

Sample Domino Web Server Application

This is a sample application that allows users to View their output on the JES2 spool Queue and allows them to submit a limited number of job based on a personal profile.

JSSMain.html

```
<!DOCTYPE HTML PUBLIC "-//SoftQuad//DTD HoTMetaL PRO
5.0::19980626::extensions to HTML 4.0//EN" "hmpro5.dtd">

<HTML>

<HEAD>
<TITLE>Job Submission System</TITLE>
</HEAD>

<FRAMESET ROWS="67,*">
<FRAME SRC="JSSMainTitle.html" SCROLLING="NO">
<FRAMESET COLS="200,*">
<FRAME SRC="JSSMainMenu.html">
<FRAME SRC="JSSMainScreen.html" NAME="JSSMain">

</FRAMESET>
<NOFRAMES>

<BODY BGCOLOR="#FFFFFF">
<P><I>This document requires a browser that can view frames.</I></P>
</BODY>
</NOFRAMES>

</FRAMESET>
</HTML>
```

JSSMainTitle.html

```
<!DOCTYPE HTML PUBLIC "-//SoftQuad//DTD HoTMetaL PRO
5.0::19981022::extensions to HTML 4.0//EN" "hmpro5.dtd">

<HTML>

<HEAD>
<TITLE>Job Submission System</TITLE>
</HEAD>

<BODY>
<H1>Job Submission System</H1>
</BODY>
</HTML>
```

JSSMainMenu.html

```
<!DOCTYPE HTML PUBLIC "-//SoftQuad//DTD HoTMetaL PRO
5.0::19981022::extensions to HTML 4.0//EN" "hmpro5.dtd">

<HTML>
<HEAD>
<BASE HREF="http://193.128.63.220/">
<TITLE></TITLE>
</HEAD>

<BODY>
<H1>Menu</H1>
<HR>
<P><A HREF="cbility/cgi-bin/JSSViewJobDir" TARGET="JSSMain">View Job
Directory
</A><BR>
<A HREF="cbility/cgi-bin/JSSViewJobStatus" TARGET="JSSMain">View Job
Status
</A><BR>
</BODY>
</HTML>
```

JSSMainScreen

```
<!DOCTYPE HTML PUBLIC "-//SoftQuad//DTD HoTMetaL PRO
5.0::19981022::extensions to HTML 4.0//EN" "hmpro5.dtd">

<HTML>

<HEAD>
<TITLE>Job Submission System</TITLE>
</HEAD>

<BODY>
<H1>Job Submission System</H1>
</BODY>
</HTML>
```

JSSPurgeJoboutput

```
/* rexx */
/*****
*
*   This utility purges the output from a job
*
*****/
/

arg userid jobinfo
scroll = 300
jobinfo = translate(jobinfo, ' ', '+')
jobgrp = word(jobinfo,1)
jobname = word(jobinfo,2)

cmd = 'ST'
viewcmd=cmd
m.1='PREFIX '||userid||'*'
m.2='OWNER *'
m.3='ST'
m.4= 'DOWN ' jobgrp
m.5= 'SET CONFIRM OFF'
m.6= 'F' jobname
m.7='++P'
m.0=7
call dosdsf
say '<H2>Job Output Deleted</H2>'
"JSSViewJobStatus refresh"
return 0

dosdsf:
sdsflns = 32
if m.0=0 then return
call bpxwdyn 'alloc fi(isfout) space(1,3) cyl new msg(2)'
call bpxwdyn 'alloc fi(isfin) space(1,1) tracks new msg(2)'
address mvs 'execio' m.0 'diskw isfin (fini stem m.'
prm='++' || sdsflns',300' /* cols */
address linkmvs 'sdsf prm'
m.0=0
address mvs 'execio * diskr isfout (fini stem m.'
call bpxwdyn 'free fi(isfout)'
call bpxwdyn 'free fi(isfin)'
return 0
```

JSSSubmitJob

```
/* rexx */
/*****
*
*   This submits a users job
*
*****/
/

/* Parse incoming CGI data          */

say 'Content-type: text/html'
say ' '
say '<PRE>'
say
eof = 'N'
pos = 0
do j=1 to __environment.0
  if substr(__environment.j,1,12) = '_BPX_USERID=' then do
    userid = substr(__environment.j,13,(length(__environment.j)-12))
  end
  if substr(__environment.j,1,13) = 'QUERY_STRING=' then do
    string = __environment.j
    string = translate(string,' ','&')
    string = substr(string,14,(length(string)-13))
    do count = 1 to words(string)
      data = word(string,count)
      pos = index(data,'=')
      var = substr(data,1,(pos-1))
      val = substr(data,(pos+1),(length(data)-pos))
      assign = var||"="||val||" "
      interpret assign
    end
  end
end

/*
/* Now lets plug the jobcard and JCL together and
/* submit the resulting JCL
/*
/*

trace 'I'
jobcardfile = 'data/'||userid||'.jobcard'
jobstepfile = 'data/'||jobname||'.jcl'
tmpfile = 'data/'||userid||'.tmp'
"cat "jobcardfile jobstepfile|" >"tmpfile
"submit " tmpfile
"rm " tmpfile
say '<H1>Job submitted !</H1>'
exit
```

JSSViewJobAction

```
/* rexx */
/*****
*
* This utility processes an action from the Job status page
*
*****/
/

/* Parse incoming CGI data */

say 'Content-type: text/html'
say ' '
say '<PRE>'
say
pos = 0
do j=1 to __environment.0
  if substr(__environment.j,1,12) = '_BPX_USERID=' then do
    userid = substr(__environment.j,13,(length(__environment.j)-12))
  end
  if substr(__environment.j,1,13) = 'QUERY_STRING=' then do
    string = __environment.j
    string = translate(string,' ','&')
    string = substr(string,14,(length(string)-13))
    do count = 1 to words(string)
      data = word(string,count)
      pos = index(data,'=')
      var = substr(data,1,(pos-1))
      val = substr(data,(pos+1),(length(data)-pos))
      assign = var||"="||val||" "
      interpret assign
    end
  end
end
end

/* Call the action processor */

select
  when action = 'View+Job+Output' then do
    "JSSViewJobOutput " userid jobgrp
    return 0
  end

  when action = 'Purge+Job+Output' then do
    "JSSPurgeJobOutput " userid jobgrp
    return 0
  end

  when action = 'Refresh' then do
    "JSSViewJobStatus refresh"
    return 0
  end

  otherwise
```

```

        say 'Script error'
        return 0
    end
end

```

JSSViewJobDir

```

/* rexx */
/*****
*

This utility lists the users job directory

*****/
/

/* Parse incoming CGI data */

say 'Content-type: text/html'
say ''
say '<PRE>'
say
pos = 0
do j=1 to __environment.0
    if substr(__environment.j,1,12) = '_BPX_USERID=' then do
        userid = substr(__environment.j,13,(length(__environment.j)-12))
    end
    if substr(__environment.j,1,13) = 'QUERY_STRING=' then do
        string = __environment.j
        string = translate(string,' ','&')
        string = substr(string,14,(length(string)-13))
        do count = 1 to words(string)
            data = word(string,count)
            pos = index(data,'=')
            var = substr(data,1,(pos-1))
            val = substr(data,(pos+1),(length(data)-pos))
            assign = var||"="||val||""
            interpret assign
        end
    end
end
end

/*
/* Open directory file
/*
call syscalls 'ON'
address syscall
path='data/'||userid||'.jss'
'open (path)',
    0_rdnly,
    000
if retval=-1 then do
    say 'Directory file not opened, error codes' errno errnojr
    return 8

```

```

end
fd=retval
/*
/* Now read the directory file and build a
/* selection list for submission
/*
/* write table headers */

say '<SCRIPT LANGUAGE="JavaScript1.1">'
say '<!--'
say 'function subform(form)'
say '{'
say ' form.submit();'
say '}'
say '// -->'
say '</SCRIPT>'
say '<FORM name="form1" ACTION="/cbility/cgi-bin/JSSSubmitJob"'
say 'METHOD=GET>'
say '<table width=60% frame=border border=8 rules=rows cols=3
cellspacing=5 cellpadding=5>'
say '<TR><TH></TH><TH>Jobname</TH><TH>Description</TH></TR>'

count = 0
retval = 81
do while retval = 81
  'read' fd 'bytes 81'
  if retval /= 81 then leave
  if count > 10 then exit 8
  select
    when substr(bytes,1,1) = '#' then iterate
    when substr(bytes,1,7) = 'JOB CARD' then iterate
    when substr(bytes,1,4) = 'JOB ' then do
      jobname = substr(bytes,5,8)
      desc = substr(bytes,14,40)
      say '<TR><TD><INPUT TYPE="radio" NAME="jobname"
VALUE=" ' || jobname || ' "></TD>'
      say '<TD>' || jobname || '</TD><TD>' || desc || '</TD></TR>'
    end
    otherwise do
      nop
    end
  end
  count = count + 1
end
'close' fd

/* Now lets add the Buttons for actions */
say '</TABLE>'
say '<INPUT TYPE="submit" Value="Submit Job" NAME="action">'
say '</FORM>'
return 0

```

JSSViewJobOutput

```
/* rexx */
/*****
*
*   This utility lists the output from a job
*
*****/
/

arg userid jobinfo
scroll = 300
jobinfo = translate(jobinfo,' ','+')
jobgrp = word(jobinfo,1)
jobname = word(jobinfo,2)
say userid
say jobgrp
say jobname

/* write form headers */
say '<FORM name="form1" ACTION="/cbility/cgi-
bin/JSSViewJobOutputScroll"'
say 'METHOD=GET>'
/*
cmd = 'ST'
viewcmd=cmd
m.1='PREFIX '||userid||'*'
m.2='OWNER *'
m.3='ST'
m.4= 'DOWN ' jobgrp
m.5= 'PRINT FILE SDSFDD'
m.6= 'F' jobname
m.7='++XC'
m.0=7
if bpxwdyn('alloc fi(sdsfdd) new space(5,5) cyl',
'msg(2)')<>0 then
do
say 'Temp data set allocation failed'
address syscall 'sleep 6'
end
call dosdsf
address mvs 'execio 'scroll' diskr sdsfdd (fini stem say.'
call bpxwdyn 'free fi(sdsfdd)'
do count = 0 to scroll
say say.count
end
/* write form button */
say '<INPUT TYPE="submit" Value="Scroll Down" NAME="action">'
/* write current position etc for scrolling*/
say '<INPUT TYPE="hidden" Value="0" NAME="scrolpos">'
say '<INPUT TYPE="hidden" Value=" '||jobgrp||' " NAME="jobgrp">'
say '<INPUT TYPE="hidden" Value=" '||jobname||' " NAME="jobname">'
say '</FORM>'
return 0
```

```

dosdsf:
  sdsflns = 32
  if m.0=0 then return
  call bpxwdyn 'alloc fi(isfout) space(1,3) cyl new msg(2)'
  call bpxwdyn 'alloc fi(isfin) space(1,1) tracks new msg(2)'
  address mvs 'execio' m.0 'diskw isfin (fini stem m.'
  prm='++' || sdsflns',300'      /* cols */
  address linkmvs 'sdsf prm'
  m.0=0
  address mvs 'execio * diskw isfout (fini stem m.'
  call bpxwdyn 'free fi(isfout)'
  call bpxwdyn 'free fi(isfin)'
  return 0

```

JSSViewJobOutputScroll

```

/* rexx */
/*****
*
*
*   This utility scrolls up or down the output from a job
*
*****
/

/* Parse incoming CGI data      */

say 'Content-type: text/html'
say ''
say '<PRE>'
say
eof = 'N'
pos = 0
do j=1 to __environment.0
  if substr(__environment.j,1,12) = '_BPX_USERID=' then do
    userid = substr(__environment.j,13,(length(__environment.j)-12))
  end
  if substr(__environment.j,1,13) = 'QUERY_STRING=' then do
    string = __environment.j
    string = translate(string,' ','&')
    string = substr(string,14,(length(string)-13))
    do count = 1 to words(string)
      data = word(string,count)
      pos = index(data,'=')
      var = substr(data,1,(pos-1))
      val = substr(data,(pos+1),(length(data)-pos))
      assign = var||"="||val||""
      interpret assign
    end
  end
end
end

```

```

end

scroll = 50

/* What are we doing - scrolling up or down ??? */

select

    when action = 'Scroll+Up' then do
        scrolpos = scrolpos - scroll
        if scrolpos < 0 then scrolpos = 0
    end

    when action = 'Scroll+Down' then do
        scrolpos = scrolpos + scroll
    end

    otherwise do
        say 'Error scrolling '
        return 8
    end

end

end

/*
cmd = 'ST'
viewcmd=cmd
m.1='PREFIX '||userid||'*'
m.2='OWNER *'
m.3='ST'
m.4= 'DOWN ' jobgrp
m.5= 'PRINT FILE SDSFDD'
m.6= 'F' jobname
m.7='++XC'
m.0=7
if bpxwdyn('alloc fi(sdsfdd) new space(5,5) cyl',
    'msg(2)')<>0 then
    do
        say 'Temp data set allocation failed'
        address syscall 'sleep 6'
    end
    call dosdsf
    scrolend = scrolpos+scroll
    do count = 0 to scrolend
        address mvs 'execio 1 diskr sdsfdd (stem say.'
        if rc = 2 then do
            eof = 'Y'
            leave
        end
        if count < scrolpos then iterate
        say say.1
    end
    address mvs 'execio 1 diskr sdsfdd (fini'
    call bpxwdyn 'free fi(sdsfdd)'
/* write form headers */
say '<FORM name="form1" ACTION="/cbility/cgi-
bin/JSSViewJobOutputScroll"'

```

```

say 'METHOD=GET>'
/* write form button */
if eof = 'N' then do
  say '<INPUT TYPE="submit" Value="Scroll Down" NAME="action">'
end
if scrolpos > 0 then do
  say '<INPUT TYPE="submit" Value="Scroll Up" NAME="action">'
end
/* write current position etc for scrolling*/
say '<INPUT TYPE="hidden" Value="'|scrolpos|'|' NAME="scrolpos">'
say '<INPUT TYPE="hidden" Value="'|jobgrp|'|' NAME="jobgrp">'
say '<INPUT TYPE="hidden" Value="'|jobname|'|' NAME="jobname">'
say '</FORM>'
return 0

```

dosdsf:

```

sdsflns = 32
if m.0=0 then return
call bpxwdyn 'alloc fi(isfout) space(1,3) cyl new msg(2)'
call bpxwdyn 'alloc fi(isfin) space(1,1) tracks new msg(2)'
address mvs 'execio' m.0 'diskw isfin (fini stem m.'
prm='++' || sdsflns',300' /* cols */
address linkmvs 'sdsf prm'
m.0=0
address mvs 'execio * diskr isfout (fini stem m.'
call bpxwdyn 'free fi(isfout)'
call bpxwdyn 'free fi(isfin)'
return 0

```

JSSViewJobStatus

```
/* rexx */
/*****
*

This utility lists the users job status - SDSF ST

*****/
/

/* Parse incoming CGI data */

arg refresh
if refresh = '' then do
  say 'Content-type: text/html'
end
say ''
say '<PRE>'
say
pos = 0
do j=1 to __environment.0
  if substr(__environment.j,1,12) = '_BPX_USERID=' then do
    userid = substr(__environment.j,13,(length(__environment.j)-12))
  end
  if substr(__environment.j,1,13) = 'QUERY_STRING=' then do
    string = __environment.j
    string = translate(string,' ','&')
    string = substr(string,14,(length(string)-13))
    do count = 1 to words(string)
      data = word(string,count)
      pos = index(data,'=')
      var = substr(data,1,(pos-1))
      val = substr(data,(pos+1),(length(data)-pos))
      assign = var||"="||val||"|"
      interpret assign
    end
  end
end
end

/* */
cmd = 'ST'
viewcmd=cmd
m.1='PREFIX '|userid|'|*'
m.2='OWNER *'
m.3=cmd
m.0=3
call dosdsf
/* write table headers */

say '<SCRIPT LANGUAGE="JavaScript1.1">'
say '<!--'
say 'function subform(form)'
```

```

say '{'
say ' form.submit();'
say '}'
say '// -->'
say '</SCRIPT>'
say '<FORM name="form1" ACTION="/cbility/cgi-bin/JSSViewJobAction" '
say 'METHOD=GET>'
say '<table width=60% frame=border border=8 rules=rows cols=4
cellspacing=5 cellpadding=5>'
say '<TR><TH></TH><TH>Jobname</TH><TH>Jobid</TH><TH>Status</TH></TR>'
row = 1
do count = 98 to m.0
  if substr(m.count,3,7) = 'COMMAND' then iterate
  if substr(m.count,3,3) = 'NP ' then iterate
  jobname = substr(m.count,8,8)
  jobid = substr(m.count,17,8)
  status = substr(m.count,40,10)
  if strip(jobname) ^= '' then do
    say '<TR><TD><INPUT TYPE="radio" NAME="jobgrp" VALUE="" ||row
jobname|| "'></TD>'
    say
'<TD>' || jobname || '</TD><TD>' || jobid || '</TD><TD>' || status || '</TD></TR>'
    end
    row = row + 1
/* say m.count */
end
say '</TABLE>'
/* Now lets add the Buttons for actions */
say '<TABLE><TR><TD>'
say '<INPUT TYPE="submit" Value="Refresh" NAME="action">'
say '<TD>'
say '<INPUT TYPE="submit" Value="View Job Output" NAME="action">'
say '<TD>'
say '<INPUT TYPE="submit" Value="Purge Job Output" NAME="action">'
say '</TABLE>'
say '</FORM>'
return 0

```

dosdsf:

```

sdsflns = 32
if m.0=0 then return
call bpxwdyn 'alloc fi(isfout) space(1,3) cyl new msg(2)'
call bpxwdyn 'alloc fi(isfin) space(1,1) tracks new msg(2)'
address mvs 'execio' m.0 'diskw isfin (fini stem m.'
prm='++' || sdsflns',300' /* cols */
address linkmvs 'sdsf prm'
m.0=0
address mvs 'execio * diskw isfout (fini stem m.'
call bpxwdyn 'free fi(isfout)'
call bpxwdyn 'free fi(isfin)'
return 0

```

jvm.properties

```
# @(#)jvm.properties      1.81 97/12/02
#
# Configuration properties for JVM and plugin dll start-up
#

# System Properties
ServletExpressVersion=1.0.0
server.root=/usr/lpp/ServletExpress
server.name=ServletExpress
java.compiler=

# NCF Properties
ncf.service.name=servletservice
ncf.service.class=com.ibm.ServletExpress.service.SEServlet
ncf.plugin.classname=com.ibm.ServletExpress.ServletSystem

#
# Enable native DLL plugin logging by setting 'ncf.native.logison'
# to 'true'.  Change 'ncf.native.logfile' to the <fully-qualified >
# path of an alternate file location if desired.
#
ncf.native.logison=true
ncf.native.logfile=/u/cbility/web/server_root/logs/native.log

#
# Enable JVM logging by setting 'ncf.jvm.stdoutlog.enabled'
# to true.  Change 'ncf.jvm.stdoutlog.file' to 'false' to write
# to a Java debugging console or 'true' for output to a log file.
# Change 'ncf.jvm.stdoutlog.filename' to the <fully-qualified >
# path of an alternate file location if desired.
#
ncf.jvm.stdoutlog.enabled=true
ncf.jvm.stdoutlog.file=true
ncf.jvm.stdoutlog.filename=/u/cbility/web/server_root/logs/ncf.log

# NCF - Admin Service Properties for BasicNCFConfig Applet
ncf.jvm.classpath=/usr/lpp/java/J1.1/lib/classes.zip:/u/cbility/web/ser
ver_root/servlets:/usr/lpp/ServletExpress/lib:/usr/lpp/ServletExpress/s
ervlets:/usr/lpp/ServletExpress/web/classes:/usr/lpp/ServletExpress/web
/admin/classes/seadmin.jar:/usr/lpp/ServletExpress/lib/servexp.jar:/usr
/lpp/ServletExpress/lib/jsdk.jar:/usr/lpp/ServletExpress/lib/jst.jar:/u
sr/lpp/ServletExpress/web/admin/classes
ncf.jvm.libpath=/bin:./usr/sbin:/usr/lpp/internet/bin:/usr/lpp/interne
t/sbin:/usr/lpp/ldap/bin:/usr/lpp/java/J1.1/bin
ncf.jvm.path=/bin:./usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/s
bin:/usr/lpp/ldap/bin:/usr/lpp/java/J1.1/bin
ncf.jvm.use.system.classpath=false

# Max Java Heap Size
ncf.jvm.mx=67108864

#
# Properties for Netscape webserver V2.01 on AIX or SOLARIS
```

```

#
#ncf.native.outofproc.runscript=/usr/bin/servlet_eng_runner.sh
#ncf.native.outofproc.port=8090
#ncf.native.outofproc.idstring="servexp"
#ncf.native.outofproc.netscapemime=<netscape_root>/config/mime.types

#
# Properties for Apache webserver on AIX or SOLARIS
#
#ncf.native.apache.outofproc.runscript=/usr/bin/apache_servlet_eng_runn
er.sh
#ncf.native.apache.outofproc.port=8082
#ncf.native.apache.outofproc.idstring="apache-servlet-engine"

# Properties for IIS
ncf.native.iis.extensionloc=/sePlugins/iis20.dll

# Properties for Domino Go
ncf.native.httpd.cnf.path=/u/cbility/web/config/httpd.conf

#

```

httpd.envvars

```

PATH=/bin:./usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:/usr
/lpp/ldap/bin:/usr/lpp/java/J1.1/bin
SHELL=/bin/sh
TZ=GMT0
LANG=C
LC_ALL=en_US.IBM-1047
NLSPATH=/usr/lib/nls/msg/%L/%N:/usr/lpp/internet/%L/%N:/usr/lpp/ldap/li
b/nls/msg/%L/%N
LIBPATH=/usr/lpp/internet/bin:/usr/lpp/internet/sbin:/usr/lpp/ldap/lib:
/usr/lpp/java/J1.1/lib/mvs/native_threads:/usr/lib
JAVA_HOME=/usr/lpp/java/J1.1
CLASSPATH=/usr/lpp/java/J1.1/lib/classes.zip:/u/cbility/web/server_root
/servlets:/usr/lpp/ServletExpress/lib:/usr/lpp/ServletExpress/servlets:
/usr/lpp/ServletExpress/web/classes
STEPLIB=CURRENT
_BPX_SHAREAS=YES
_BPX_SPAWN_SCRIPT=YES

```