Colledia™

WritePrivateProfile

March 7, 2016

Global network of innovation

SIEMENS

1 WritePrivateProfile

A command-line utility for patching .ini files, or other files written using Windows' WritePrivateProfileString() function.

1.1 Introduction

This utility is to automate patching of ini files and db files, or any other file in that format which is in the configuration directory of a project. For example it might be used to set all the 'Simulation' flags for a system under development. This might be done from the 'nosync' batch file run by Workstation Manager.

It is controlled by an xml file which specifies for any workstation which entries are to be written.

From Version 1.0.1 it is able to write to any file.

From version 1.0.2 it is no longer v4 specific (ie only using %CC_ROOT%/%CC_SYSTEM%/config/system'). Ie it will work using the v3 or v1/v2 file locations.

From version 1.0.3 it is able to do patching on any workstation not just the one matching a single <workstation> element. Also a variable "workstation" is automatically created with the current workstation number.

1.2 Parameters

The program takes one parameter, which is the name of an xml file that controls the edits. If the program is run with '–?' or '/?' as parameter, a brief help message is displayed to this effect. If the program is run with no parameters, or more than one, an even briefer message is displayed.

The extension '.xml' will be added if it is not supplied.

The name is first searched for in the configuration directory. If a matching file is not found there the given name will be searched for as a normal absolute or relative file-name.

1.3 Configuration Locations.

The program is able to work using all three conventional locations for the files, both its own configuration file and the dev.ini files being patched.

1.3.1 The V4 location

If the environment variables CC_ROOT and CC_SYSTEM are present, the "V4" location is used.

The configuration file is in a folder specified using environment variables CC_ROOT and CC_SYSTEM.

The config file is first searched for in %CC_ROOT%\%CC_SYSTEM%\config\.

The files being patched are assumed to be in %CC_ROOT%\%CC_SYSTEM%\config\system.

1.3.2 The V3 location

If the environment variables CC_ROOT and CC_SYSTEM are not both present, the "V3" location is used.

The configuration file is in a folder specified in C:\bncs_config.ini. The location of the config files is given by the ConfigPath key.

The files being patched are in a folder given by the SystemPath key.

1.3.3 The v1/v2 location

If neither of the above locations are specified, the "V2" location is used.

The configuration file, and the files to be patched, is in C:\Windows or C:\Winnt, according to the value of the WINDIR environment variable, as also seen using the GetWindowsDirectory function.

1.4 Qualifiers

1.4.1 -var

The program may take any number of '-var' qualifiers. These create variables that may be referred to by the parameter file.

The syntax is '-var:name=value'. This creates a variable called 'name', containing 'value'.

The variables may be used in any attribute in the profile.xml file. They should be referenced in the form '%name%'.

For example:

If '-var:Enable=1 –var:Key=Simulation' were given as the parameter.

```
<setting file="dev_101.ini" section="Grd" key="%Key%"
   value="%Enable%"/>
```

Would be equivalent to :

```
<setting file="dev_101.ini" section="Grd" key="Simulation"
   value="1"/>
```

1.4.2 The "workstation" variable

A variable called "workstation" is always available, containing the workstation number of the workstation it's running on.

1.5 Parameter file

The file that controls the program is an xml file. There is a dtd file available to specify its structure (profile.dtd).

The outermost element is <workstations> of which there should be one.

Within this is a <workstation> element for each workstation to be edited. Each <workstation> element has an 'id' attribute holding the workstation number.

If the 'id' attribute has the value "*" the element will be used when run on any workstation, whereas a numeric 'id' will only be used when run on the workstation with that number. (From version 1.0.3)

Within each <workstation> element are a number of <setting> elements. Each of these has attributes as follows:

'file'	The name of the file to be edited.
'section'	The name of the section within the file.
'key'	The name of the key to be written, within the given section.
'value'	The value to be written to the key.

Filenames are assumed to be relative to the location as described in Section 1.3 Configuration Locations.

From version 1.0.1 it is also possible to put a full path name here. Eg C:\Windows\caplog.ini.

1.6 Main Program Function

The parameter file is read. If a <workstation> element is found whose 'id' attribute value matches the number of the workstation where it is being run, or whose 'id' attribute is '*', the associated <setting> elements are used.

Each <setting> element's attributes give the name of a file to be edited and the section and key to be edited or added, and the value it is to be given.

1.6.1 Example

Using profile.xml with contents as follows.

```
<?xml version="1.0" encoding="utf-8" ?>
<workstations>
  <workstation id="901">
    <setting file="dev_100.ini" section="Grd" key="Simulation" value="1"/>
  </workstation>
  <workstation id="903">
    <setting file="dev_251.ini" section="Configuration" key="Com" value="2"/>
    <setting file="dev_252.ini" section="Configuration" key="Com" value="3"/>
  </workstation>
  <workstation id="*">
    <setting file=" ../ws/%ws_in%/bncs_system.ini" section="Network"
    key="Workstation" value="%ws_in%"/>
  </workstation>
</workstations>
```

The effect of running the command with this parameter file would be as follows:

When run on workstation 901, dev_100.ini would have :

```
[Grd]
Simulation=1
```

Added if this key was not present, and the value rewritten as shown if it was already present.

When run on workstation 903, dev_251.ini would have :

```
[Configuration]
Com=2
```

Added if this key was not present, and the value rewritten as shown if it was already present.

And dev_251.ini would have :

```
[Configuration]
Com=3
```

Added if this key was not present, and the value rewritten as shown if it was already present.

When run on any workstation, the file bncs_system.ini, in a folder under “ws” whose number corresponds to the “ws_in” variable, would include :

[Network]

Workstation=x

Where ‘x’ matched the ‘ws_in’ variable set on the command line.

1.6.2 Bulk generation of bncs_system.ini files

The wildcard option may be used to build sets of ini files for a project, a requirement that particularly arises when many files are required under the ‘ws’ folder.

The following Windows batch file may be used as a template for this activity.

It should be placed in config\ws and the “first” and “last” variables be set to the limits of the set of files required.

```
rem  Generate loads of bncs_system.ini

set first=991
set last=995

set count=%first%

:again

mkdir %count%

rem  Use copy to copy in the template batch file.
copy/y bncs_system_ref.ini %count%\bncs_system.ini

rem  WritePrivateProfile to stamp the workstation number.
WritePrivateProfile Profile -var:ws_in=%count%

set /a count=%count%+1

if %count% leq %last% goto again
```

2 Version Control

Version	Date	Author	Comments
0.1	27/10/2006	Richard Kerry	First version using new template.
0.2	08/01/2009	Richard Kerry	Able to write to any path.
0.3	23/11/2009	Richard Kerry	Able to work with v1/v2/v3 configurations.
0.4	02/09/2010	Richard Kerry	Added wildcard workstation number. Added automatic workstation number variable.