Optimizing SmartPlant® Instrumentation Delivery through Citrix

Omied Sadeghi

March 2014

SmartPlant Instrumentation has been firmly established as an integral component of project execution and client operations within our industry. However, the increasing number of projects containing multiple licensors and third party participation has strained currently used methods of delivering the application to users. Here, the ability to limit exposure of intellectual property and increase quality is presented.

Background

Deploying SmartPlant Instrumentation using a common configuration file has been a standard method of delivery over Citrix. However, extensive use of misconfigured application sequences or environment variables have resulted in significant impacts to the project. Although these methods have granted the ability to easily apply custom project properties to standard SmartPlant Instrumentation deployments, securing confidential data and end user efficiencies are not being considered. Application administrators rely heavily on user isolation to manage access and work processes within the tool, consequently adding a layer of vulnerability to security or trading off user efficacy within the software. It is widely accepted that user isolation is required for proper change control, thus a solution to accommodate the business requirements will need to be applied.

> When an intools.ini and application temp path is shared
> - The previous user's logon credentials are visible
> - Default properties are frequently restored
> - Users are prevented from defining permanent preferences
> - Orphaned files are generated in application paths
> - Unauthorized data is exposed to all users

Solution Overview

Antiquated vendor supplied documentation provides incomplete instruction and legacy variables when deploying over Citrix, so this must strictly be used for adapting the baseline concepts to the application environment. Familiarity with the application architecture will be required to effectively tune SmartPlant Instrumentation in thin client mode. Present GPOs and roaming profiles will provide a firm foundation for the environment.

Define and configure an application path utilizing common environment variables. Develop the launch scripts to generate user profile parameters for use with application paths. This ensures that each user is provided with a unique configuration file to customize their SmartPlant Instrumentation preferences. In addition, unique temporary directories are available for each user.

As a result, this will ensure IP management protocols are being strictly followed in accordance to licensor requirements and effective use of the application is increased.

Application Server

Identifying the environment variables is pivotal to successful configuration. The operating system, environment paths and registry hives all contain entries requiring modification. In an effort to simplify variables across the spectrum, the drive letter X: will be defined for translating unique user paths to common parameters for SmartPlant Instrumentation.

> Application Environment
> - Windows Server 2008 R2
> - SmartPlant Instrumentation 2013 (10.00.00.0230)
> - Citrix XenApp 6.5 Platinum
> - Oracle 11g (11.2.0.2)

Application Installation

The selected location for installing SmartPlant Instrumentation can vary from the default. As such, the application was installed to “E:\SmartPlant\SPI” during implementation. This path is the primary reference of modifications.

> The entry is present in 3 primary locations
> - HKEY_LOCAL_MACHINE Registry Hive
> - Windows Server PATH System Variable
> - Intools.ini (E:\SMARTPLANT\SPI\intools.ini)

Modifying the Registry

As noted in the previous section, the application path is present in the HKLM hive of the Windows Server registry. This can be found in the following entry:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Intergraph\INtools\10.00.00.0230]
"Path"="E:\\SMARTPLANT\\SPI\\"
```

Figure 1

This string will need to be replaced with the common parameter for SmartPlant Instrumentation

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Intergraph\INtools\10.00.00.0230]
“Path”=”X:\\SMARTPLANT\\SPI\\”
```

Figure 2

Modifying the Windows Server System Variable

To identify the current application path entry for SmartPlant Instrumentation, access the server Environment Variables within Advanced System Settings or echo the output within a command prompt.

```
E:\SMARTPLANT\SPLM\bin;E:\Oracle\product\11.2.0\client_1\bin;C:\windows\system32;C:\windows;C:\windows\
System32\Wbem;C:\windows\System32\WindowsPowerShell\v1.0\;C:\ProgrmFiles\Dell\SysMgt\oma\bin;C:\Progr
amFiles\Dell\SysMgt\shared\bin;C:\ProgramFiles\Dell\SysMgt\idrac;C:\IBM\ITM\bin;C:\IBM\ITM\TMAITM6;C:\Pro
gramFiles(x86)\Citrix\system32\;C:\ProgramFiles(x86)\Citrix\System32\Citrix\IMA;C:\ProgramFiles(x86)\Citrix\Syst
em32\Citrix\IMA\Subsystems;C:\windows\System32\Citrix\IMA;C:\ProgramFiles(x86)\Citrix\system32;C:\Program
Files(x86)\CommonFiles\Citrix\System32\;E:\SMARTPLANT\SPI;C:\IBM\ITM\InstallITM
```

Figure 3

The common parameter “X:\SMARTPLANT\SPI” for SmartPlant Instrumentation will need to be added as a PATH to the servers Environment Variables. **Do not remove the original entry.** As stated in the document, legacy variables for the software to fully function have been inherited to require the original entry for administration. When modifying the PATH, it is crucial to have it placed in a primary position for the application to properly initialize.

```
E:\SMARTPLANT\SPLM\bin;E:\Oracle\product\11.2.0\client_1\bin;C:\windows\system32;C:\windows;C:\windows\
System32\Wbem;C:\windows\System32\WindowsPowerShell\v1.0\;C:\ProgrmFiles\Dell\SysMgt\oma\bin;C:\Progr
amFiles\Dell\SysMgt\shared\bin;C:\ProgramFiles\Dell\SysMgt\idrac;C:\IBM\ITM\bin;C:\IBM\ITM\TMAITM6;C:\Pro
gramFiles(x86)\Citrix\system32\;C:\ProgramFiles(x86)\Citrix\System32\Citrix\IMA;C:\ProgramFiles(x86)\Citrix\Syst
em32\Citrix\IMA\Subsystems;C:\windows\System32\Citrix\IMA;C:\ProgramFiles(x86)\Citrix\system32;C:\Program
Files(x86)\CommonFiles\Citrix\System32\;X:\SMARTPLANT\SPI;E:\SMARTPLANT\SPI;C:\IBM\ITM\InstallITM
```

Figure 4

Modifying the intools.ini

With a common parameter established for SmartPlant Instrumentation, defining global properties within the intools.ini file will translate to user's unique profile. Modify the following sections of the configuration file located in the installation directory **E:\SmartPlant\SPI\**

```
[GENERAL]
HELPPATH=E:\SmartPlant\SPI\
PATHTEMP=X:\SmartPlant\SPI\
[Loop]
PathTemp=X:\SmartPlant\SPI\
[Project]
LogoPath=\\Global Path
UserName=
```

The UserName property

- Highlighted to illustrate vulnerability
- Property will be populated with user's ID
- INI will be unique, removing shared vulnerability

With modification complete, rename the intools.ini to intools.ini.orig

Preparing Citrix Application Launch Script

With the configuration file renamed in **E:\SmartPlant\SPI\**, the application cannot initialize at this point. Referring to the PATH variable, the primary PATH for the application is defined as **X:\SmartPlant\SPI\.** In order to function, a script will be used to establish the user profile variables to agree with the application path modifications prior to being invoked.

The programmable method for generating the application launch script can be performed using various methods. A batch script is illustrated to provide easy interpretation of its use.

Create a blank document using a local text editor and place the following script within the body When this has been completed, save the file as a .cmd

```
@echo off
set __compat_layer=runasinvoker
subst X: %userprofile%
:path
IF EXIST "X:\SmartPlant\SPI" (goto intools) ELSE MD "X:\SmartPlant\SPI\"
:intools
if exist "X:\SmartPlant\SPI\intools.ini" (goto app) ELSE copy "E:\SmartPlant\SPI\intools.ini.orig"
"X:\SmartPlant\SPI\intools.ini"
:app
start pdlice –l
START /D E:\SmartPlant\SPI\ E:\SmartPlant\SPI\INTOOLS.EXE
```

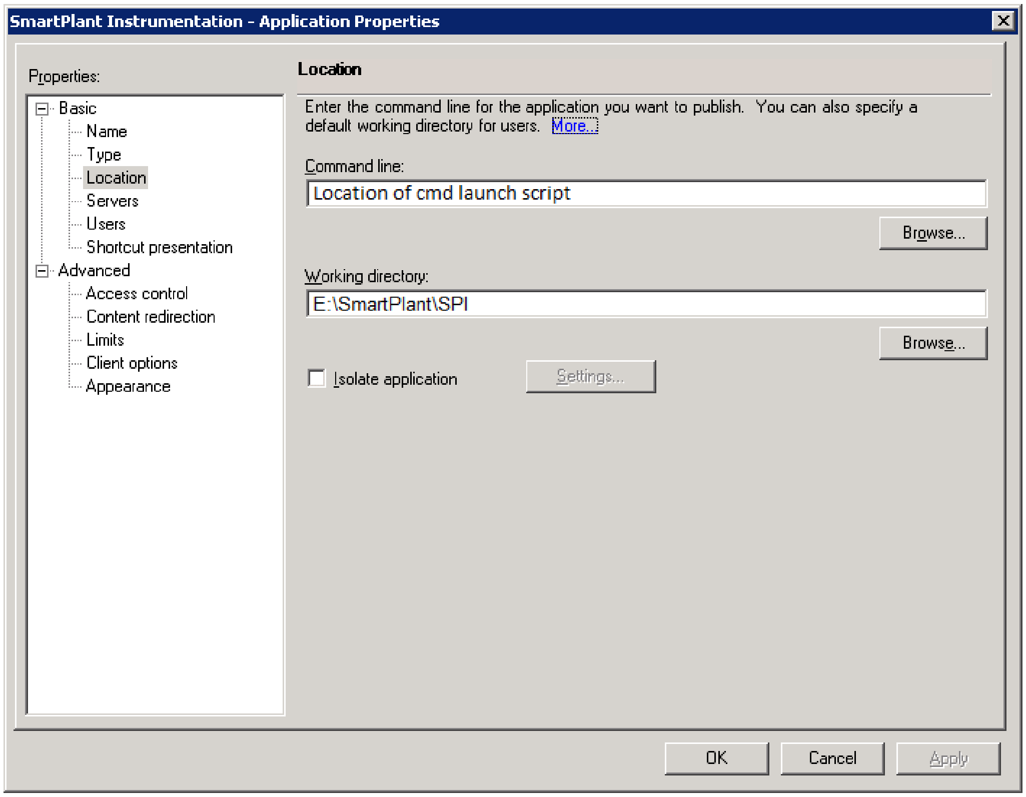
Figure 5

Published Citrix Application Properties

Defining the working directory is crucial for character sets and units of measure to display properly in SmartPlant Instrumentation. A common practice of publishing the application is to leave the default or parent location of the command line executable. This creates numerous engineering challenges and would need to be superseded within the command line of the launch script as illustrated in figure 5, or preferably in the Citrix published application's working directory location properties shown in figure 6.

When the working directory is incorrectly defined

- Hidden characters display
- Units of measure can reset to default values
- Safety and the quality of engineering data decline


SmartPlant Instrumentation - Application Properties

Location

Enter the command line for the application you want to publish. You can also specify a default working directory for users. More...

Command line: Location of cmd launch script

Working directory: E:\SmartPlant\SPI

☐ Isolate application

Figure 6

Multi-Tenant Use of Native Deployment

Additional source intools.ini files can be deployed for various SPI domains to accommodate multiple projects. The intools.ini can be named in accordance to contract taxonomy for easy identification in launch scripts.

When using multiple native SPI domains, consider the following

- Include the respective intools.ini within the launch script
- Add appropriate Oracle connection strings in each INI file
- Version control is shared for native install deployments
- Additional licensing will apply for 3rd party use

To properly designate license servers for SPLM seat reservation, a registry import entry can be included within the launch script illustrated in Figure 5. This will assist projects in maintaining vendor requirements for hosted applications and assist in logical separation of shared resources.

Figure 7 demonstrates an organization specific SPLM registry string for import to the user's profile when executing the launch script.

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Intergraph\SPLM\LicenseMachines]
“CompanyName”=”FQDNServer 10.98.76.54”
```

Figure 7

Native Ghostscripts for Bulk Specification Reports

Driverless printing has demonstrated numerous benefits in managing Citrix farms. Although agile and requiring less maintenance, geographic challenges require careful consideration for alternatives.

Native PDF generation has shown to improve completion times exponentially, and as a result, allow designers to rapidly continue work on a single session/instance of the application. For global participants, a file repository or profile on the local host site can be used for retrieval via FTP

Disclaimer

This document has been modified from its original to protect intellectual property

About the Author

Omied Sadeghi
Information Management
Execution Systems Architect
omied.sadeghi@fluor.com

"Pioneer of hybrid cloud EPC execution platforms"

Linked in profile