "FeedBack",
                     INPUT "Message",
                     INPUT "You have no such item...",
                     OUTPUT cError).

RETURN.

END.

/* Find out the room the user is in */

FIND WebState NO-LOCK
WHERE WebState.SessionID = cSessionID
    AND WebState.Category = "Map"
    AND WebState.Name = "CurrentRoom"
NO-ERROR.

ASSIGN iCurrentRoom = INT(WebState.Data).

/* Leave the item in the room */

ASSIGN
RPGItem.OnPerson = NO
RPGItem.RoomNumber = iCurrentRoom
RPGItem.PersonID = "".

/* Feed back to the user. */

RUN WriteState.p (INPUT cSessionID,
```

```

INPUT "FeedBack",
INPUT "Message",
INPUT "You have dropped the item...",
OUTPUT cError).

```

Inventory.p

```

/*
 * 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 INPUT PARAMETER cSessionID AS CHARACTER NO-UNDO.

DEF VAR cMessage AS CHARACTER NO-UNDO.
DEF VAR cError AS CHARACTER NO-UNDO.

/* Figure out who is calling this routine */

FIND WebState NO-LOCK
WHERE WebState.SessionID = cSessionID
AND WebState.Category = "Player"
AND WebState.Name = "Login"
NO-ERROR.

/* Put thier inventory into the message back to them */

FOR EACH RPGItem NO-LOCK
WHERE RPGItem.PersonID = WebState.Data:

```

```

    ASSIGN cMessage = cMessage
                   + RPGItem.Description
                   + " ".
END. /* FOR EACH RPGItem */

RUN WriteState.p (INPUT cSessionID,
                  INPUT "FeedBack",
                  INPUT "Message",
                  INPUT cMessage,
                  OUTPUT cError).

```

Additional Words and Verbs

```

"Sword" "NOUN"
"Gilted Sword" "NOUN"
"TAKE" "VERB"
"DROP" "VERB"
"INVENTORY" "VERB"

"TAKE" "verb/TakeItem.p"
"DROP" "verb/DropItem.p"
"INVENTORY" "verb/Inventory.p"

```

These items are available in the total source package found under the rpg directory at www.amduus.com.

Note that all the source code is not listed here. Some files were changed that are not listed. Be sure to pick up the latest release of all the source code to catch these changes.

RPGItem Database Table

09/02/03 20:31:42 PROGRESS Report
 Database: amduus (PROGRESS)

=====
 ===== Table: RPGItem =====

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

| Table Name | Dump Name | Table Flags | Field Count | Index Count | Table Label |
|------------|-----------|-------------|-------------|-------------|-------------|
| RPGItem | item | | 6 | 2 | Item |

Description: Items available in a role playing game
 Storage Area: Schema Area

```
===== FIELD SUMMARY =====
===== Table: RPGItem =====
```

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

| Order | Field Name | Data Type | Flags | Format | Initial |
|-------|-------------|-----------|-------|--------------|---------|
| 10 | ItemID | char | i | x(8) | |
| 20 | OnPerson | logi | | yes/no | no |
| 30 | PersonID | char | | x(8) | |
| 40 | RoomNumber | inte | i | ->, >>>, >>9 | 0 |
| 50 | Description | char | | x(8) | |
| 60 | Takeable | logi | | yes/no | no |

| Field Name | Label | Column Label |
|-------------|-------------|--------------|
| ItemID | ItemID | ItemID |
| OnPerson | OnPerson | OnPerson |
| PersonID | PersonID | PersonID |
| RoomNumber | RoomNumber | RoomNumber |
| Description | Description | Description |
| Takeable | Takeable | Takeable |

```
===== INDEX SUMMARY =====
===== Table: RPGItem =====
```

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

| Flags | Index Name | Cnt | Field Name |
|-------|------------|-----|--------------|
| | key1 | 1 | + RoomNumber |
| pu | pukey | 1 | + ItemID |

```
** Index Name: key1
Storage Area: Schema Area
** Index Name: pukey
Storage Area: Schema Area
```

```
===== FIELD DETAILS =====
===== Table: RPGItem =====
```

Coding Article: Web Based Parameter Maintenance Screen

Written by Scott Auge sauge@amduus.com

Often programs will need to interact with other programs. An example is a progress program using smtpmail.p to interact with a mail delivery agent (MDA) on another machine. How can we easily identify to the program where to find the MDA? What from address should be used if it is not the user themselves (such as no-reply@yourdomain.com.)



```
def var x as int init 1.  
    x eq 12.  
display x.
```

If you were using Prolint, you wouldn't have spent the last two hours tracking down this bug.

Proparse Lite - Only US\$65

Lint your code before check-in.

Find bugs before your customer does.

www.joanju.com

Perhaps we want the program to place its log files in a particular directory? How can we easily manipulate this behavior? Perhaps at another time, the system administrator will prefer the log file end up in a different directory.

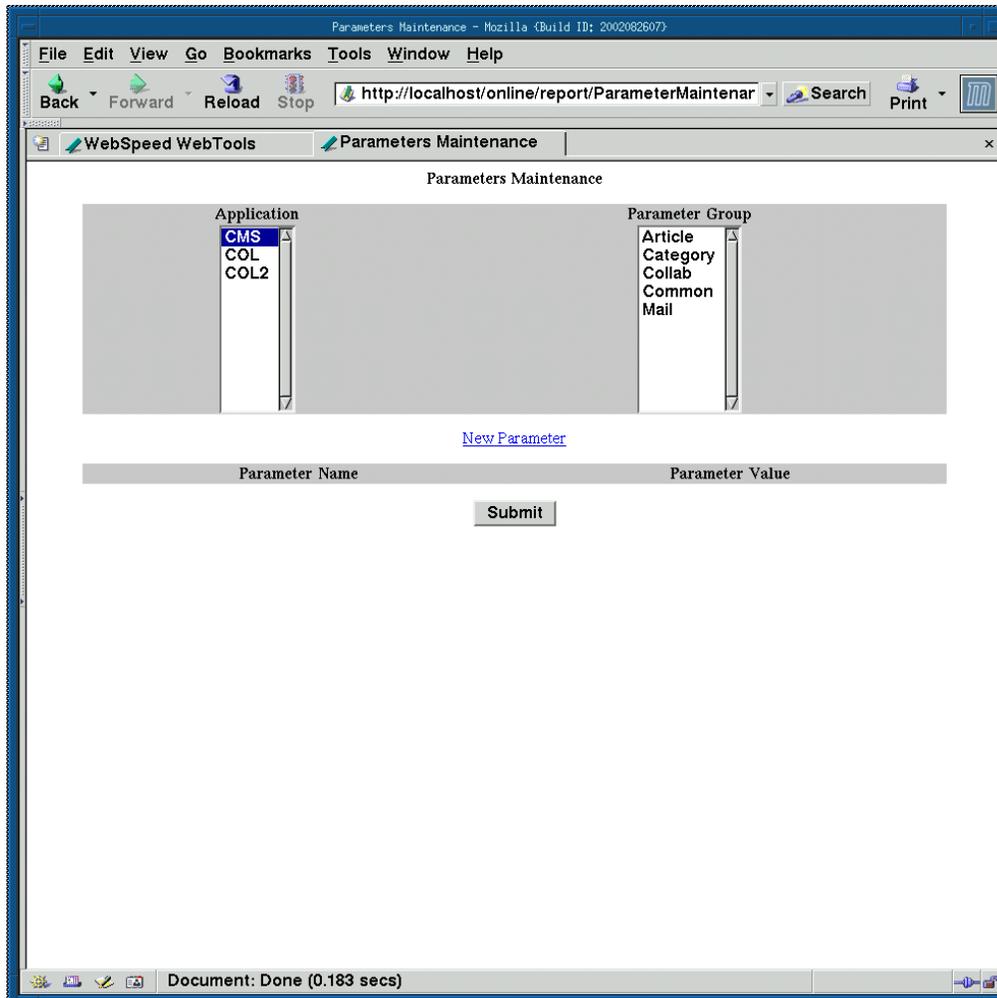
Hard coding these values is possible, but not convenient. When a change is required, a programmer will need to sift through the source code making changes here and there, and then doing a recompile and re-install of the application.

One could also set up a particular pre-processor file to hold these values, but

again a programmer will be needed to modify the include and then perform a re-compile and re-install of the application.

Another option is to use a .conf file (see issue 3 and issue 6 for example code and explanation.) But then one needs to log directly on the machine, and know where to find such a file.

In this issue, we make a more user friendly version of a parameter maintenance functionality. A web page.

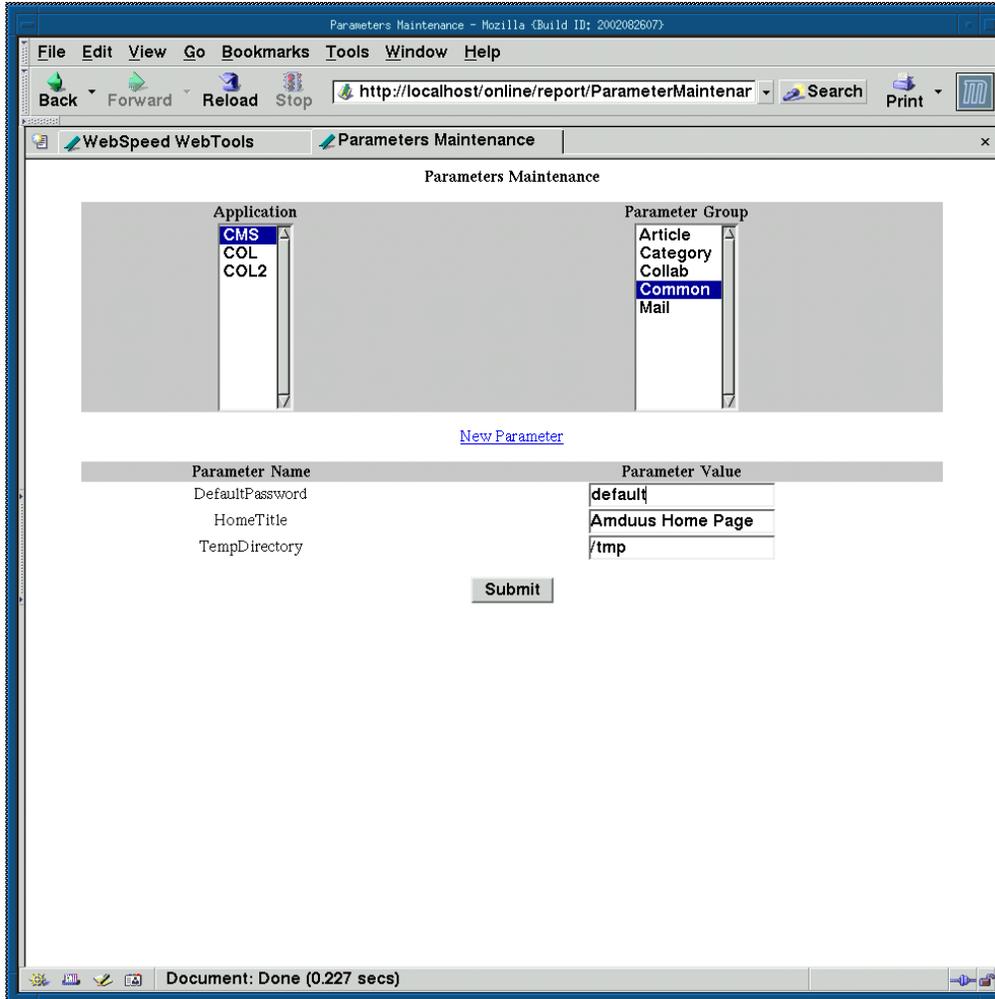


Clicking on the CMS application brings up the parameter groups found under CMS.

The user would log into an application and access this web page. Parameters are organized into application, parameter group, and finally a parameter name. Application would be a way to organize the parameter via module in a large ERP type system. A parameter group could be for a certain web page using parameters, or a more generic collection of parameters. Finally the parameter has a name.

Analysts Express, Inc.
Webspeed Training and progress programming.
Call James Arnold at
888-889-9091
or
jarnold@mylinuxisp.com

The user would click on the desired application for the parameter. This will cause the screen to refresh, using the selected application to find the group names under that value. The user will then click on the group name and a list of parameters will appear.

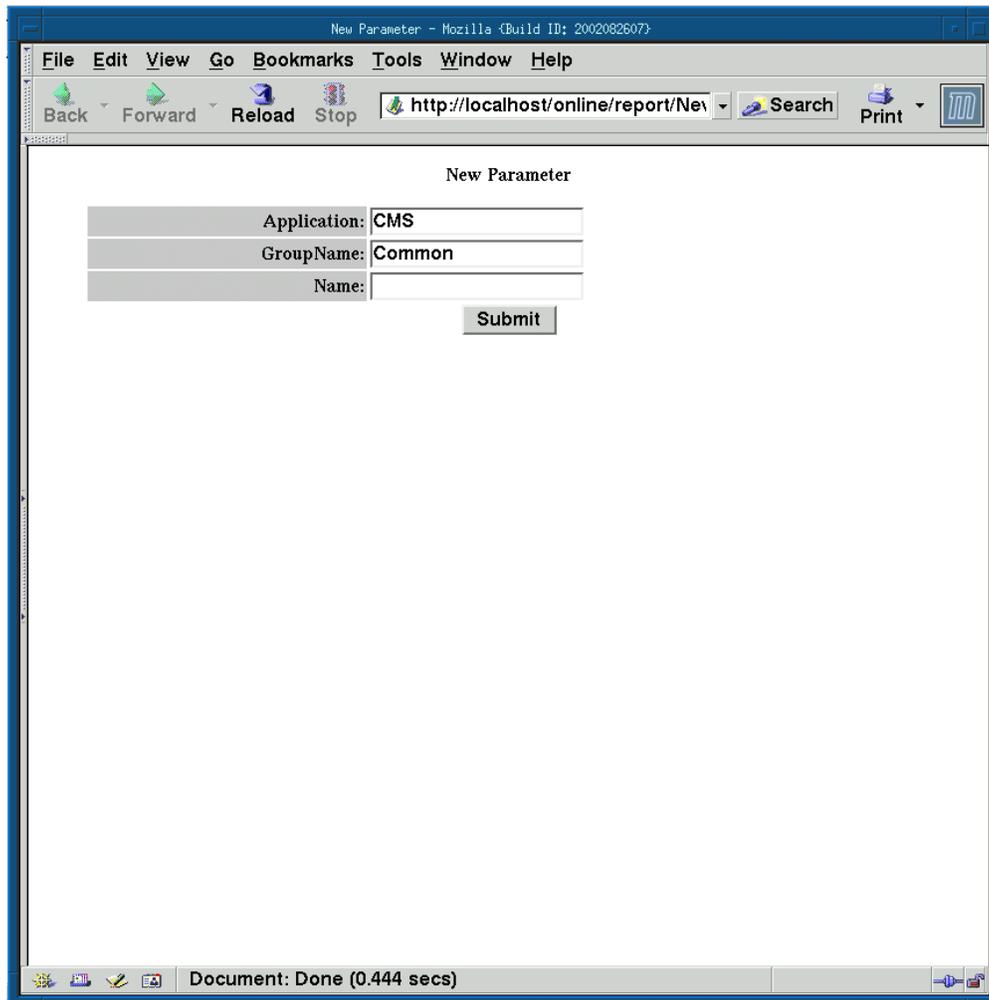


Clicking on the parameter group called Common brings up the parameters under that group's category.

The list of parameters will be shown for that application and group name. Next to the name of the parameter will be an input box for the parameter value. Current values will be pre-populated in the box. They can be changed by entering a new value and clicking the “Submit” button.

| Application | GroupName | Name | Purpose |
|-------------|-----------|----------|--|
| ALL | Logging | Logfile | Name and location of the log file for the parameter. |
| ALL | Logging | LogLevel | 0 – no logging, 1 – informational, 2 – detailed, 3 - debug |

Example documenting of parameters in an application....



After choosing Application and GroupName values, clicking New Parameter will have those values pre-populated.

To enter a new parameter into the list of parameters, click on the New Parameter link. This will display a screen allowing the user to enter a new parameter. The value of the parameter will need to be entered in the main parameter maintenance screen.

ParameterMaintenance.html

This is the complete source code to the main parameter maintenance screen.

```
<!--WSS  
  
DEF VAR cCurApplication      AS CHARACTER NO-UNDO.  
DEF VAR cCurGroup          AS CHARACTER NO-UNDO.  
DEF VAR cListApplication    AS CHARACTER NO-UNDO.  
DEF VAR cListGroup         AS CHARACTER NO-UNDO.  
  
{htmltools.i}
```

```

/*****
/* Function that will list all the possible applications */
*****/

FUNCTION ListApplications RETURNS CHARACTER:

    DEF VAR cList AS CHARACTER NO-UNDO.

    FOR EACH ParmS NO-LOCK:

        IF can-do(cList, ParmS.Application) = NO THEN
            ASSIGN cList = cList + ParmS.Application + ",".

    END. /* FOR EACH ParmS */

    RETURN SUBSTRING(cList, 1, LENGTH(cList) - 1).

END. /* FUNCTION ListApplications */

/*****
/* Function that will list all the possible groups within an application */
*****/

FUNCTION ListGroups RETURNS CHARACTER (INPUT cApplication AS CHARACTER):

    DEF VAR cList AS CHARACTER NO-UNDO.

    FOR EACH ParmS NO-LOCK
    WHERE ParmS.Application = cApplication:

        IF Can-DO(cList, ParmS.GroupName) = NO THEN
            ASSIGN cList = cList + ParmS.GroupName + ",".

    END. /* FOR EACH ParmS */

    RETURN SUBSTRING(cList, 1, LENGTH(cList) - 1).

END.

/*****
/* Determine the current application we are working on. */
*****/

ASSIGN cCurApplication = GET-VALUE("Application").
ASSIGN cCurGroup = GET-VALUE("Group").

/*****
/* Prep our listings of applications and groups based on application */
/* chosen. */
*****/

ASSIGN cListApplication = ListApplications().
IF cCurApplication <> "" THEN cListGroup = ListGroups(cCurApplication).

/*****
```

```

/* Determine if we are posting from the submit button. This is known */
/* from the switch value DisplayedParameters. If YES, then we have dis- */
/* played the parameters and can assume the user got here via the submit */
/* button. */
/*****/

IF GET-VALUE("DisplayedParameters") = "YES" THEN DO:

    /*****/
    /* Make sure we got here via the submit button and not via a change of */
    /* application or group name. If we are on the same group and ap- */
    /* application, make the changes to the record. */
    /*****/

    IF GET-VALUE("PreviousApplication") = cCurApplication
    AND GET-VALUE("PreviousGroup") = cCurGroup THEN DO:

        FOR EACH ParmS EXCLUSIVE-LOCK
        WHERE ParmS.Application = cCurApplication
          AND ParmS.GroupName = cCurGroup:

            ASSIGN ParmS.ParmValue = GET-VALUE("Value." + STRING(RowID(ParmS))).

        END. /* FOR EACH ParmS */

    END. /* IF GET-VALUE("PreviousApplication") = cCurApplication... */

END. /* IF GET-VALUE("DisplayedParameters") = "YES" */

-->
<html>
<head>
<title>Parameters Maintenance</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body bgcolor="#FFFFFF">
<p align="center"><b>Parameters Maintenance</b></p>
<form method="post" name="a">
  <table width="90%" border="0" align="center" cellspacing="0">
    <tr bgcolor="#CCCCCC">
      <td>
        <div align="center"><b>Application</b></div>
      </td>
      <td>
        <div align="center"><b>Parameter Group</b></div>
      </td>
    </tr>
    <tr bgcolor="#CCCCCC">
      <td>
        <div align="center">
          `HTMLSelect("a","Application",10,
            cListApplication,cListApplication,
            cCurApplication,'onChange="this.form.submit()"`
        </div>
      </td>
    </tr>
  </table>
</form>

```

```

</td>
<td>
  <div align="center">
    `HTMLSelect("a","Group",10,cListGroup,
      cListGroup,cCurGroup,'onChange="this.form.submit()"')`
  </div>
</td>
</tr>
</table>
<p align="center"><a
href="NewParameter.html?Application=`cCurApplication`&GroupName=`cCurGroup`"
target="_blank">New Parameter</a></p>
<table width="90%" border="0" align="center" cellspacing="0">
<tr bgcolor="#CCCCCC">
<td>
  <div align="center"><b>Parameter Name</b></div>
</td>
<td>
  <div align="center"><b>Parameter Value</b></div>
</td>
</tr>
<!--WSS
IF cCurGroup <> "" THEN DO:
-->
<input type="hidden" name="DisplayedParameters" value="YES">
<input type="hidden" name="PreviousApplication" value="`cCurApplication`">
<input type="hidden" name="PreviousGroup" value="`cCurGroup`">
<!--WSS
FOR EACH Parms NO-LOCK
WHERE Parms.Application = cCurApplication
AND Parms.GroupName = cCurGroup:

-->
<tr>
<td>
  <div align="center">
    `Parms.ParmName`
  </div>
</td>
<td>
  <div align="center">
    <input type="text" name="Value.`STRING(RowID(Parms))`"
value="`Parms.ParmValue`">
  </div>
</td>
</tr>
<!--WSS
END. /* FOR EACH Parms */
END. /* IF cCurGroup <> "" */
-->
</table>
<p align="center">
<input type="submit" name="Submit" value="Submit">
</p>
</form>

```

```
<p>&nbsp;</p>
</body>
</html>
```

NewParameter.html

This is the complete source code to the new parameter maintenance screen.

```
<!--WSS
DEF VAR cApplication AS CHARACTER NO-UNDO.
DEF VAR cGroupName AS CHARACTER NO-UNDO.
DEF VAR cName AS CHARACTER NO-UNDO.
DEF VAR cValue AS CHARACTER NO-UNDO.
DEF VAR cMessage AS CHARACTER NO-UNDO.

ASSIGN
cApplication = GET-VALUE("Application")
cGroupName = GET-VALUE("GroupName")
cName = GET-VALUE("Name")
cValue = GET-VALUE("Value").

/* Try making the parms record */

IF cApplication <> ""
AND cGroupName <> ""
AND cName <> "" THEN DO:

    IF NOT CAN-FIND (Parms WHERE Parms.Application = cApplication
                    AND Parms.GroupName = cGroupName
                    AND Parms.ParmName = cName) THEN DO:

        CREATE Parms.

        ASSIGN
        Parms.Application = cApplication
        Parms.GroupName = cGroupName
        Parms.ParmName = cName
        Parms.ParmValue = cValue.

        ASSIGN cMessage = "Created Parameter".

    END.

END.

-->
<html>
<head>
<title>New Parameter</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body bgcolor="#FFFFFF">
<div align="center">
<p><b>New Parameter</b></p>
<p>`cMessage`</p>
<form method="post" action="">
<table width="90%" border="0">
<tr>
<td bgcolor="#CCCCCC">
<div align="right"><b>Application:</b></div>
</td>
<td>

```

```

        <input type="text" name="Application" value="`cApplication`">
    </td>
</tr>
<tr>
    <td bgcolor="#CCCCCC">
        <div align="right"><b>GroupName:</b></div>
    </td>
    <td>
        <input type="text" name="GroupName" value="`cGroupName`">
    </td>
</tr>
<tr>
    <td bgcolor="#CCCCCC">
        <div align="right"><b>Name:</b></div>
    </td>
    <td>
        <input type="text" name="Name" value="`cName`">
    </td>
</tr>
<!--WSS /* Should use parameter maintenance to do this
    <tr>
        <td bgcolor="#CCCCCC">
            <div align="right"><b>Value:</b></div>
        </td>
        <td>
            <input type="text" name="Value" value="`cValue`">
        </td>
    </tr>
*/ -->
    <tr>
        <td colspan="2">
            <div align="center">
                <input type="submit" name="Submit" value="Submit">
            </div>
        </td>
    </tr>
</table>
</form>
<p>&nbsp;</p>
</div>
</body>
</html>

```

HTMLSelect.i

This is ancillary code needed to make the ParameterMaintenance.html program function. It makes a drop down select box.

```

/*E4GL-I*/ {&OUT} '<meta NAME="wsoptions" CONTENT="include">~n'.

/*Tag=<!--WSS*/
FUNCTION HTMLSelect RETURNS CHARACTER
(INPUT cFormName AS CHARACTER,
 INPUT cSelectName AS CHARACTER,
 INPUT cSize AS INTEGER,
 INPUT cValueList AS CHARACTER,
 INPUT cDispList AS CHARACTER,
 INPUT cSelected AS CHARACTER,
 INPUT cJavaScript AS CHARACTER):

    DEF VAR i AS INTEGER NO-UNDO.
    /*Tag=-->*/
    {&OUT} '<select name=" ' /*Tag=`*/ cSelectName /*Tag=`*/ ' " size=" '
    /*Tag=`*/ cSize /*Tag=`*/ ' " ' /*Tag=`*/ cJavaScript /*Tag=`*/ ' '>~n'.

```

```

/*Tag=<!--WSS*/ DO i = 1 TO NUM-ENTRIES(cValueList): /*Tag=-->*/
{&OUT} ' <option value="" /*Tag=`*/ ENTRY(i,cValueList) /*Tag=`*/ ' ' '
/*Tag=`*/ IF ENTRY(i,cValueList) = cSelected THEN "SELECTED" ELSE ""
/*Tag=`*/ '>' /*Tag=`*/ ENTRY(i,cDispList) /*Tag=`*/ '</option>~n'.
/*Tag=<!--WSS*/ END. /*Tag=-->*/
{&OUT} '</select>~n'.
/*Tag=<!--WSS*/
RETURN "".
END.
/*Tag=-->*/

/***** END OF HTML *****/
/*
** File: /appl/report/src/htmltools.i
** Generated on: 2002-12-18 13:59:37
** By: WebSpeed Embedded SpeedScript Preprocessor
** Version: 2
** Source file: /appl/report/src/htmltools.html
** Options: include,wsoptions-found
**
** WARNING: DO NOT EDIT THIS FILE. Make changes to the original
** HTML file and regenerate this file from it.
**
*/

/* end */

```

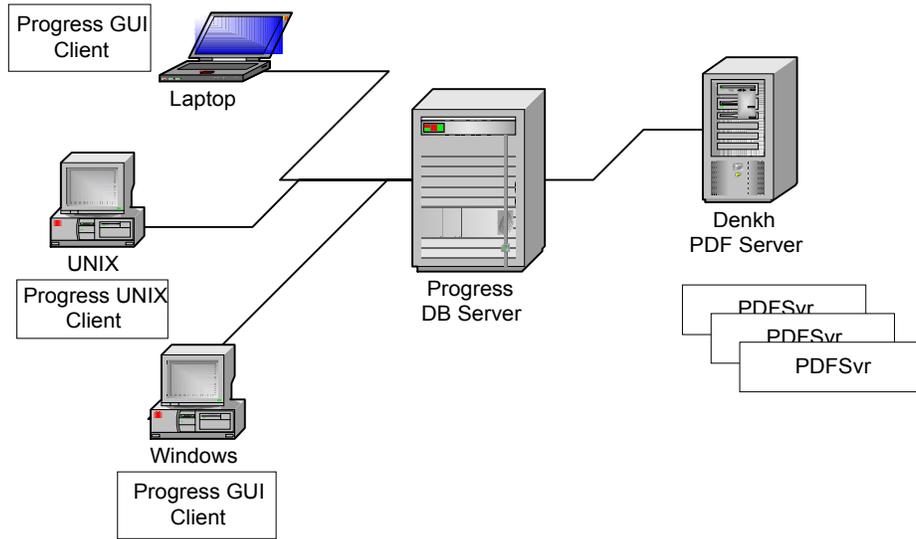
Coding Article: Poor Man's AppServer

Written by Scott Auge sauge@amduus.com

One can use the database to send messages to batch programs running out on other machines. Some of these machines might have software available that are not available locally. One might also have more modules of code on another machine – a place that can handle activities from any type of interface available from Progress.

The example in this article is The Denkh PDF Server software which allows a highly flexible means of generating PDF files for the users of Progress clients. Progress clients can be GUI/WWW/CHUI, while the PDF server process (a progress batch program waiting for a message to do something) runs on a Linux server equipped with the ghostscript and groff packages.

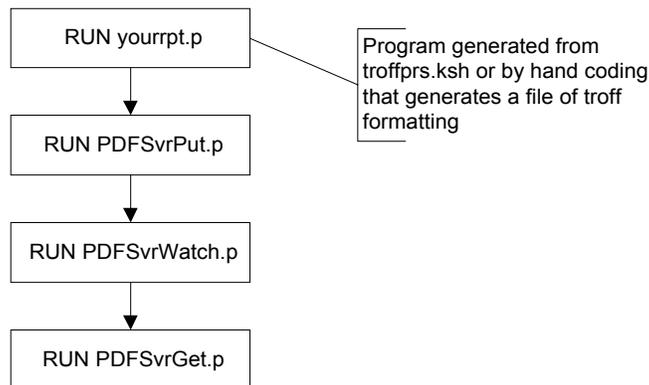
Architecture



The application that needs PDF reports run on various clients available from Progress Software. These clients use an API of routines to request reports of the PDFSvr process which is running on a server. This set up is required because the PDFSvr process needs Linux to properly render PDF files for the client application.

Using the Client API to request a PDF

Using the API is very simple. It is composed of three procedure files: one routine places a request onto the server, another watches the status of the request, and the final pulls the requested PDF (when available.)



Using the example program from the supplied source found in the Denkh/src/examples directory, we run a program created from the t4.troff program. (See above for converting *.troff routines into .p routines.)

```
RUN examples/t4.p (INPUT
"/tmp/t4.data").
```

When your operating system can handle generating PDFs directly with the Denkh package (Linux required), you can continue on as explained in the above section. When not, you need to call a Denkh PDF Server to render the report into PDF format for you.

Amduus Information Works, Inc. also provides documentation services! 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!

First we make a call to the Put routine to put a request into the Denkh PDF Server:

```
RUN PDFClT/PDFSvrPut.p (INPUT "/tmp/t4.data", OUTPUT cReportID).
```

We send the filename of the output of the report program to the API, and receive a ReportID number. We need this ReportID number to reference the report in the other APIs. This is a very important number because one will lose the report in the wilds of the system – think of it as the request’s name.

It may be that the Denkh PDF Server is busy processing other reports (or simply is not running) – hence an API is available to find out what is happening with the report. It returns the report’s status. When the status becomes “DONE” then the Denkh PDF Server has processed the report and the programmer can call the PDFSvrGet.p routine to get the PDF report.

```
RUN PDFClT/PDFSvrWatch.p (INPUT cReportID, OUTPUT cStatus).
```

The status can be of the following:

| | |
|------------|---|
| SUBMITTED | The report has been submitted and awaiting processing |
| PROCESSING | The PDF Server is processing the report right now |
| LOCKED | The record about the report is LOCKED. |
| NO_RECORD | The report ID is bad, or the record has been lost |
| DONE | The PDF file is ready! |

Once the report is ready for the client application, the PDFSvrGet.p routine is used to pull the report out of the system onto the file system of the client application:

```
RUN PDFClT/PDFSvrGet.p
(INPUT "/tmp/myname.pdf",
```

```
INPUT cReportID,  
OUTPUT cStatus).
```

If the report is not yet ready, the routine will return the status of the report.

The arguments are the name of the report you wish the PDF file to have, the ReportID naming the report, and an output telling you the status of the report. Any status other than DONE will not result in a PDF file available for the user.

There after, refer to the report as a file on your client's machine, as in the example it would be /tmp/myname.pdf.

Administration of the Denkh PDFSvr process

Starting the Denkh PDF Server process:

```
Denkh/script/PDFSvr.ksh >> /tmp/DenkhLog
```

Be sure to tailor the script for your system's configuration.

There can be multiple processes running to help load distribution on heavily used systems, however, the rendering is generally very fast and many places need only one process running.

Below is source to the above mentioned functionality. The full source is available in the Denkh package found at www.amduus.com.

PDFSvr.ksh

```
DLC=/usr/dlc  
export DLC;  
  
PROPATH=/appl/Denkh/src  
export PROPATH;  
  
PFFILE=/db/amduus/parm/amduus.pf  
export PFFILE;  
  
$DLC/bin/_progres -b -p PDFSvr/PDFSvr.p -pf $PFFILE
```

PDFSvr.p

```
DEF VAR RCSVersion AS CHARACTER INIT "$Header:"
```

```
/home/appl/DenkH/src/PDFsvr/RCS/PDFsvr.p,v 1.1 2002/07/12 03:41:08 sauge Exp
sauge $" NO-UNDO.

/* The PDF Server is a process that runs in the background and performs */
/* a continous loop looking for PDFReport records with a ReportStatus */
/* of SUBMITTED. Processing the report changes the status to PROCESSING. */
/* Once completed the status is changed to DONE and the client can use */
/* PDFClnt/PDFsvrGet.p to pull the PDF file from the system. */

/* This program needs to run on a Linux machine with Ghostscript and */
/* groff on it. Such a server can be built for you by Amduus Information */
/* Works, Inc. */

/* Multiple instances of this program can run at the same time for load */
/* bearing tweeking. */

/* Just kill the process to shutdown this processor. */
/* Newer versions will include a cleaner shutdown method. */

DEF VAR cPDFReportID AS CHARACTER NO-UNDO.
DEF VAR cOSFileID AS CHARACTER NO-UNDO.
DEF VAR cError AS CHARACTER NO-UNDO.

OUTPUT TO /dev/tty UNBUFFERED.

REPEAT:

    FIND FIRST PDFReport EXCLUSIVE-LOCK
    WHERE PDFReport.ReportStatus = "SUBMITTED"
    NO-ERROR.

    /* Determine if the report record is available. If not, then wait a bit */
    /* and try again. */

    IF NOT AVAILABLE PDFReport THEN DO:

        PAUSE 2 NO-MESSAGE.
        MESSAGE "Pausing".
        NEXT.

    END. /* IF NOT AVAILABLE PDFReport */

    /* We got one! Change the status so PDFsvrWatch.p can pick up what is */
    /* happening over here. Release it so that the client can pick up on */
    /* this at their side. */

    MESSAGE "Processing Report Request: " PDFReport.PDFReportID.

    ASSIGN PDFReport.ReportStatus = "PROCESSING".
    ASSIGN cPDFReportID = PDFReport.PDFReportID.

    RELEASE PDFReport.

    /* Refind the PDFFile with a NO-LOCK status and process it */

    FIND PDFReport NO-LOCK
    WHERE PDFReport.PDFReportID = cPDFReportID
    NO-ERROR.

    /* Pull out the groff stuff. File is named with the PDFReportID */
    /* as since that is unique it should be unique on the file sytem */

    MESSAGE "Pulling out troff source: /tmp/" PDFReport.PDFReportID.
```

```

RUN file/RetrieveFile.p
(INPUT PDFReport.INOSFileID,
 INPUT PDFReport.PDFReportID,
 INPUT "/tmp/",
 OUTPUT cError).

/* Run it though the routines that will ploput a PDF file */

MESSAGE "Creating PDF: " "/tmp/" + PDFReport.PDFReportID + ".pdf".

RUN groff2pdf.p
(INPUT "/tmp/" + PDFReport.PDFReportID,
 INPUT "/tmp/" + PDFReport.PDFReportID + ".pdf").

/* Read the PDF File into the DB */

MESSAGE "Storing PDF: /tmp/" PDFReport.PDFReportID + ".pdf".

RUN file/StoreFile.p
(INPUT PDFReport.PDFReportID + ".pdf",
 INPUT "/tmp/",
 INPUT "PDFReport",
 OUTPUT cOSFileID,
 OUTPUT cError).

/* Update the PDFReport record to note it is processed and ready for the */
/* client.                                                                    */

FIND PDFReport EXCLUSIVE-LOCK
WHERE PDFReport.PDFReportID = cPDFReportID
NO-ERROR.

ASSIGN
PDFReport.ReportStatus = "DONE"
PDFReport.OutOSFileID = cOSFileID.

MESSAGE "Report " cPDFReportID " marked DONE".

END. /* REPEAT: */

```

PDFSvrPut.p

```

DEF VAR RCSVersion AS CHARACTER INIT "$Header:
/home/appl/Denkh/src/PDFClt/RCS/PDFSvrPut.p,v 1.1 2002/07/11 20:15:41 sauge Exp
$" NO-UNDO.

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

DEF VAR cOSFileID AS CHARACTER NO-UNDO.
DEF VAR cError    AS CHARACTER NO-UNDO.

{MakeID3.i}
{FileName.i}
{FilePath.i}

```

```
/* Load the file into the Progress DB and aquire an OSFileID */  
  
RUN file/StoreFile.p  
(INPUT FileName(cFilename),  
 INPUT FilePath(cFileName),  
 INPUT "PDFReport",  
 OUTPUT cOSFileID,  
 OUTPUT cError).  
  
/* Create a PDFReport record to store it in and let the PDF Server know it */  
/* needs to be processed. */  
  
CREATE PDFReport.  
  
ASSIGN PDFReport.PDFReportID = MakeID3(10).  
  
ASSIGN  
PDFReport.InOSFileID = cOSFileID  
PDFReport.ReportStatus = "SUBMITTED"  
PDFReport.CreateDate = TODAY  
PDFReport.CreateTime = TIME.  
  
ASSIGN cPDFReportID = PDFReport.PDFReportID.  
  
RELEASE PDFReport.
```

PDFSvrWatch.p

```
DEF VAR RCSVersion AS CHARACTER INIT "$Header:  
/home/appl/DenkH/src/PDFClt/RCS/PDFSvrWatch.p,v 1.1 2002/07/11 20:15:41 sauge  
Exp $" NO-UNDO.  
  
DEF INPUT PARAMETER cPDFReportID AS CHARACTER NO-UNDO.  
DEF OUTPUT PARAMETER cReportStatus AS CHARACTER NO-UNDO.  
  
FIND PDFReport NO-LOCK  
WHERE PDFReport.PDFReportID = cPDFReportID  
NO-ERROR NO-WAIT.  
  
IF LOCKED PDFReport THEN DO:  
  
    ASSIGN cReportStatus = "LOCKED".  
    RETURN.  
  
END.  
  
IF NOT AVAILABLE PDFReport THEN DO:  
  
    ASSIGN cReportStatus = "NO_RECORD".  
    RETURN.  
  
END.  
  
ASSIGN cReportStatus = PDFReport.ReportStatus.
```

```
RELEASE PDFReport.
```

PDFSvrGet.p

```
DEF VAR RCSVersion AS CHARACTER INIT "$Header:
/home/appl/Denkh/src/PDFClt/RCS/PDFSvrGet.p,v 1.1 2002/07/11 20:15:41 sauge Exp
$" NO-UNDO.

DEF INPUT PARAMETER cFileName          AS CHARACTER NO-UNDO.
DEF INPUT PARAMETER cPDFReportID       AS CHARACTER NO-UNDO.
DEF OUTPUT PARAMETER cReportStatus     AS CHARACTER NO-UNDO.

{FileName.i}
{FilePath.i}

/* Determine if we can find the PDFReport record mentioned. */

IF NOT CAN-FIND(PDFReport WHERE PDFReport.PDFReportID = cPDFReportID) THEN DO:

    ASSIGN cReportStatus = "NO_RECORD".
    RETURN.

END.

/* Lets see if we can nab it */

FIND PDFReport EXCLUSIVE-LOCK
WHERE PDFReport.PDFReportID = cPDFReportID
NO-ERROR NO-WAIT.

IF LOCKED PDFReport THEN DO:

    ASSIGN cReportStatus = "LOCKED".
    RETURN.

END.

/* Lets see if it is done by the PDF Server */

IF PDFReport.ReportStatus <> "DONE" THEN DO:

    ASSIGN cReportStatus = PDFReport.ReportStatus.
    RETURN.

END.

/* Looks like it is ours for the taking! Process it. */

RUN file/RetrieveFile.p
(INPUT PDFReport.OutOSFileID,
 INPUT FileName(cFileName),
 INPUT FilePath (cFileName),
 OUTPUT cReportStatus).

ASSIGN cReportStatus = PDFReport.ReportStatus.

RELEASE PDFReport.
```

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 www.peg.com.

Products Available From Amduus:

Amduus Information Works, Inc. is a Progress reseller and ASPen partner. We primarily develop UNIX/Linux based applications for the web. We also perform integration of 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.

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.

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!