

**OpenSite Designer
OpenRoads Designer
OpenRail Designer
Workspace Setup Guide**

Document Revision #2

Revision History

#	Date	By	Description	Approved
0	4/4/2018	Dan Ahern	Original document	
1	4/26/2019	Dan Ahern	Added ProjectWise Managed Workspace	
2	6/11/2019	Dan Ahern	Correction to step 2 in both Method 1 and Method 2.	

Contents

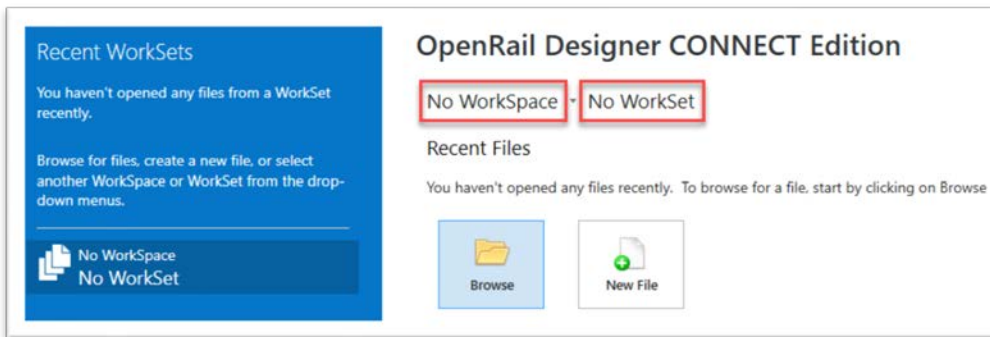
Configuring OpenRoads Designer on a Server	4
The Parts of the Workspace Configuration.....	5
Network Server Setup.....	6
Storing Workspace on Server - Method 1.....	6
Storing Workspace on Server - Method 2.....	9
ProjectWise Managed Workspace Setup.....	11
Setup Workspace Files and Folders on ProjectWise.....	11
Setup Project Space	12
Setup ProjectWise Managed Workspace Configuration Blocks	12
Setup Individual Projects	15

Configuring OpenRoads Designer on a Server

There are different methods to store a workspace configuration on a server, other local folder, or in a ProjectWise Managed environment. This document describes two recommended methods to setup OpenRoads Designer on a local or server folder and the recommended method to setup a ProjectWise Managed environment.

The same workspace configuration setup described herein for OpenRoads Designer can be applied to and used by other products such as MicroStation and OpenBridge Designer.

First, let's get clear a confusing bit of vocabulary because unfortunately the word "workspace" gets used to mean two different things. The folder structure where standards are defined and the configuration variable definitions as a collection are commonly referred to as a workspace. In this document, we use the term *workspace configuration* to describe this folder structure and configuration. New to the CONNECT Edition software is the ability to select a WorkSpace when opening a file.



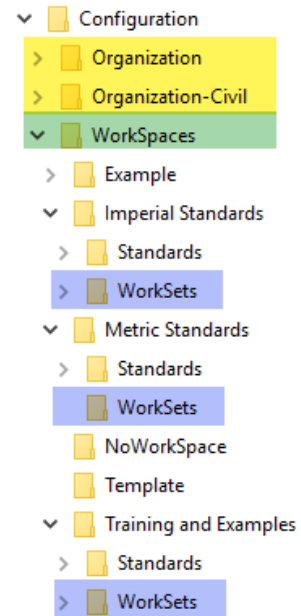
Notice the capitalization of the W and S when we refer to the selectable WorkSpace which is a component of the workspace configuration. This WorkSpace can optionally contain additional configuration files and is the parent to WorkSets which are essentially Project definitions. A separate WorkSet is required for each project.

OpenRoads Designer contains the MicroStation workspace configuration so there is no need to combine them, just point both MicroStation and OpenRoads designer to the OpenRoads Designer workspace configuration. Different WorkSpaces (selectable in the user interface) for MicroStation and OpenRoads Designer can still be used if desired even though they are all stored on a single workspace configuration (folder structure) on the server.

The Parts of the Workspace Configuration

The workspace configuration is made up of three parts that load in successive order. Those parts are the *Organization*, the *WorkSpace*, and the *WorkSet*. The software loads the definitions and resource files from the Organization level, then the WorkSpace, and then the WorkSet. This means the Organization is essentially the base resources and then additional resources specific to a WorkSpace or WorkSet (project) are layered on.

Organization vs Organization-Civil – The standard (MicroStation) workspace configuration includes a folder structure named Organization. The resources added to this folder structure are always loaded no matter which WorkSpace is selected. This is good if there is only one base standard used. However, when work is done for several different countries, or owners, or departments of transportation each generally have different base standards they require be used. This is the purpose of the Organization-Civil folder. In this folder are sub-folders that each contain **a different base standard**. You can have as many base standards as necessary for your needs. For example, if work is done with 4 DOTs there is a folder for each DOT. Or if work is done in 2 different countries there would be a different folder for each country's base or localized standards. Which base standard is used is specified by the variable CIVIL_ORGANIZATION_NAME which is defined in each WorkSpace .cfg file. This allows different WorkSpaces to load different base standards as necessary instead of always loading the same base standard from the Organization folder for all WorkSpaces.



The software is delivered with all three parts of the workspace configuration stored in the Configuration folder. The *Organization* (including Organization-Civil when OpenRoads Designer is loaded) and the *WorkSpace* have their own folders at the root of the Configuration folder. The *WorkSet* is a child of the WorkSpace and is stored in a subfolder within each WorkSpace.

Network Server Setup

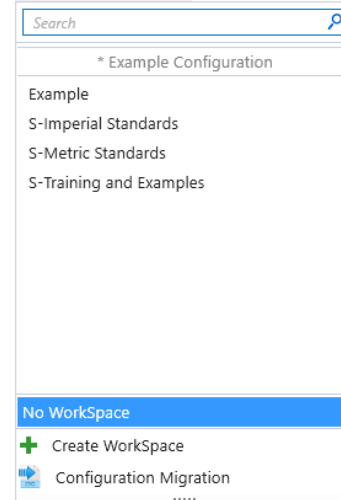
Storing Workspace on Server - Method 1

Using this method MicroStation and OpenRoads Designer only use the WorkSpaces on the server and present only those in the user interface Example Configuration section. The WorkSpaces delivered in the ProgramData folder are not shown in the user interface. Method 2 described below results in WorkSpaces stored on both the server and the local machine to be displayed.

1. Copy the complete
C:\ProgramData\Bentley\OpenRoads Designer CE\Configuration folder to a server location.
2. Delete the ConfigurationSetup.cfg file from the Configuration folder on the server. These should only exist on the local machine.
3. Point OpenRoads Designer to the server folder by defining the necessary variables in the local .cfg file. This may not be an appropriate choice in larger organizations where changes to individual machines is undesirable. Larger organizations should consider setting the variables using environment variables as described the “Using an Environment Variable” section below.
 - a. Edit the file *C:\ProgramData\Bentley\OpenRoads Designer CE\Configuration\WorkspaceSetup.cfg*

OpenRoads Designer CONNECT Edition

No Workspace ▾ No WorkSet



- b. Define the location of the **Organization-Civil** folder by uncommenting and setting the MY_CIVIL_ORGANIZATION_ROOT variable.
For example:
MY_CIVIL_ORGANIZATION_ROOT = //Server_Name/Configuration/Organization-Civil/
- c. Define the location of the **WorkSpaces** folder by uncommenting and setting the MY_WORKSPACES_LOCATION variable. The WorkSpaces folder does NOT need to be a child of the Configuration folder. It can be in a different folder or even a different server.
For example:
MY_WORKSPACES_LOCATION = // Server_Name/WorkSpaces/
- d. OPTIONAL – While not recommended, if you choose to define standards in the Organization folder in addition to the Organization-Civil folders, the location of the **Organization** folder must be defined by adding the _USTN_ORGANIZATION variable to the WorkspaceSetup.cfg file. This variable is not defined in the WorkspaceSetup.cfg file by default.
4. Point other applications (MicroStation, OpenRail Designer, etc.) to the Workspace Configuration.
 - a. Copy the WorkspaceSetup.cfg from the OpenRoads Designer folder and put it in the appropriate products folder. For example, the MicroStation folder is
C:\ProgramData\Bentley\MicroStation CONNECT Edition\Configuration\.

Using an Environment Variable – An alternative method for steps 3-4 is to define the variables as Windows environment variables instead of editing the WorkspaceSetup.cfg file for each product installed on a machine. The advantage is the variables are defined once for all products instead of having to edit and copy individual .cfg files. Windows environment variables can also be defined on a server for each user and then no changes need to be made on the local machines.

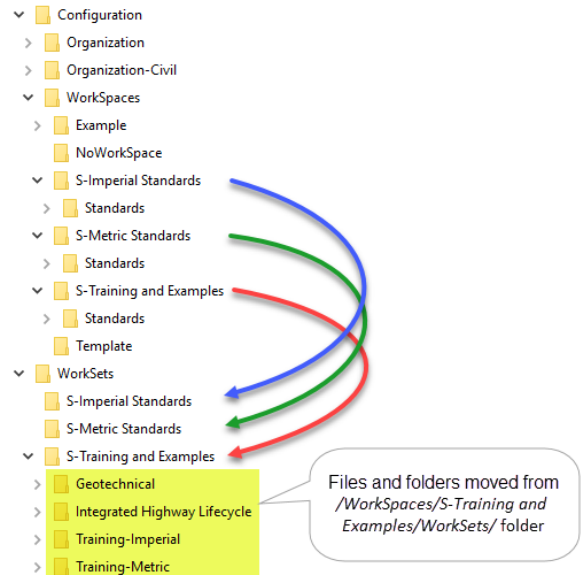
1. Start the Window Control Panel.
2. Select System > Advanced > Environment Variables.
3. Define new variable with the same name as those described above as necessary for your configuration. Be sure to include the trailing slash which is required by MicroStation when defining folder paths.
 1. MY_CIVIL_ORGANIZATION_ROOT
 2. MY_WORKSPACES_LOCATION
 3. *(Optional)* _USTN_ORGANIZATION
 4. *(Optional)* MY_WORKSET_LOCATION (see step 5 below)

5. OPTIONAL – It is also possible to store WorkSet (project) folders at a different location than the WorkSpace. **CAUTION** - Make sure this is really what is desired before implementing this option because it may cause additional administrative effort each time a new WorkSpace or WorkSet is created. This optional setup should **only be considered** if you have a requirement to isolate the project files (the WorkSet) from the WorkSpace.

- Create a new folder on the server for the WorkSets.
- Create a new folder in the WorkSets folder for EACH WorkSpace. The folders must be the same name as the WorkSpace.
- MOVE the existing files and folders from the /WorkSpaces/{WorkSpace name}/WorkSets/ folder to the new /WorkSets/{WorkSpace name}/ folders created in step 3b. Do not copy the WorkSets folder itself, just the contents (files and sub folders).
- Define location of the **WorkSets** folder on the server by setting the MY_WORKSET_LOCATION variable either with an environment variable or by editing the WorkspaceSetup.cfg file.

For example:

```
MY_WORKSET_LOCATION = // Server_Name/WorkSets/
```



There is an alternative setup where WorkSets are not stored by WorkSpace. They are still children of a WorkSpace, but are not organized that way in the folder structure. To make this work you will need to make a change to the WorkspaceSetup.cfg file. Uncomment the `_USTN_WORKSETSROOT=$(MY_WORKSET_LOCATION)/` line and comment out the line above it. When using this setup, skip step 5b in the above process. The disadvantage to this setup is that all WorkSets appear no matter which WorkSpace is selected. However, since the proper WorkSpace and WorkSet are branded into a file, they will correctly be opened even if a file from a non active WorkSpace is selected.

```
%if defined (MY_WORKSET_LOCATION) && exists ($(MY_WORKSET_LOCATION))
  _USTN_WORKSETSROOT = $(MY_WORKSET_LOCATION)$( _USTN_WORKSPACENAME) /
# _USTN_WORKSETSROOT = $(MY_WORKSET_LOCATION) /
%else
  _USTN_WORKSETSROOT = $( _USTN_WORKSPACEROOT)WorkSets/
%endif
```

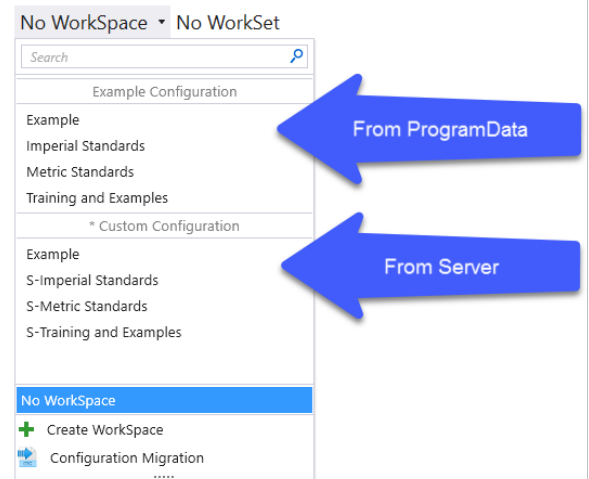

Storing Workspace on Server - Method 2

NOTE: While this method is the most straight forward because you can redirect everything with a single variable, it limits your ability to store workspace configuration files and project files in separate server locations.

Also, using this method BOTH the WorkSpaces on the local machine and the server location are displayed in the interface. These are presented in the user interface as the *Example Configuration* and *Custom Configuration* respectively.

1. Copy the complete
`C:\ProgramData\Bentley\OpenRoads Designer
 CE\Configuration\` folder to a server location.
2. Delete ConfigurationSetup.cfg from the Configuration folder on the server. These should only exist on the local machine.
3. Point OpenRoads Designer to the server folder by defining the necessary variables in the local .cfg file. This may not be an appropriate choice in larger organizations where changes to individual machines is undesirable. Larger organizations should consider setting the variables using environment variables as described the “Using an Environment Variable” section below.
 - a. Edit the file `C:\ProgramData\Bentley\OpenRoads Designer
 CE\Configuration\ConfigurationSetup.cfg`
 - b. Define server location of the **Configuration** folder by uncommenting and setting the variable.
 For example:
`_USTN_CUSTOM_CONFIGURATION = //Server_Name/Configuration/`
4. Point other applications (MicroStation, OpenRail Designer, etc.) to the Workspace Configuration.
 - a. Copy the WorkspaceSetup.cfg from the OpenRoads Designer folder and put it in the appropriate products folder. For example, the MicroStation folder is
`C:\ProgramData\Bentley\MicroStation CONNECT Edition\Configuration\`.

OpenRoads Designer CONNECT Edition



Using an Environment Variable – An alternative method for steps 3-4 is to define the variable as Windows environment variable instead of editing the WorkspaceSetup.cfg file for each product installed on a machine. The advantage is the variable is defined once for all products instead of having to edit and copy individual .cfg file. Windows environment variables can also be set on a server for each user and then no changes need to be made on the local machines.

1. Start the Window Control Panel.
2. Select System > Advanced > Environment Variables.
3. Define a new variable named `_USTN_CUSTOM_CONFIGURATION` with a value set to the location of the **Configuration** folder on the server or alternate local folder. Be sure to include the trailing slash which is required by MicroStation when defining folder paths.

4. Inspect the *ConfigurationSetup.cfg* file for each product and make sure there are no uncommented lines that read `_USTN_CUSTOM_CONFIGURATION=`. If one exists, add a comment (`#` symbol) to the beginning of the line. If you do not do this MicroStation or OpenRoads Designer will read this line and overwrite the Windows environment variable definition with blank.

```
1  #-----
2  # ConfigurationSetup.cfg - Configures the root Configuration directory
3  # for Your Organization
4  #
5  # The main function of this configuration file is to allow user to specify
6  # the root Configuration directory to activate. The active Configuration
7  # directory is represented by _USTN_CONFIGURATION. By default, it points to
8  # the installed Configuration defined by _USTN_INSTALLED_CONFIGURATION,
9  # which can consist of example WorkSpaces and WorkSets.
10 # If your organization has its own Configuration directory, you can define
11 # _USTN_CUSTOM_CONFIGURATION to that directory path and use it as follows.
12 #
13 # _USTN_CUSTOM_CONFIGURATION = D:/.../MyConfiguration/
14 # _USTN_CONFIGURATION : ${_USTN_CUSTOM_CONFIGURATION}
15 #-----
16 #
17 #-----
18 # START: The section defines user selection at the time of installation.
19 # These lines are generated by installer.
20 [General]
21 _USTN_CUSTOM_CONFIGURATION=
22
23 [SetConfiguration]
24 %if !defined (_USTN_USER_CONFIGURATION)
25 %if defined (_USTN_CUSTOM_CONFIGURATION) && (${_USTN_CUSTOM_CONFIGURATION} != "") && exists (${_USTN_CUSTOM_CONFIGURATION})
26 _USTN_CONFIGURATION = C:/ProgramData/Bentley/MicroStation CONNECT Edition/Configuration/
27 %endif
28 %endif
```

ProjectWise Managed Workspace Setup

This is a brief overview of setting up OpenRoads Designer in a ProjectWise Managed Workspace environment.

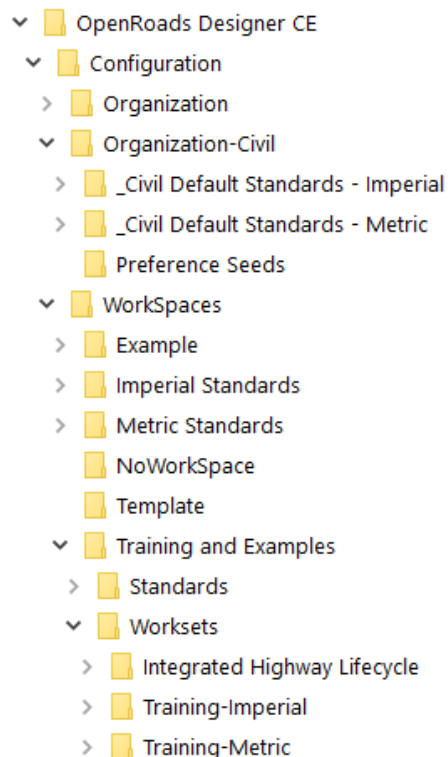
Setup Workspace Files and Folders on ProjectWise

1. Setup and test the workspace configuration outside of ProjectWise first. This will make it easier to do the setup and troubleshoot problems before introducing the ProjectWise Managed Workspace environment.
2. Copy the entire workspace configuration folder structure, including the .cfg files, into ProjectWise. The location can be anywhere in your ProjectWise folder structure.

- The Organization-Civil area contains your standards and should typically be read only to users.
- By default, the WorkSpaces area includes WorkSets or project settings. The easiest setup is to keep this native folder structure. However, it is a common practice to locate WorkSets or projects in a different ProjectWise folder. This may be desirable to match existing folder structure and/or to allow read/write access to the WorkSet or project folders.

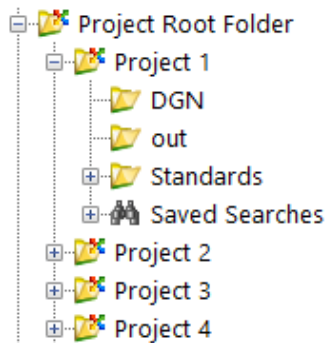
NOTE: At this time all WorkSet .cfg and .dgnws files must be in a single folder. It is not possible to locate individual WorkSet .cfg and .dgnws files in individual project folder structures. This capability is planned to be added in a future release.



- Below is an example of the workspace configuration folder structure as it exists in the delivered software.



Setup Project Space

Create a folder in ProjectWise where all the projects will be stored. Projects should be organized into individual folders inside a single container folder. In the following example the folder named “Project Root Folder” is the container. This container folder can be named anything that fits your needs. All project data is in the respective Project folders which again can be named anything that meets your needs. The Project folders can include any subfolders needed. The subfolders shown here are just an example, not a requirement.



For the Managed Workspace to function as described in this document, the Project Root Folder and the Project folders must be ProjectWise Work Areas (indicated by the ) icon), not standard folders. All other folders inside a project should be standard folders (indicated by the ) icon).

Setup ProjectWise Managed Workspace Configuration Blocks

Configuration Blocks define variables that are evaluated when a file is opened in ProjectWise. These variables define where the rest of the workspace configuration is in ProjectWise.

Variables defined in Configuration Blocks take longer to evaluate than variables defined in .cfg files. Therefore, the number of variable definitions in Configuration Blocks should be kept to a minimum with most of the variables defined in .cfg files just like a workspace configuration outside of ProjectWise. When working with a Managed Workspace, these .cfg files are stored in ProjectWise as defined in the previous section of this document.

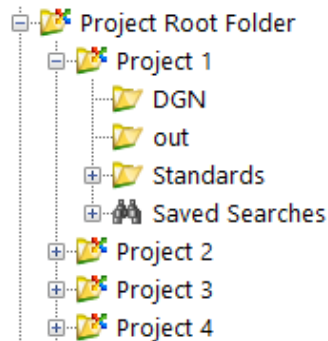
The configuration blocks described below are a recommended way to setup a typical managed workspace configuration, but they are not the only way. You may need to adapt exactly which configuration blocks are created and what variables they define to meet the needs of your environment. The general guidance is to keep your configuration blocks as simple as possible.

1. Start the ProjectWise Administrator and log into the Datasource.
2. Define a *Predefined* Configuration Block that defines where the overall configuration and WorkSpaces are located.
 - a. Browse to the *WorkSpaces > Managed > Predefined* area.
 - b. Create a new *Managed Workspace Configuration Block* in the **Predefined** area.
 - c. Open the *Configuration Block Properties* and select the **Configuration** tab.
 - d. Define where the configuration is stored in ProjectWise.
 - i. Select **+ Add Variable** to add a new variable definition.

- ii. Set *Name* to ***_USTN_CONFIGURATION***
 - iii. Enable the **Locked** check box.
 - iv. Select **Add**.
 - v. Set *Operation Type* to **'=' – Assignment**
 - vi. Set *Value Type* to **ProjectWise Folder**
 - vii. Click on the selector next to the Value field and browse to the top-level folder where the workspace structure is stored in ProjectWise. In the above image that is the **OpenRoads Designer CE** folder.
 - viii. Select **OK** to accept and close the Edit Value dialog.
 - ix. Select **OK** to accept and close the New Variable dialog.
 - e. If the WorkSpaces folder is in the default location under the configuration folder setup in step 2d, skip to step 3. Otherwise the location of the WorkSpaces folder must be defined.
 - i. Select **+ Add Variable** to add a new variable definition.
 - ii. Set *Name* to ***_USTN_WORKSPACESROOT***
 - iii. Enable the **Locked** check box.
 - iv. Select **Add**.
 - v. Set *Operation Type* to **'=' – Assignment**
 - vi. Set *Value Type* to **ProjectWise Folder**
 - vii. Click on the selector next to the Value field and browse to the top-level folder where the WorkSpaces are stored in ProjectWise. In the above image that is the **OpenRoads Designer CE\WorkSpaces** folder.
 - viii. Select **OK** to accept and close the Edit Value dialog.
 - ix. Select **OK** to accept and close the New Variable dialog.
 - f. Select **OK** to accept and close the Configuration Block dialog.
3. Define a *Workspace* Configuration Block that defines the Workspace Name and WorkSet variables.
- a. Browse to the *WorkSpaces > Managed > Workspace* area.
 - b. Create a new Managed Workspace Configuration Block in the Workspace area.
 - c. Open the *Configuration Block Properties* and select the **Configuration** tab.
 - d. Define the name of the Workspace .cfg to be loaded.
 - i. Select **+ Add Variable** to add a new variable definition.
 - ii. Set *Name* to ***_USTN_WORKSPACENAME***
 - iii. Enable the **Locked** check box.
 - iv. Select **Add**.
 - v. Set *Operation Type* to **'=' – Assignment**
 - vi. Set *Value Type* to **String**
 - vii. Set *Value* to the name of the Workspace. For example, **Imperial Standards**.
 - viii. Select **OK** to accept and close the Edit Value dialog.
 - ix. Select **OK** to accept and close the New Variable dialog.

- e. Define the location of the WorkSet .cfg files.

These files will be in the parent folder to the project folders. For example, in the following image the WorkSet .cfg files are in the folder named “Project Root Folder”.



- i. Select **+ Add Variable** to add a new variable definition.
- ii. Set *Name* to **_USTN_WORKSETROOT**
- iii. Enable the **Locked** check box.
- iv. Select **Add**.
- v. Set *Operation Type* to **'=' – Assignment**
- vi. Set *Value Type* to **String**
- vii. Set *Value* to **\$(dms_parentproject (_dgndir))**

TIP: When running in ProjectWise and using Work Areas, the `dms_parentproject` variable points to the Work Area two levels above the active .dgn file. Consider our example folder structure where a .dgn file can be in any folder within a project. The first Work Area above the .dgn file is the Project folder such as Project 1, Project 2, etc. The second Work Area above the .dgn file is the Project Root Folder in our example. This assignment is to set the WorkSet Root folder to the parent of the folder where the projects are located. The folder could also be hard coded into the variable definition but using `$(dms_parentproject (_dgndir))` makes it flexible so that the same Managed Workspace can be used on many different top level folders in ProjectWise.

- viii. Select **OK** to accept and close the Edit Value dialog.
 - ix. Select **OK** to accept and close the New Variable dialog.
- f. Define location of the WorkSet .dgnws files. This must be the same location as the WorkSet .cfg files.
- i. Select **+ Add Variable** to add a new variable definition.
 - ii. Set *Name* to **_USTN_WORKSETDGNWSROOT**
 - iii. Enable the **Locked** check box.
 - iv. Select **Add**.
 - v. Set *Operation Type* to **'=' – Assignment**
 - vi. Set *Value Type* to **String**
 - vii. Set *Value* to **\$(dms_parentproject (_dgndir))**
 - viii. Select **OK** to accept and close the Edit Value dialog.
 - ix. Select **OK** to accept and close the New Variable dialog.

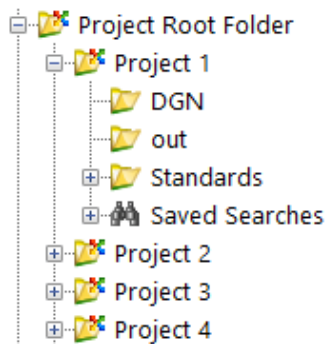
- g. Define the WorkSet name to be the same as the Project folder name.
 - i. Select **+ Add Variable** to add a new variable definition.
 - ii. Set *Name* to **`_USTN_WORKSETNAME`**
 - iii. Enable the **Locked** check box.
 - iv. Select **Add**.
 - v. Set *Operation Type* to **'=' – Assignment**
 - vi. Set *Value Type* to **String**
 - vii. Set *Value* to **`$(lastdirpiece (dms_project (_dgndir)))`**

TIP: The `dms_project (_dgndir)` sting resolves to the first Work Area above the active .dgn or the project folder path. The `lastdirpiece()` function strips the path and returns only the last project folder name itself.

- viii. Select **OK** to accept and close the Edit Value dialog.
- ix. Select **OK** to accept and close the New Variable dialog.

Setup Individual Projects

1. Create new folder for the project and make it a ProjectWise Work Area.



2. Copy the **WorkSet Template.cfg** into the parent folder to the project folder (Project Root Folder in this example image) and name the file the same as the Project folder. For example, Project 1.cfg, Project 2.cfg, etc. in the example.
3. The .dgnws file will automatically be created the first time the WorkSet/Project is used.