

EC-2000-007  
II-A-015

# Report of Biometric In-house Test<sup>1</sup>

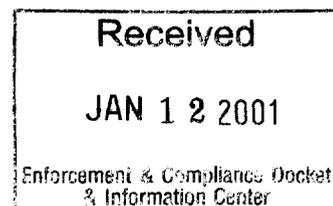
*Web-based Submission of the Discharge Monitoring Report*  
2

EPA Contract #68-W5-0030<sup>3</sup>

Delivery Order #0004

**Revised September 30, 1999**

1	Scope	2
2	Discussion of Specific Test Results	2
2.1	Interaction of Graphics Tablet Drivers with the Operating System and Other Drivers	3
2.2	Vertical Orientation of Printed Handwritten Signature	4
2.3	Data Exchange Format Incompatibility Between the Adobe Acrobat Exchange Plug-in and the Application Server	4
2.4	PenOp Dynamic Link Library Version Incompatibility	4
2.5	Intermittent Server Errors	5
3	E-mail Messages Related to Biometric In-house Test Results	6
3.1	Discussion of the Data Exchange Format Between the Adobe Acrobat Exchange Plug-in and the HAHTsite Application Server	6
3.2	Discussion Related to PenOp DLL Versions	10
3.3	Intermittent Server Errors	11



## 1 Scope

This document describes the results of in-house testing of biometric digital signature components in preparation for Phase 2 of a pilot test of the Web-based submission of the New York State Discharge Monitoring Report (DMR) conducted in the State of New York June – November, 1999. The technical issues described in this document include those identified prior to the installation of biometric hardware and software components on the pilot participant's computers and extend to tests performed on these initial installations during September of 1999 with assistance from the New York State Department of Environmental Conservation and the pilot participants.

## 2 Discussion of Specific Test Results

The in-house testing period revealed problems and issues related to the interaction of the hardware device used to capture the biometric handwritten digital signatures with the client computer operating system and with drivers for other hardware components, such as the mouse pointing device. Printing DMR forms signed with a biometric handwritten signature showed cases in which the vector image of the handwritten signature was displayed vertically, rather than horizontally, on the printed output. In-house testing also revealed a problem related to the handling of data exchange formats between the client-side Adobe Acrobat Exchange electronic form plug-in and the server-side HAHTsite application server. A compatibility problem between two different versions of a dynamic link library file provided by the PenOp biometric software was also identified. Intermittent server errors reported by the application server and traced to either the biometric signature validation process or the SQL database were observed and not fully resolved during the in-house test period and experience with the early installs in the Phase 2 pilot.

In general the architecture of the server-side processes required to validate the biometric handwritten signature proved more challenging than the validation of cryptographic signatures because the biometric signature validation process requires that the Adobe Acrobat Exchange form application with the PenOp biometric signature plug-in must run on the server and be controlled by an automated process to load the submitted form and signature data and then execute the PenOp plug-in to perform the signature validation. Since the Adobe Acrobat Exchange form is a single-threaded application, biometric signature validations must be queued and presented one at a time to a single instance of the Adobe Acrobat Exchange application running on the server. Appendix A contains examples of E-mail messages which reveal vignettes of some of the experience and discussion which occurred during the biometric in-house testing period.

Specific biometric in-house test results are grouped by category in the subsections below.

### ***2.1 Interaction of Graphics Tablet Drivers with the Operating System and Other Drivers***

Tests of the installation of the CalComp UltraSlate graphics tablet and TabletWorks Version 5.0 software revealed a difference in the behavior of the software drivers across different computers. Installation of the TabletWorks 5.0 software on one IBM ThinkPad 770 laptop running Windows 95 prevented the laptop from booting up as long as TabletWorks drivers were installed on the laptop. A duplicate test with another IBM ThinkPad 770 laptop also running Windows 95 did not show this behavior. When the same laptop was booted up in the Windows NT 4.0 operating system, however, the presence of graphics tablet drivers disabled the mouse, the graphics tablet and keyboard until the TabletWorks drivers were updated to Version 5.2. A potential compatibility problem between the graphics tablet drivers and the Logitech mouse drivers was identified in discussions with CalComp technical support. CalComp technical support also stated that the Win32 and Windows 95 versions of the graphics tablet drivers may have different behaviors, and that it may be necessary to try an alternate driver version in some cases.

During initial tests of the graphics tablet during installations on the pilot participant's computers at the pilot participant's facilities, a Windows 98 computer at Indeck Energy Systems recognized the graphics tablet with Version 5.0 software, but a similar Windows 98 computer at Montgomery County Sanitary District No. 1 did not recognize the graphics tablet until the graphics drivers were updated to Version 5.2.

In summary, test experience with the installation of the graphics tablet suggested that each computer must be treated as an individual case with respect to the behavior of different versions of the graphics tablet software drivers.

### ***2.2 Vertical Orientation of Printed Handwritten Signature***

When pilot participants completed the DMR form within Adobe Acrobat Exchange and then signed the form with their biometric handwritten signature, it was observed that the printed representation of the form showed the vector image of the handwritten signature in a vertical, rather than horizontal orientation.

### ***2.3 Data Exchange Format Incompatibility Between the Adobe Acrobat Exchange Plug-in and the Application Server***

A data exchange format incompatibility was observed related to the interaction of the client-side electronic form environment, Adobe Acrobat Exchange Version

3.01, with the application server, HAHTsite Versions 3.1 and 4.0. During implementation of the server-side biometric signature validation process, it was observed that the application server process which handled the submitted FDF data stream received from the client-side Adobe Acrobat Exchange form, and then launched the server-side biometric signature validation process, was executed in two duplicate instances with a time separation of approximately 4 seconds. Since the Adobe Acrobat Exchange electronic form application is single-threaded, the second instance of the application server process launched a second instance of the Adobe Acrobat Exchange form application on the server. This second Adobe Acrobat Exchange instance failed because only one instance can run at any one time.

This problem was ultimately traced to a poorly documented programming convention which is used to indicate whether Adobe FDF or Web HTML data format is being passed from the application server to the Adobe Acrobat Exchange browser plug-in on the client computer. When the Adobe Acrobat Exchange browser plug-in on the client received a data stream from the server in an unexpected format, the plug-in responded with a duplicate data stream sent to the application server.

#### **2.4 *PenOp Dynamic Link Library Version Incompatibility***

During in-house testing and in initial testing of installations at the pilot participant's facilities, it was observed that biometric handwritten signatures created on client computers running Windows NT Workstation 4.0 were reliably enrolled and verified on the server running Windows NT Server 4.0. However, biometric handwritten signatures created on client computers running Windows 95 or 98 were not reliably enrolled and verified on the server. This problem was traced to a version incompatibility between one PenOp dynamic link library file which was used on both the client and server computers. This file was updated on the client computers to match the version on the server to resolve this problem.

#### **2.5 *Intermittent Server Errors***

During in-house testing and in initial tests of installations performed at the pilot participant's facilities, intermittent server errors were sometimes observed after biometric enrollment form was submitted to the server. These error messages sometimes identified a problem with the biometric signature enrollment or validation process, or with the SQL database. Tests showed that a repeat of the enrollment process would in many cases result in a successful enrollment without any known change made at the client or at the server. The cause of these intermittent server errors remained unresolved at the time of this report.

## *Appendix A*

### 3 E-mail Messages Related to Biometric In-house Test Results

The following E-mail messages are related to some of the issues identified during in-house testing of the biometric handwritten signature mechanism used in Phase 2 of the DMR pilot. These example messages provide a window into the nature and detail of some of the discussions related to these tests within a time snapshot defined by the scope of the individual E-mail messages. These messages do not track any given issue from conception to resolution.

#### 3.1 *Discussion of the Data Exchange Format Between the Adobe Acrobat Exchange Plug-in and the HAHTsite Application Server*

-----Original Message-----

From: Michael Kelley [mailto:michaelk@haht.com]  
Sent: Tuesday, September 28, 1999 6:19 PM  
To: Lewis, Todd  
Cc: Liu, WeiShing; 'Tracy Michelle Adewunmi'  
Subject: Re: HAHTsite problem with double insert (Incident #41424)

I believe the result is understandable and predictable. But it has to do with the way the Acrobat Reader plug-in works, and not HAHTtalk Basic code. Here's the explanation (with a little bit of hand-waving):

- 1) When the scenario is kicked off, FDFSetSubmitFormAction() is executed in the HAHTsite Application Server. This is a call to the Adobe FDF Toolkit and it causes FDF to be generated. The "dsp\_btnSubmit#FDF" (or "dsp\_btnSubmit") argument to FDFSetSubmitFormAction() gets stored in the FDF as a string.
- 2) The FDF generated in Step #1 is shipped to the browser. This causes the browser to load the Acrobat Reader plug-in, and the plug-in processes the FDF. The FDF specifies what PDF file to load, and what form controls (text boxes, submit button, etc.) to draw on the PDF as displayed in the browser by the plug-in.
- 3) The browser user enters data into the two text boxes, and clicks the submit button. The Acrobat Reader plug-in then

generates an HTTP POST request to the web server with the URL specified in `FDFSetSubmitFormAction()` in #1 above. This URL specifies "dsp\_btnSubmit" as the page that is to process the POST request. The plug-in also looks to see if the #FDF is present on the end of the URL. If it is present, then the plug-in expects to get FDF back from the POST request. If the #FDF is not present, then the plug-in expects to get regular HTML back. (This is no doubt a clunky mechanism, but that's how Adobe did it.)

4) The web server receives the POST request, and sends it on to the HAHTsite Application Server. The AppServer processes the values in the POST request and executes the code to do the database insert. Then it generates some HTML to send back to the browser (because that's what dsp\_btnSubmit is coded to do).

5) The browser receives the HTML back from the AppServer. If it was expecting HTML, then the right thing happens. The Acrobat Reader plug-in is dropped (either unloaded or just made not visible -- I don't know which) and the HTML is displayed in the usual fashion. At this point, activity stops and no double-submit has occurred.

6) On the other hand, if the browser was expecting FDF, then the Acrobat Reader plug-in receives the HTML, and processes it as if it were FDF. FDF is a command language that tells the Reader plug-in what to do. The Reader plug-in becomes confused because it has HTML instead of FDF, and in its confusion, it does another submit, and we start over again with Step #4.

Evidently, the second time around the browser \*does\* figure out that it is getting HTML (and not FDF), so we don't get stuck in an infinite loop. I don't know the reason for this other than that the Reader plug-in is just confused. If the HTML was different, however, different behavior could ensue. All kinds of strange things might happen.

One of the key clues to this behavior is the timestamps in the web server logs. The first submit is received, and then the second one shows up about 4 seconds later. In the past when I've seen double-submits due to JavaScript errors, both submits essentially went out at the same time, and the web server logs show two submits with the same time stamp (or just 1 second apart). But the 4 second interval in your experiment indicates that the first submit was going out

normally, and it was the \*result\* of that submit going back to the browser that was provoking the browser into doing the submit again.

So in summary, this problem has nothing to do with HAHTtalk Basic or the HAHTsite Application Server. There's no way for HAHTsite to detect this as an error either at compile or runtime as the code is all perfectly legal. It's essentially a logic error in the application in that you told the Reader plug-in to expect FDF, but you actually sent it HTML. Not knowing any better, the Reader plug-in took the HTML, processed it as if it were FDF, and acted accordingly.

-----Original Message-----

From: Lewis, Todd  
Sent: Tuesday, September 28, 1999 5:26 PM  
To: 'michaelk@haht.com'  
Cc: Liu, WeiShing; 'Tracy Michelle Adewunmi'  
Subject: RE: HAHTsite problem with double insert (Incident #41424)

Michael, we will try this and will probably have a result by sometime tomorrow. In the interim, do you see the effect of this change (dsp\_btnSubmit#FDF vs. dsp\_btnSubmit) as understandable and predictable from the way HAHTtalk would be expected to work (i.e., would using the #FDF be predicted to execute a duplicate instance of a section of code based on the logic of the HAHTtalk language)? Or is this the sort of thing that would otherwise return an error at compilation or execution, but just happens (by some mysterious mechanism) in this instance to execute a duplicate instance of code in some computer configurations? [We are having difficulty understanding why the execution of the duplicate instance of code would occur, and, since it does occur, what is really happening internally.]

Thank you,

Todd  
[TLewis@idinc.com](mailto:TLewis@idinc.com)

-----Original Message-----

From: Michael Kelley [<mailto:michaelk@haht.com>]  
Sent: Tuesday, September 28, 1999 4:55 PM  
To: Lewis, Todd  
Cc: Liu, WeiShing; 'Tracy Michelle Adewunmi'

Subject: Re: HAHTsite problem with double insert (Incident #41424)

I have made a simple change that gets rid of the double-insert problem on my system. Please forgive me if somehow I've missed the point in the way that I've made this change. Maybe this is just an artifact of your sample. But perhaps it is the core problem.

In the function name\_initialize() in name.hbs, there is the line of code:

```
rc = FDFSetSubmitFormAction(fdfhandle, "btnSubmit", 3,
baseURL & "dsp_btnSubmit#FDF", 4)
```

You've modified this line to point at "dsp\_btnSubmit" instead of one of the built-in HAHTsite methods for doing the database insert. This is fine. But you have to remove the "#FDF"!! That's because the page "dsp\_btnSubmit" is not returning FDF -- it's returning ordinary HTML. Note that around line 738 of your name.hbs file, there's a comment that describes the need to do this. And I've just been reading through Adobe's FDF Toolkit overview document, and this is alluded to on page 11 in the section "Introduction to FDF".

So I modified the line of code to be:

```
rc = FDFSetSubmitFormAction(fdfhandle, "btnSubmit", 3,
baseURL & "dsp_btnSubmit", 4)
```

and the double-submit no longer occurs.

This still doesn't explain why it works OK on the system where you were running the browser and server on the same computer. But give the above change a try in your larger application and see if it makes a difference.

Let me know what happens!

### **3.2 Discussion Related to PenOp DLL Versions**

-----Original Message-----

From: Support [<mailto:Support@penop.com>]

Sent: Monday, September 27, 1999 6:05 AM

To: Liu, WeiShing

Subject: RE: Enrollment Failure under NT if Signature is

Created under Windows 95/98

You will have to use the same DLL on both the client and Server systems for compatibility. Our recommendation is to standardise on v3.1. The only change you need to make is to replicate the file PENOPCCO.DLL in the windows\system folder. We will look at the btk files shortly.

Regards  
- Gerry

-----Original Message-----

**From:** Lewis, Todd  
**Sent:** Tuesday, September 21, 1999 6:43 PM  
**To:** 'PenOp Support'  
**Cc:** Liu, WeiShing  
**Subject:** Enrollment Failure under NT if Signature is Created under Windows 95/98

Dear Friends, in an operational pilot in the State of New York sponsored by the Environmental Protection Agency, in which the goal is to demonstrate that Adobe Acrobat Exchange 3.01 forms installed as a plug-in to Netscape Navigator 4.51 under Windows NT/95/98 can be signed with PenOp Version 2.6 (using the PenOp Adobe form plug-in) and then sent via an FDF data stream over the Internet to a Web application server (HAHTsite 3.1) under Windows NT Server 4.0 SP 4 where the signature is verified by loading the submitted form data and PenOp signature into an Adobe Acrobat Exchange application with the PenOp plug-in running on the server, the following results were obtained with regard to the enrollment process:

- 1) If the PenOp plug-in to an Adobe Acrobat Exchange form running as a browser plug-in on a client computer with the Windows NT Workstation 4.0 operating system is used to capture 5 signatures for enrollment purposes, and if these signatures are stored in hidden fields in the form and then subsequently transmitted to the server via an FDF data stream, the signatures will be successfully enrolled on the server.
  
- 2) If the PenOp plug-in to an Adobe Acrobat Exchange form running as a browser plug-in on a client computer with the Windows 95 or 98 operating system is used to capture 5 signatures for enrollment purposes, and if these signatures are stored in hidden fields in the form and then subsequently transmitted to the server via an FDF data stream, an attempt to enroll the signatures on the server will usually (but not always) fail with the error message from FDFVerify: Unexpected error (4) enrolling signature. A bitmap image of this error is attached to this E-mail message.



error.bmp

What is "unexpected error (4)" and why does it occur?

Why do the enrollment results differ between Windows NT Workstation 4.0 and Windows 95/98 for the client operating system if the server is NT Server 4.0 SP4?

What is a fix or workaround to this problem?

Thank you,

Todd Lewis  
Information Dynamics, Inc. (IDI)  
TLewis@idinc.com

### 3.3 *Intermittent Server Errors*

-----Original Message-----

**From:** Todd Lewis [mailto:TLewis@idinc.com]  
**Sent:** Wednesday, September 29, 1999 8:37 PM  
**To:** WLiU@idinc.com  
**Cc:** TLewis@idinc.com  
**Subject:** Server error after processing submit on enrollment

WeiShing, I received the following server error after pressing "submit" on the Phase 2 biometric signature enrollment page. I was doing this test from Windows 95 with the updated PenOp DLL and updated CalComp UltraSlate drivers: HAHTsite 3.1 webapps Server reports the following:

The application had a runtime error while processing this page. Report this problem to the site webmaster along with a copy of the URL that caused this message.

HAHTsite 3.1 webapps Server reports the following Error:

The application had an unhandled error while running page dsp\_btnEnrollment. Runtime error 900 occurred at line 434 in hbrcode\Menu: A system exception occurred executing line 434 in routine dsp\_btnEnrollment.

Todd

-----Original Message-----

**From:** Steve Vogler [mailto:sevogler@gw.dec.state.ny.us]  
**Sent:** Tuesday, September 28, 1999 1:13 PM  
**To:** TLewis@idinc.com; wliu@idinc.com  
**Cc:** Chuck Haugh  
**Subject:** Hi Todd; Chuck an I were at Rosendale WWTP this morning doing the PenOp install.

Hi Todd; Chuck an I were at Rosendale WWTP this morning

doing the PenOp install. When Pat Marsh tried to do the enrollment (this was between 10:30 am and 10:45 am) she got the following error messages

- 1) The application had a runtime error while processing this page. Report this problem to the site webmaster.
- 2) The application had an error while running page dsp\_btn enrollment. Run time error 900 occurred at line 434 in hbscode\menu. A system exception occurred executing line 434 in routine dsp\_btn enrollment.

I tried the enrollment on my computer when we got back to the DEC office (at about 12:45) and it looks like it worked O.K. Was there something wrong with the server this morning?

Thanks  
Steve

-----Original Message-----

From: Steve Vogler [<mailto:sevogler@gw.dec.state.ny.us>]  
Sent: Tuesday, September 21, 1999 7:42 AM  
To: TLewis@idinc.com; wliu@idinc.com  
Subject: Error message

Hi Todd; when I tried to login at the Internet Security and Form Submission Prototype screen the following error message occurred.

HAHTsite 3.1 webapps Server reports the following:

The page generated by the application had invalid HTML format: the HTTP headers were incomplete or there was no content. This usually indicates an error in the application. Report this problem to the site webmaster along with a copy of the URL that caused this message.

Deliverable 8.4, InDyne, Inc. (formerly Information Dynamics, Inc.)

<sup>1</sup> A field test in the State of New York of the digital signing and submission of the Discharge Monitoring Report using an Adobe Acrobat Exchange plug-in to a Web browser as the electronic form environment which is connected interactively across the Internet to a receiving Web site. Cryptographic and handwritten biometric digital signatures are evaluated in this pilot.

<sup>2</sup> Submission of Environmental Data Under the Taiwan-USEPA Technical Cooperation Agreement