
NetPhantom Ticket Proof of Concept

NetPhantom Ticket Proof of Concept

Version 5.0

Document Revision 4

20 March 2010



The screenshot shows a web application interface with a blue header and a dark blue sidebar. The header contains the 'NetPhantom' logo in white. The sidebar lists navigation options under 'Application routines' and 'Parameter routines'. The main content area features a photograph of a man in a suit talking on a mobile phone, with the text 'Ticket Proof of Concept' and a call to action below it. A footer contains copyright information.

NetPhantom

Application routines
Main page
Java Application
HTML Application
Documentation

Parameter routines
CustID=123
CustID=234
CustID=NEW

HOME

Ticket Proof of Concept

Click the various links in the menu to start applications.

Phantom® and NetPhantom® are registered trademarks of Codectica Systems S.A.R.L.
© Copyright Codectica Systems S.A.R.L., 2008.

Nexum Technologies S.A.R.L

NetPhantom®

Version 5.0

© Copyright Nexum Technologies SARL, 2010. All rights reserved.

Information in this document is subject to change without notice. Companies, names, and data used in examples are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Nexum Technologies.

Nexum Technologies may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. The furnishing of this document does not give you the license to these patents, trademarks, copyrights, or other intellectual property rights except as expressly provided in any written license agreement from Nexum Technologies.

Phantom® and NetPhantom® are registered trademarks of Nexum Technologies SARL. Java is a trademark of Sun Microsystems Incorporated. ActiveX, Microsoft, Windows are either registered trademarks or trademarks of Microsoft in the United States and/or other countries. IBM is a registered trademark of International Business Machines Corporation.

Nexum Technologies SARL

**World Trade Center, Nice – Sophia Antipolis
1300, route des Crêtes – BP 255
FR-06905 Sophia Antipolis Cedex
FRANCE**

Telephone: +33 954 15 30 20
Fax: +33 959 15 30 20
Web: <http://netphantom.com>
E-mail: info@nexum-tech.com

Support and Information

Germany: +49 69 401 590 987 (info.de@nexum-tech.com)
Sweden: +46 8 525 00 123 (info.se@nexum-tech.com)
World-wide: +33 954 15 30 20
E-mail: support@netphantom.com

Contents

1 NetPhantom Ticket Proof of Concept	2
1.1 Installation.....	2
1.2 Configuration of the NetPhantom Server.....	2
Base Configuration.....	2
Configuration of Session Pooling.....	3
Configuration of the Web Server	4
2 The Recorded AS/400 Session	5
3 The NetPhantom Application	7
3.1 Login Panel.....	7
Listing of LOGIN.....	7
Listing of SETPARAMS	8
3.2 Main Panel.....	9
Listing of PARAMS	9
3.3 User Panel	9
3.4 Change Password Panel.....	10
Listing of CHGPASS – CHGPASS2.....	10
4 The HTML Application	11
4.1 Tour of the HTML Application.....	11
Testing User parameters.....	13
5 Session Pooling Script	15

1 NetPhantom Ticket Proof of Concept

NetPhantom Ticket Proof of Concept is a sample application (AS/400 based) running under NetPhantom. The purpose is to have an external CRM Microsoft Internet Information Server system to become seamlessly integrated with an AS/400 application using NetPhantom HTML Integration. This is done by having HTML frames on a page and running the NetPhantom application in the main frame.

The application can be run “live” to an AS/400, but also against the provided “EmulatorEmulator” (EE) application (`ticket.eem`).

The CRM system starts the NetPhantom HTML application by having a hyperlink concatenated by `?params` where `params` can be several parameters separated by ‘+’ as `parameterName=value`.

Example:

```
http://netphantom_server/ticket/run.jsp?CustNo=123+Routine=2
```

In order to run the application without the CRM system, a framework of frames and a menu bar has been created. The screen capture on the front page of this document is what is seen when the framework is started.

1.1 Installation

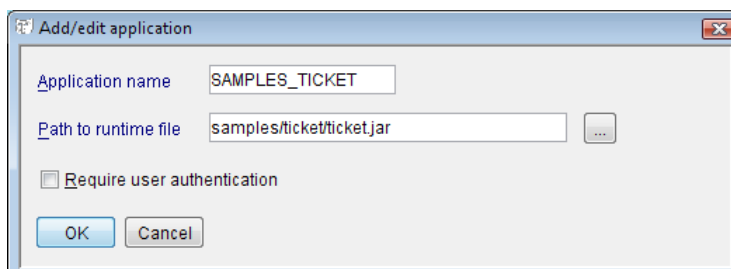
This sample is by default included in the NetPhantom installation.

1.2 Configuration of the NetPhantom Server

The following section describes the changes needed to configure the NetPhantom Server for the application using the Server Administration program.

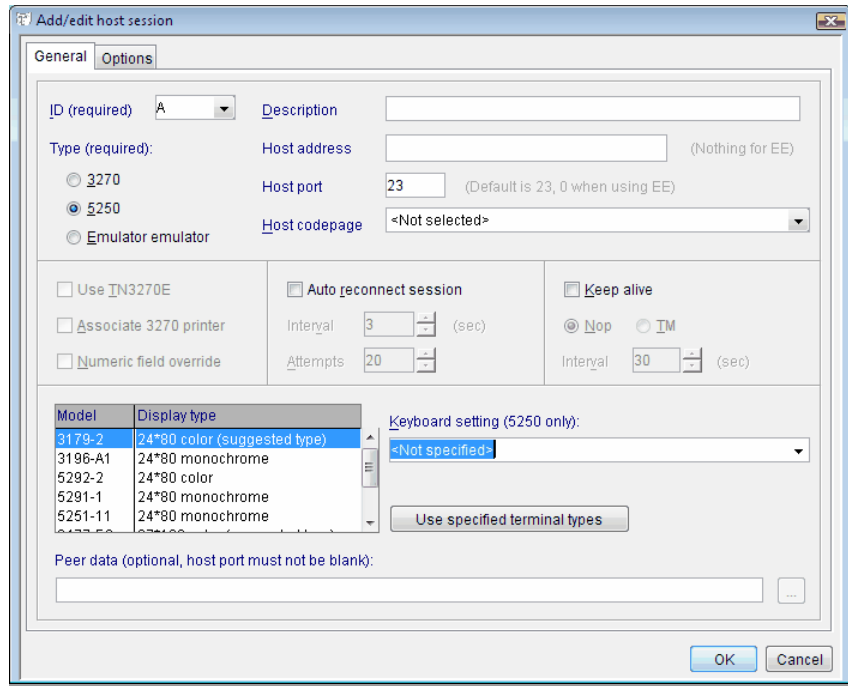
Base Configuration

Select menu item **Server – Configure – Base**. Add the Ticket application using the Application name `TICKET` and the runtime file name `ticket/TICKET.jar` at the **Applications** tab.

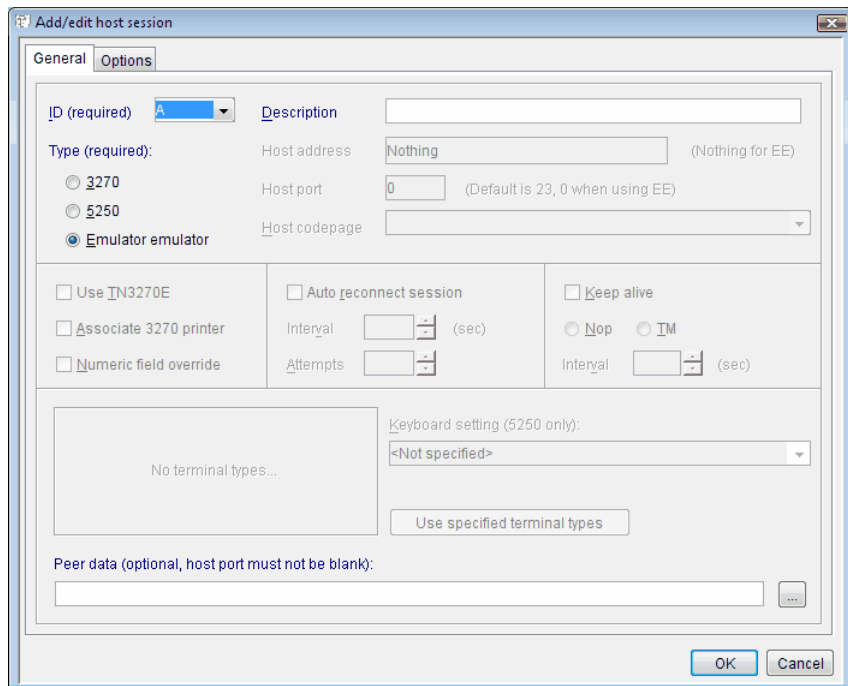


At the host section, make sure to select the **Host codepage 278 Finland/Sweden**.

Create a “live” AS/400 (5250) host session as:



or an EmulatorEmulator application (ticket.eem) as:



When these steps are done, press **OK** to close the **Configure base** dialog.

Note: You will then have to *restart the server* using the **Hard JVM restart** option.

Configuration of Session Pooling

Once the server has restarted, select menu item **Server – Configure – Session pooling**.

Create an entry as the dialog box below shows:

Note: do not forget to press the **Add** button. When this is done, the session pool is created with 10 pre-started sessions.

Configuration of the Web Server

Next the NetPhantom Web Server must be configured. Select menu item **Server – Configure – Web server**. Create the HTML Web Application using the **Web Applications** tab. Enter the information below, and then press **Add**.

This will create the Web Application **TICKET**. Make sure that you check the option **Use previous session**. If you don't do this, a new session is created when a new HTML application with parameters is started and the user must always logon to the AS/400.

The **Session time-out** should be set to a value about 5 minutes shorter than the *User stand-by time* in the AS/400. If the **Session time-out** is much shorter, e.g. a 5 minutes time-out, no harm is done, but after user inactivity of 5 minutes, he/she must logon to the AS/400 again.

Next a resource must be created for this application. This is done under the **Resources** tab.

Enter the following information and press **Add**.

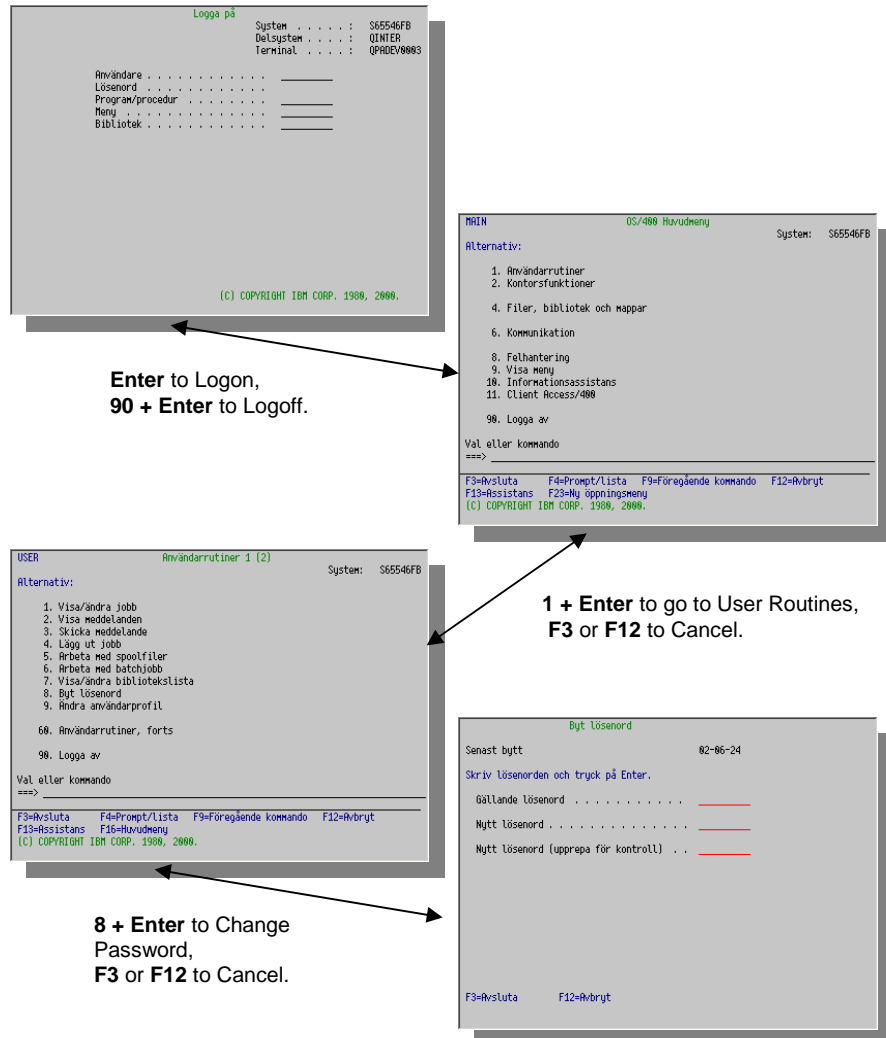
Now close the dialog using **OK**, and **restart** the server (**Hard JVM Restart**).

This completes the configuration of the Server.

2 The Recorded AS/400 Session

In order to test the application without an AS/400, use the `ticket.eem` EmulatorEmulator application.

The following picture describes the flow of the application.



3 The NetPhantom Application

The NetPhantom application is designed using NetPhantom Editor and consists of 4 panels and a few REXX programs.

The panels for this application are not very elegant, but this does not really matter as only HTML pages will be displayed to the end user.

3.1 Login Panel

The Login panel is connected to the Object LOGIN as well as the **Login** button. User parameters can be set by entering them in the multiple line entry field and pressing the **Set parameters** button (using the SETPARAMS object, see below).

When the panel is created, the REXX code gets and parses the parameters passed into the program and checks if a valid User ID and Password (UI/PW) exist. If it does, the UI/PW is entered and the Enter key is pressed. The LOGIN REXX macro performs these functions.

Listing of LOGIN

```

/* LOGIN.CMD */

Parse Arg argId, argMsg, argStr

if argMsg='CRT' then
do
  /* Skip if CRT for the ENTER button */
  if argId<>'LOGIN' then return 0

  /* Get params */
  rc=getParams()

  /* Restore previously stored UI/PW */
  ui=GlobVarGet('LOGIN_UI')
  pw=GlobVarGet('LOGIN_PW')
  rc=PanSetCtlData('UI',ui)
  rc=PanSetCtlData('PW',pw)
  if length(ui)>0 & length(pw)>0 then
    rc=HostSend('@E')
  return 0
end

/* Only when CMD is received */
if argMsg<>'CMD' | argId<>'ENTER' then return 0

/* When "pressing Enter", save UI/PW */
ui=PanGetCtlData('UI')
pw=PanGetCtlData('PW')
rc=GlobVarSet('LOGIN_UI',ui)
rc=GlobVarSet('LOGIN_PW',pw)
rc=HostSend('@E')

Return 0

```

```
/* Extracts parameters in HTTP request line as
http://host/page.jsp?params */
/* delimited by '+' signs */

getParams: procedure

params=GlobVarGet('NETPHANTOM_HTTP_URIPARAMS')
/*say 'HTTP_URIPARAMS="'params'"'*/
if length(params)=0 then
do
say 'No start params...'
return 0
end

/* Extract the params delimited by '+' */
ii=1
do while length(params)>0
parse var params var '+' rest
params=rest
rc=GlobVarSet('PARAM_'ii,var)
/*say 'PARAM_'ii' = 'var'*/
end

rc=GlobVarDelete('NETPHANTOM_HTTP_URIPARAMS')

return 0
```

Listing of SETPARAMS

```
/* SETPARAM.CMD */

parse arg ii,aa,ss

p=PanGetCtlData('PARAMS')

crlf='0D'X || '0A'X
param=''
ii=1

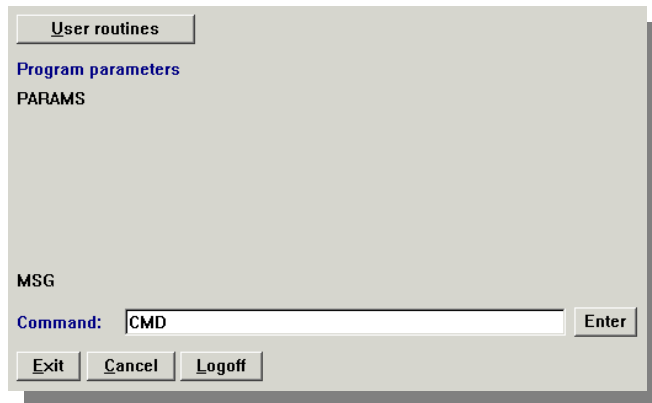
do while length(p)>0
d1=pos('0D'X,p)
d2=pos('0A'X,p)
if d1=0 & d2=0 then
do
if length(p)>0 then
do
param=param || 'PARAM_'ii || '=' || p || crlf
rc=GlobVarSet('PARAM_'ii,p)
ii=ii+1
end
leave
end
d=d1
if d=0 | (d2>0 & d2<d) then d=d2
p1=left(p,d-1)
p=substr(p,d+1)
if p1='' then iterate
param=param || 'PARAM_'ii || '=' || p1 || crlf
rc=GlobVarSet('PARAM_'ii,p1)
ii=ii+1
end

rc=GlobVarSet('PARAM_'ii,'')

rc=message(2,2,'Parameters set:'crlf || param)

return 0
```

3.2 Main Panel



The Main panel is used to display parameters passed to the program. This is done using the PARAMS REXX macro using the CRT object message. The object is connected to the output text PARAMS above.

Listing of PARAMS

```
/* PARAMS.CMD */

crlf='0D'X || '0A'X

txt='UserID = ' || GlobVarGet('LOGIN_UI') || crlf
txt=txt || 'Password = ' || GlobVarGet('LOGIN_PW') || crlf

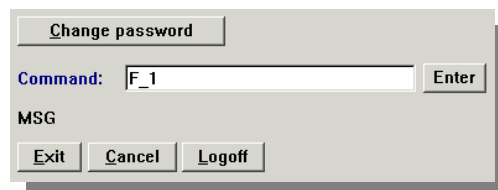
ii=1
do forever
  rc=GlobVarGet('PARAM_'ii)
  if rc='' then leave
  txt=txt || 'PARAM'ii || '=' || rc || crlf
  ii=ii+1
end

rc=PanSetCtlData('PARAMS',txt)

return 0
```

3.3 User Panel

This panel is only used as a transportation panel between the MAIN and CHGPASS panels.



3.4 Change Password Panel

The panel ID is CHGPASS and is used to change the password of the current user. While this is performed in the AS/400, the application stores the new password if a future session needs to logon the user (by automatically entering the UI/PW information).

The screenshot shows a panel titled "Enter passwords and press Enter." with the following fields and controls:

- Last changed:** 02-06-24
- Current password:** PW
- New password:** NPW1
- New password [repeat for check]:** NPW2
- MSG:** (empty)
- Buttons:** Change password, Exit, Cancel

Once the **Change password** button is pressed, the connected REXX macro CHGPASS is started. It saves the password into a temporary global variable and then starts the REXX application CHGPASS2 (the same REXX code as for CHGPASS). It checks for the host session to become unlocked. Once unlocked, the new password is accepted by the AS/400 if the host screen is no longer CHGPASS (but USER and USER displays a message that the password has been changed successfully). If the screen remains at CHGPASS, the user has entered the information erroneously.

Listing of CHGPASS – CHGPASS2

```

/* CHGPASS.CMD */

Parse Arg argId, argMsg, argStr

if argId='CHGPASS' then
do
  pw=PanGetCtlData('NPW1')
  rc=GlobVarSet('LOGIN_NPW',pw)
  rc=PanHostInhibit(1)
  rc=PanLock(1)
  rc=HostSend('@E')
  rc=CallObject('CHGPASS2','THREAD','CMD','')
  return 0
end

/* Async processing to see if screen changes from CHGPASS */
rc=HostWait(3)
if rc=0 & HostGetScreen()<>'CHGPASS' then
do
  pw=GlobVarGet('LOGIN_NPW')
  rc=GlobVarSet('LOGIN_PW',pw)
end
rc=PanHostInhibit(0)
rc=PanLock(0)

return 0

```

4 The HTML Application

Start the browser and enter the URL of the NetPhantom Server.

Example (running the server on the same machine as the browser):

```
http://localhost/ticket/index.html
```

This will display the HTML page using frames that is displayed on the front page of this document.

The HTML Application is started by clicking on one of the links **HTML Application** or the **Parameter routines** (CustID=123, CustID=234, CustID=NEW) from the menu bar located on the left hand side.

4.1 Tour of the HTML Application

Let's assume we click on the link **CustID=123**. The following page is then displayed:

Login

Enter User ID and Password and press **Login**.

System S65546FB
Subsystem QINTER
Terminal QPADEV0003

User ID
Password

Login

(C) COPYRIGHT IBM CORP. 1980, 2000.

User parameters

Enter User parameters as `VARIABLE=Text` on separate lines, then press **Set parameters**. This will set the parameters in the application as if they had been entered in the HTTP request line.

Set parameters

Enter **User ID** and **Password** correctly (for the recorded AS/400 session, you can enter anything). Then click on the **Login** button.

The Main page is then displayed:

Main Menu

Select **User Routines** below to change the password for the current user.

User routines

The following program parameters are currently specified for this session:

UserID = TESTUSER
Password = verysecret
PARAM1=CustID=123

Command: **Enter**

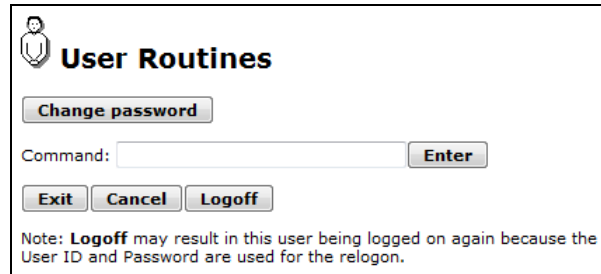
Exit **Cancel** **Logoff**

Note: **Logoff** may result in this user being logged on again because the User ID and Password are used for the relogin.

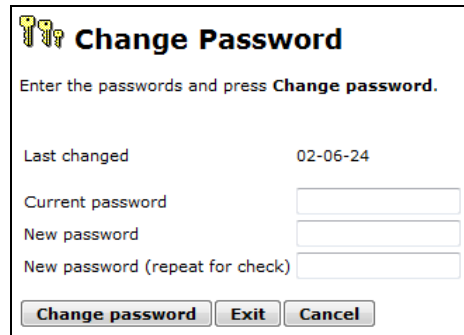
The Main panel displays the parameters passed into the session (stored in global variables PARAM_nn; in this case PARAM_1 is CustID=123, and PARAM_2 is an empty string indicating no more parameters are available).

The User ID and Password are store in the global variables named LOGIN_UI and LOGIN_PW respectively.

Click the **User routines** button and the following page is displayed:

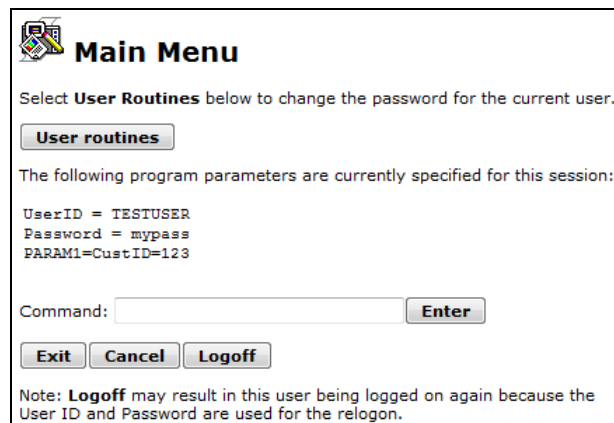


This is a page used for transportation between changing the password page and the main page. Click on the **Change password** button, and the following page is displayed:



Enter “verysecret” as the Current password and “mypass” as the new password, then navigate back to the Main page by pressing **Change password** followed by **Exit** button on the User routine page.

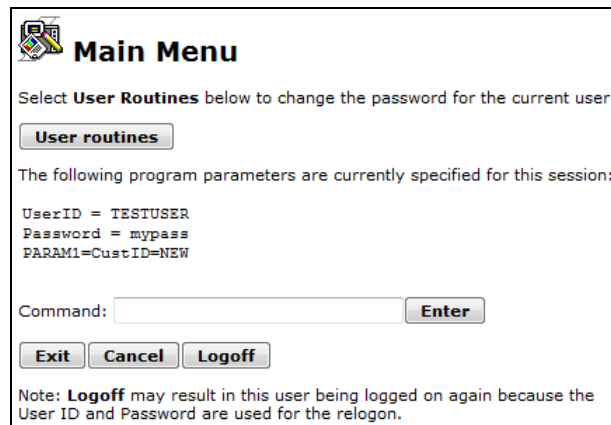
The Main page will then display the following parameter information (the new password “mypass”):



Now use the menu bar on the left hand side and click on the menu item **CustNo=NEW**.

The Login page will not be displayed (if you clicked the menu item before the session had timed out) because the application remembers the User ID and Password.

The Main page will then be displayed as follows (PARAM1=CustNo=NEW):



Main Menu

Select **User Routines** below to change the password for the current user.

User routines

The following program parameters are currently specified for this session:

```
UserID = TESTUSER
Password = mypass
PARAM1=CustID=NEW
```

Command: **Enter**

Exit **Cancel** **Logoff**

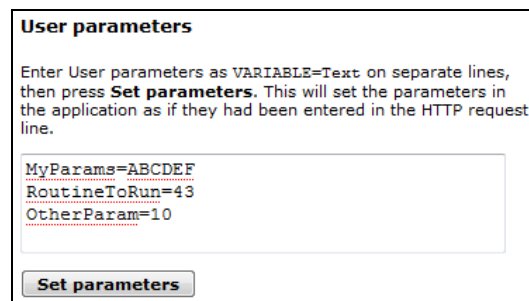
Note: **Logoff** may result in this user being logged on again because the User ID and Password are used for the relogin.

Testing User parameters

To test different user parameters for the application, start a new browser window and click on the link **HTML Application** in the menu bar on the left hand side.

This will display the Login panel.

Enter the following data:



User parameters

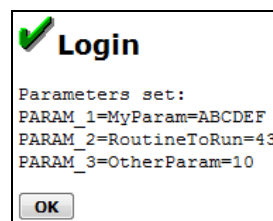
Enter User parameters as VARIABLE=Text on separate lines, then press **Set parameters**. This will set the parameters in the application as if they had been entered in the HTTP request line.

```
MyParams=ABCDEF
RoutineToRun=43
OtherParam=10
```

Set parameters

When you press the button **Set parameters**, a message box is displayed for the application (you can also test it by running the Java Client of this application from the menu item **Java Application** from the menu bar).

The following page is displayed for the message box:



Login

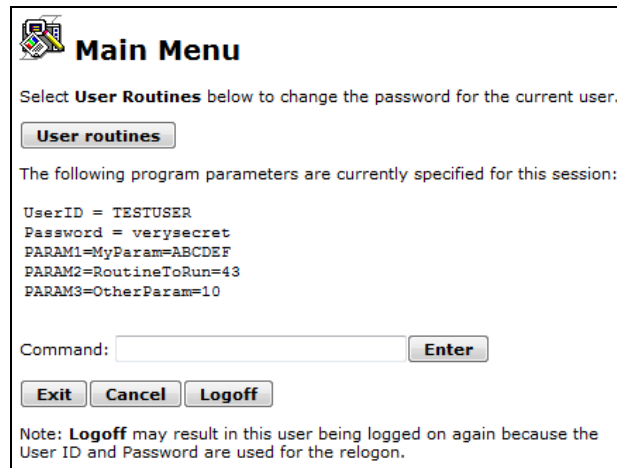
Parameters set:

```
PARAM_1=MyParam=ABCDEF
PARAM_2=RoutineToRun=43
PARAM_3=OtherParam=10
```

OK

The page being displayed is called `messagebox.html` and is always used to display all message boxes for the application. Press **OK** to dismiss the page. This will return you to the Login page again. Enter User ID and Password as “TESTUSER” and “mypass”.

The Main page is then display as:



Main Menu

Select **User Routines** below to change the password for the current user.

User routines

The following program parameters are currently specified for this session:

```
UserID = TESTUSER  
Password = verysecret  
PARAM1=MyParam=ABCDEF  
PARAM2=RoutineToRun=43  
PARAM3=OtherParam=10
```

Command: **Enter**

Exit **Cancel** **Logoff**

Note: **Logoff** may result in this user being logged on again because the User ID and Password are used for the relogin.

This concludes the Tour...

5 Session Pooling Script

The following XML script ticket.xml is used to handle host session operations in conjunction with Session Pooling.

```
<?xml version="1.0"?>

<SessionPoolingScript
  version="1.00"
  implclass="se.entra.phantom.server.DefaultSessionPoolingHandler"
  runtimeid="TICKET">

  <extensions>
    <!-- No extensions -->
  </extensions>

  <actions>

    <!-- Start Script: check for LOGIN screen -->
    <action name="START" maxtime="20">
      <script>
        <!-- Dispose if not LOGIN screen -->
        <screen name="LOGIN">
          <return true=""/>
        </screen>
        <dispose/>
      </script>
    </action>

    <!-- Ping Script: check for LOGIN screen -->
    <action name="PING" maxtime="1">
      <script>
        <!-- Dispose if not LOGIN screen -->
        <screen name="LOGIN">
          <return true=""/>
        </screen>
        <dispose/>
      </script>
    </action>

    <!-- Check Script: check for LOGIN screen -->
    <action name="CHECK" maxtime="1">
      <script>
        <!-- Dispose if not LOGIN screen -->
        <screen name="LOGIN">
          <return true=""/>
        </screen>
        <dispose/>
      </script>
    </action>

    <!-- Reclaim Script: check for LOGIN screen and clear UI/PW fields -->
    <action name="RECLAIM" maxtime="1">
      <script>
        <!-- Dispose if not LOGIN screen -->
        <screen name="LOGIN">
          <!-- Clean up UI/PW fields and set cursor correctly -->
          <log text="Reclaiming session from LOGIN screen..."/>
          <set hostfield="UI" text=""/>
          <set hostfield="PW" text=""/>
          <set cursor="" hostfield="UI"/>
          <return true=""/>
        </screen>
        <dispose/>
      </script>
    </action>

    <!-- Dispose Script: does nothing -->
    <action name="DISPOSE">
      <script>
        <!-- Does nothing... -->
      </script>
    </action>

  </actions>

</SessionPoolingScript>
```