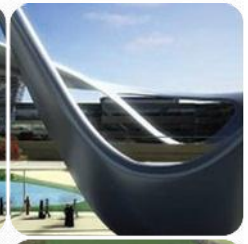
Bentley
Sustaining Infrastructure

MicroStation CONNECT Configuration Changes

Barry Bentley




Topics

Terminology Changes

Directory Structure Differences

Configuration Levels

Configuration Concepts

Configuration File Syntax

Walkthrough of Configuration File Processing

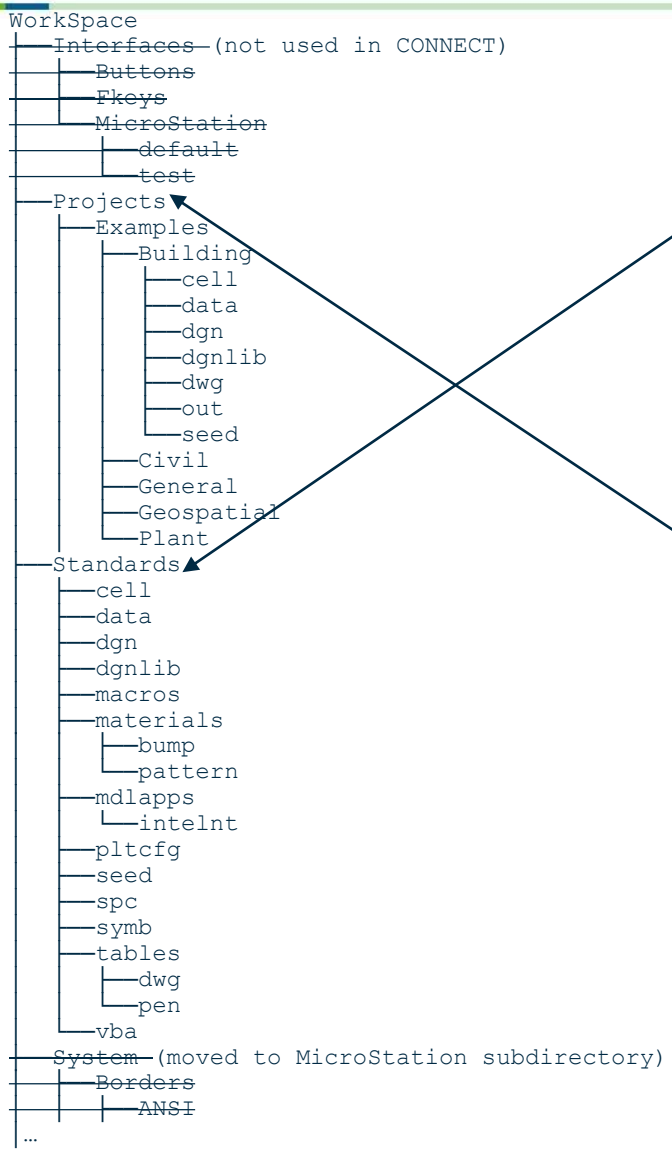
Bentley Sustaining Infrastructure

Terminology Changes

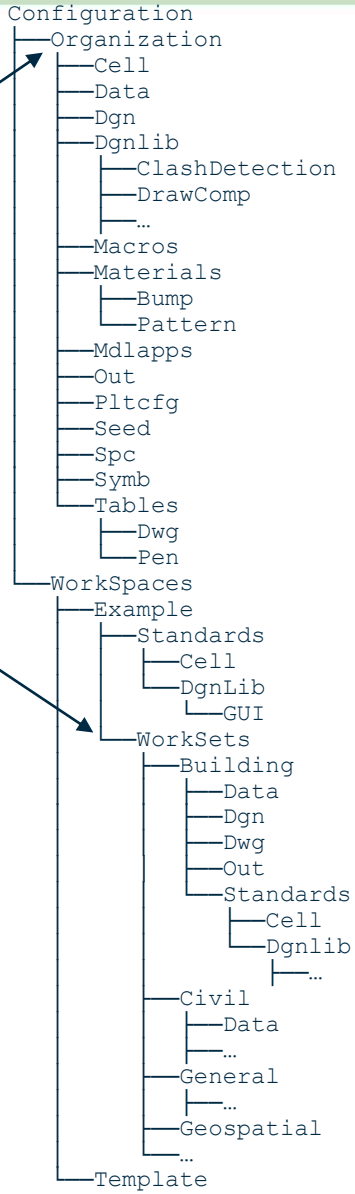
- "WorkSpace" => Configuration
- Site => Organization
- Project => WorkSet
- User now really means User
- Interface gone.
- WorkSets organized by WorkSpace

Directory Structure changes

V8i



CONNECT



Bentley Sustaining Infrastructure

MicroStation\Default

- In V8i, there was a lot of "system" data delivered in the WorkSpace\System directory.
- But some of it differed between versions of MicroStation.
- Made it difficult to make a Configuration that worked reliably with multiple versions of MicroStation.
- That data has been moved to the "Default" subdirectory of MicroStation.
- None of those files should be opened for read/write while running MicroStation.

Configuration Levels

V8i:

- System
- Application
- Site
- Project
- User

CONNECT:

- System
- Application
- Organization
- WorkSpace
- WorkSet
- Role
- User

Configuration Variables defined at "higher" levels (further down the list) override definitions at "lower" levels.

WorkSpace

- Grouping mechanism for WorkSets.
  - This function was served by "User" in V8i, but that had some flaws
  - User Configuration Level was "higher" than Project Configuration Level
  - Changing "User" changed user configuration and user preference files, which really should be "per user".
- Different user organizations will want to use different terminology for the WorkSpace concept (Client, Owner, Department, Facility, or whatever grouping is desired). The label for it in the UI is set by cfg var.
- "Standards" files like fonts, dgnlibs, cells, materials, etc., can be established at the WorkSpace level.
- Pick a WorkSpace, then a WorkSet, then a file.

Role

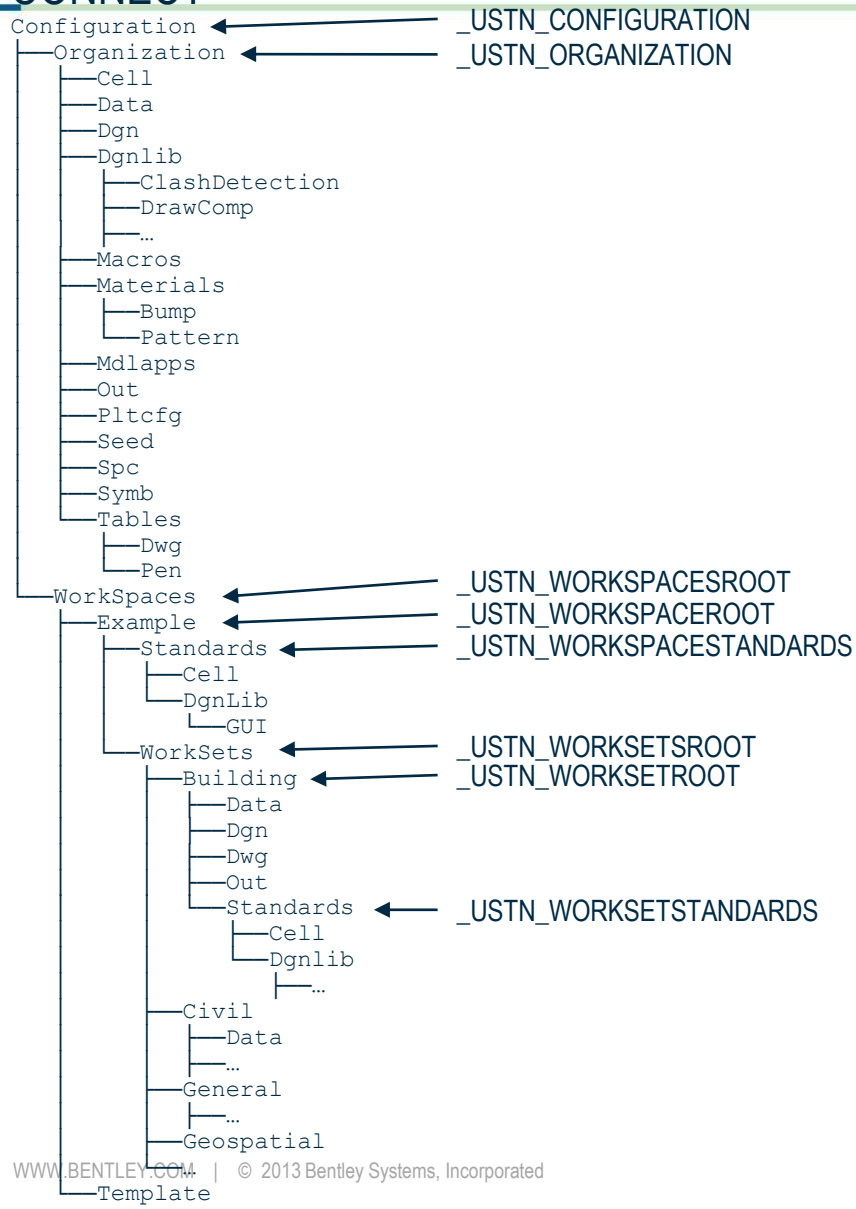
- Users have been asking for an additional Configuration Level to establish role- (or discipline-) based standards.
- Use of this configuration level is completely optional.
- MicroStation does not currently provide a way to set the Role, perhaps that could be a feature of Bentley CONNECT in the future.
- For now, a system environment variable or some "%if" logic in the Organization, WorkSpace, or WorkSet configuration files could cause a Role configuration file to be read.

Configuration Concepts

- Two categories of Configuration Variables:
  - Framework Configuration Variables start with _USTN_, and are generally building blocks for other configuration variables.
    - Often specify installation root directories, or roots of a directory tree where users elect to store standards or working data.
    - Rarely is the value of a FCV checked in program code.
  - Operational Configuration Variables usually start with MS_.
    - Generally specify a directory, file, list of directories, or some other value that directs the flow of MicroStation at runtime.
    - File, directory, and list of directories Operation Configuration Variables often refer to the value of Framework Configuration Variables.

Framework Configuration Variables

CONNECT



Bentley Sustaining Infrastructure

Configuration File Syntax

- Flow Directives control the flow through Configuration Files.
- Variable Directives control certain aspects of Configuration Variables
- Assignment statements set the values of Configuration Variables.
- Expressions and operators manipulate strings or Configuration Variables to yield results that can be used in directives or assignments
- When an assignment refers to other Configuration Variables, the syntax $(OtherCfgVar) defers evaluation of OtherCfgVar until the variable being defined is evaluated. ${OtherCfgCar} evaluates OtherCfgVar immediately.

Flow Directives

- %include
- %if, %else, %elif, %endif
- %ifdef, %ifndef
- %error
- %echo (new in CONNECT)

Variable Directives

- %lock
- %undef
- %level (setting level by name added in CONNECT, i.e. %level WorkSpace

Assignment Statements

Assignment operator	Meaning
=	Assign the Configuration Variable at the current level, regardless of whether it is currently defined. Example: MS_SHEETMODELNAME = 2D Sheet
:	Assign the Configuration Variable at the current level, but only if it not already defined. Example: MS_BACKUP : $(_USTN_OUT)
>	Appends the operand to the existing definition of the variable, separating the existing value and the operand by semicolons (treating the variable as a path). Example: MS_RFDIR > $(_USTN_WORKSETROOT)Borders/
<	Prepends the operand to the existing definition of the variable, separating the existing value and the operand by semicolons (treating the variable as a path). Example: MS_RFDIR <$(_USTN_WORKSETROOT)Borders/
+	Appends the operand to the existing definition of the variable without separator. _USTN_WORKSETDESCR + In Development

Operators

Operator	Syntax	Meaning
basename	basename (<expression>)	Returns the filename of <expression> without directory or extension.
concat	concat (<arg1>,<arg2>…)	Returns the concatenation of the arguments, similar to the + operator, but allows multiple arguments.
devdir	devdir (<expression>)	Returns the device and directory of <expression>, including a trailing directory separator.
dev	dev (<expression>)	Returns the device (e.g., c:) of <expression>.
dir	dir (<expression>)	Returns the directory (without the device) of <expression>.
ext	ext(<expression>)	Returns the file extension of <expression>.
filename	filename (<expression>)	Returns the filename and extension of <expression>.
first	first (<expression>)	Returns the first portion of a expression (i.e., the part preceding the first semicolon).
firstdirpiece	firstdirpiece (<expression>)	Returns the root directory (without device) of <expression>.
lastdirpiece	lastdirpiece (<expression>)	Returns the portion of the directory closest to the file in <expression>.
noext	noext (<expression>)	Returns the full path of <expression>, omitting the extension.
parentdevdir	parentdevdir (<expression>)	Returns the parent directory, including the device, of <expression>.
parentdir	parentdir (<expression>)	Returns the parent directory, excluding the device, of <expression>
registryread	registryread (regvar)	Returns the contents of the registry variable regvar.

Walkthrough

- Configuration files are a simple program.
- Execution starts at mslocal.cfg
  - generated at install time (or build time for programmer builds)
  - includes msdir.cfg, which identifies the MicroStation directory
  - Then includes msconfig.cfg which is the main show.
- msconfig.cfg
  - Completely reorganized, in-line documentation explains what it does.
  - Defines absolutely ***no*** MS_xxx (operational) configuration variables.
  - Don't even think about changing it.

Walkthrough

```
_USTN_HOMEROOT          : $(_USTN_LocalUserAppDataPath)
_USTN_HOMEPREFS         : $(_USTN_HOMEROOT)prefs/

#--------------------------------------------------------------------
# Set directories for the system and application configuration files shipped with MicroStation
#--------------------------------------------------------------------
_USTN_SYSTEM            : $(MSDIR)config/system/
_USTN_APPL              : $(MSDIR)config/appl/

#--------------------------------------------------------------------
# Set Directories for the data files shipped with MicroStation.
#  Note: In MicroStation V8i, this data was in $(_USTN_CONFIGURATION)system/
#--------------------------------------------------------------------
_USTN_SYSTEMROOT        : $(MSDIR)Default/
_USTN_GUIROOT           : $(_USTN_SYSTEMROOT)GUI/
_USTN_SYSTEMTABLES      : $(_USTN_SYSTEMROOT)tables/
_USTN_RASTERTABLE       : $(_USTN_SYSTEMTABLES)raster/
_USTN_RASTERGDALDATA    : $(MSDIR)Gdal_Data/


#--------------------------------------------------------------------
# Define Bentley required applications and file handlers.
#--------------------------------------------------------------------
_USTN_REQUIREDAPPS  : $(MSDIR)mdlsys/required/*.ma
_USTN_FILEHANDLERS  : $(MSDIR)mdlsys/filehandler/*.ma

#--------------------------------------------------------------------
# Include all the delivered system configuration files.
# These define System level configuration variables.
#--------------------------------------------------------------------
%include $(_USTN_SYSTEM)*.cfg level System
```


Walkthrough

```
#------------------------------------------------------------------------
# Include the delivered application configuration files.
# These define Application level configuration variables.
#------------------------------------------------------------------------
%include $(_USTN_APPL)*.cfg level Application

%level System
#------------------------------------------------------------------------
# The names of the user preference (.upf) and user configuration (.ucf) files,
# which are stored in the $(_USTN_HOMEPREFS) directory, are determined by _USTN_USERNAME.
#------------------------------------------------------------------------
_USTN_USERNAME          : Personal
_USTN_PREFNAMEBASE      : $(_USTN_HOMEPREFS)$(_USTN_USERNAME)

#------------------------------------------------------------------------
# Define the root directory for the Configuration data.
#------------------------------------------------------------------------
_USTN_CONFIGURATION     : ${_USTN_BENTLEYROOT}Configuration/

#------------------------------------------------------------------------
# The _USTN_ORGANIZATION is for configuration files and data used throughout the user organization.
#------------------------------------------------------------------------
_USTN_ORGANIZATION      : $(_USTN_CONFIGURATION)Organization/

#------------------------------------------------------------------------
# Set directory for WorkSpaces. This is the default directory where the workspace .cfg files are kept.
#  It can be changed in the WorkSpaceSetup.cfg file (see below).
#------------------------------------------------------------------------
_USTN_WORKSPACESROOT    : $(_USTN_CONFIGURATION)WorkSpaces/
```


Bentley Sustaining Infrastructure

Walkthrough

```
#------------------------------------------------------------------------
# Set default locations for WorkSpace Standards and WorkSet.
#------------------------------------------------------------------------
_USTN_WORKSPACEROOT          : $(_USTN_WORKSPACESROOT)$(_USTN_WORKSPACENAME)/
_USTN_WORKSPACESTANDARDS     : $(_USTN_WORKSPACEROOT)Standards/

#------------------------------------------------------------------------
# Set variables for WorkSpace WorkSet directories.
# By default, _USTN_WORKSETSROOT for a given WorkSpace is a subdirectory of $(_USTN_WORKSPACESROOT)
# with a name equal to $(_USTN_WORKSPACENAME). _USTN_WORKSPACENAME is set in the WorkSpace .cfg file.
# The definition of $(_USTN_WORKSETSROOT) can be overriden in individual WorkSpace .cfg files.
#
# The root directory for all WorkSets for a particular WorkSpace can be changed by
# overriding _USTN_WORKSETSROOT in the WorkSpace .cfg file.
#
# The root directory for a particular WorkSet can be be changed by overriding _USTN_WORKSETROOT in
individual WorkSet .cfg files.
#------------------------------------------------------------------------
_USTN_WORKSETSROOT       : $(_USTN_WORKSPACEROOT)WorkSets/

_USTN_WORKSETROOT        : $(_USTN_WORKSETSROOT)$(_USTN_WORKSETNAME)/
_USTN_WORKSETSTANDARDS   : $(_USTN_WORKSETROOT)Standards/

_USTN_WORKSETDATA        : $(_USTN_WORKSETROOT)
_USTN_OUT                : $(_USTN_WORKSETROOT)out/

#------------------------------------------------------------------------
# Set variables to identify WorkSpace template and WorkSet template.
#------------------------------------------------------------------------
_USTN_WORKSPACETEMPLATE : $(_USTN_WORKSPACESROOT)Template/WorkSpace.Template
_USTN_WORKSETTEMPLATE   : $(_USTN_WORKSPACESROOT)Template/WorkSet.Template
```


Walkthrough

```
#-------------------------------------------------------------------------
# The WorkSpaceSetup.cfg file, which is in the $(_USTN_WORKSPACESROOT) directory,
# is a user-editable configuration file that controls how the user interface
# labels the concept of a selectable "WorkSpace".
#
# A WorkSpace is used to group WorkSets. There can be multiple WorkSpaces,
# each which is represented by a configuration file in the $(_USTN_WORKSPACESROOT)
# directory.
#
# MicroStation's UI allows the user to select a WorkSpace. The configuration
# variable _USTN_WORKSPACELABEL defines the label that the user sees in the UI.
# Some users will want to group WorkSets by Client. Others may group WorkSets
# by Department, Asset, Owner, or Contract.
#
# Note: For administrators familiar with MicroStation V8i configuration files,
#  WorkSpace configuration files are similar in concept to the User ".ucf" files,
#  except they don't have the side effects of User configuration files (setting
#  a different user preference file, for example).
#
#-------------------------------------------------------------------------
%level System
%if exists ($(_USTN_CONFIGURATION)WorkSpaceSetup.cfg)
%  include $(_USTN_CONFIGURATION)WorkSpaceSetup.cfg
%endif
```


Walkthrough

```
#----------------------------------------------------------------------
# Include the Organization specific configuration files.
# The configuration files in the _USTN_ORGANIZATION directory are intended to
# set configuration variables that point to organization-wide standards
# such as level libraries, cell libraries, etc. Those settings can be
# augmented or overridden at the WorkSpace or WorkSet level.
#----------------------------------------------------------------------
%if exists ($(_USTN_ORGANIZATION)*.cfg)
%  include $(_USTN_ORGANIZATION)*.cfg level Organization
%endif

#----------------------------------------------------------------------
# Include the personal.ucf configuration file. _USTN_USERCFG is predefined
# It is included here because it it may contain definitions for _USTN_WORKSPACENAME
#  and _USTN_WORKSETNAME. We need those before we try to include the
#  WorkSpace and WorkSet configuration files.
#----------------------------------------------------------------------
%if exists ($(_USTN_USERCFG))
%  include $(_USTN_USERCFG) level User
%else
%  error Exiting, $(_USTN_USERCFG) not found
%endif
```


Walkthrough

```
%if defined (_USTN_WORKSPACENAME)
%  if exists ($(_USTN_WORKSPACESROOT)$(_USTN_WORKSPACENAME).cfg)
     _USTN_WORKSPACECFG = $(_USTN_WORKSPACESROOT)$(_USTN_WORKSPACENAME).cfg
%    include $(_USTN_WORKSPACECFG) level WorkSpace
%  endif
#--------------------------------------------------------------------------
# When we get to this point, we have a WorkSpace defined.
# There may be .cfg files within the WorkSpace. Process those here.
#--------------------------------------------------------------------------
%  if exists ($(_USTN_WORKSPACEROOT)*.cfg)
%    include $(_USTN_WORKSPACEROOT)*.cfg level WorkSpace
%  endif
%endif



%level System
%if defined (_USTN_WORKSETNAME)
%  if exists ($(_USTN_WORKSETSROOT)$(_USTN_WORKSETNAME).cfg)
     _USTN_WORKSETCFG = $(_USTN_WORKSETSROOT)$(_USTN_WORKSETNAME).cfg
%    include $(_USTN_WORKSETCFG) level WorkSet
%  endif

#--------------------------------------------------------------------------
# When we get to this point, we have a WorkSet defined.
# There may be .cfg files within the WorkSet. Process those here.
#--------------------------------------------------------------------------
%  if exists ($(_USTN_WORKSETROOT)*.cfg)
%    include $(_USTN_WORKSETROOT)*.cfg level WorkSet
%  endif
%endif
```


Walkthrough

- ```
  #-----------------------------------------------------------------------
  # If it is defined at any of the preceding levels, include $(_USTN_ROLECFG)
  #-----------------------------------------------------------------------
  %if defined (_USTN_ROLECFG)
  %  include $(_USTN_ROLECFG) level Role
  %endif
  ```


  ```

# How can users customize their Configuration?

## System Level

- There is never a need to change any of the configuration files delivered with MicroStation. They are delivered in the MicroStation/config program directory to make that clear.
- WorkSpaceSetup.cfg is the only .cfg file delivered in the Configuration root directory. It is the first user "touchpoint":
  - Change the label of WorkSpace to Client, Department, Facility, whatever the user prefers.
  - Can redirect _USTN_ORGANIZATION to point to a network directory.
  - Can redirect _USTN_WORKSPACESROOT to point to a network directory.

# How can users customize their Configuration?

## Organization Level

- Every .cfg file located in _USTN_ORGANIZATION is processed.

# How can users customize their Configuration?

## WorkSpace Level

- Each WorkSpace must have a WorkSpace .cfg file in the _USTN_WORKSPACESROOT directory (<workspacename>.cfg)
- That WorkSpace .cfg file can be used to:
  - Redirect the entire WorkSpace to a network directory by changing _USTN_WORKSPACEROOT.
  - Redirect the WorkSpace standards to a network directory by changing _USTN_WORKSPACESTANDARDS
  - Redirect the WorkSet .cfg files to a network directory by changing _USTN_WORKSETSROOT.
- Additional .cfg files can be put into the _USTN_WORKSPACEROOT directory. They are all processed at the WorkSpace level.

# How can users customize their Configuration?

## WorkSet Level

- Each WorkSet must have a WorkSet .cfg file in the _USTN_WORKSETSROOT directory of its WorkSpace (<worksetname>.cfg)
- That WorkSet .cfg file can be used to:
  - Redirect the entire WorkSet (both standards and data) to a network directory by changing _USTN_WORKSETROOT.
  - Redirect the WorkSet standards to a network directory by changing _USTN_WORKSETSTANDARDS
  - Redirect the WorkSet DGN, DWG and other files to a network directory by changing _USTN_WORKSETDATA.
- Additional .cfg files can be put into the _USTN_WORKSETROOT directory. They are all processed at the WorkSet level.

# How can users customize their Configuration?

## Role Level

- If _USTN_ROLECFG is defined at any level, the file it points to is processed.

# How can users customize their Configuration?

## User Level

- The user is not expected to edit the _USTN_USERCFG file as a text file. It is located in the users _USTN_HOMEPREFS directory and called Personal.ucf. There is no longer a "User" selection in the GUI. Changes made in the Configuration dialog are stored in Personal.ucf.

# Migrating existing Project files

- When a .pcf file (Project ConfigurationFile) from V8i is found in _USTN_WORKSETSROOT, MicroStation automatically modifies its _USTN_PROJECTxxx configuration variables to _USTN_WORKSETxxx, and writes the translated .cfg files to the same directory.
- Other .cfg files that build Configuration Variables from _USTN_PROJECTxxx Configuration Variables will have to be manually edited.
- There is no backwards migration path (CONNECT to V8i), but it is possible to set up Configuration Files in V8i and CONNECT that allow data files (DGN and DWG files) to be accessed from both versions. But that is not recommended.

# Other Configuration-related changes

- microstation –debug now defaults to creating and opening a text file in Notepad that shows how Configuration Files were processed and the resulting definitions (similar to microstation –debugopenfile in V8i).
- Command "Show Configuration" now shows the current values of all configuration variables in a text file opened with Notepad