

# More about User Defined Hole Standards

You can define [hole standards](#) in an .xml file.

The following topics are discussed:

**Related Topics**  
[Creating Hole Standards](#)

- [File Structure](#)
- [Hole Standard Parameters](#)
- [Location of Hole Standards Files](#)
- [Name of Hole Standard](#)
- [Description of Hole Standard](#)

## File Structure

You can define [hole standards](#) in .xml files with specific structures.

Sample file structure for simple [hole](#)

```
<?xml version="1.0" ?>

<std:node name="S1_standard" xmlns:std="http://www.dsweb.com/std">

<std:node name="Units">
<std:node name="Length">
<std:strval name="Length">mm</std:strval>
</std:node>
<std:node name="Angle">
<std:strval name="Angle">deg</std:strval>
</std:node>
</std:node>

<std:typedef name="SimpleHoleValues">
<std:strval name="Name">S1</std:strval>
<std:floatval name="MainDiameter">5.</std:floatval>
<std:floatval name="MainDepth">10.</std:floatval>
<std:floatval name="BottomAngle">110.</std:floatval>
<std:floatval name="ThreadDiameter">6.</std:floatval>
<std:floatval name="ThreadDepth">8.</std:floatval>
<std:floatval name="Pitch">1.</std:floatval>
</std:typedef>

<std:node name="SimpleHoleValues">
<std:typeval name="SimpleHoleValues">
<std:strval name="Name">S1</std:strval>
<std:floatval name="MainDiameter">5.</std:floatval>
<std:floatval name="MainDepth">10.</std:floatval>
<std:floatval name="BottomAngle">105.</std:floatval>
<std:floatval name="ThreadDiameter">6.</std:floatval>
<std:floatval name="ThreadDepth">8.</std:floatval>
<std:floatval name="Pitch">1.</std:floatval>
</std:typeval>
</std:node>
</std:node>
```

Sample file structure for tapered [hole](#)

```
<?xml version="1.0" ?>

<std:node name="T1_standard" xmlns:std="http://www.dsweb.com/std">

<std:node name="Units">
<std:node name="Length">
<std:strval name="Length">mm</std:strval>
</std:node>
<std:node name="Angle">
<std:strval name="Angle">deg</std:strval>
</std:node>
</std:node>

<std:typedef name="TaperedHoleValues">
<std:strval name="Name">T1</std:strval>
<std:floatval name="MainDiameter">5.0</std:floatval>
<std:floatval name="MainDepth">10.</std:floatval>
<std:floatval name="TaperedAngle">90.0</std:floatval>
<std:intval name="AnchorPoint">0</std:intval>
<std:floatval name="BottomAngle">110.0</std:floatval>
<std:floatval name="ThreadDiameter">6.0</std:floatval>
```

```

<std:floatval name="ThreadDepth">8.</std:floatval>
<std:floatval name="Pitch">1.0</std:floatval>
</std:typedef>

<std:node name="TaperedHoleValues">
<std:typeval name="TaperedHoleValues">
<std:strval name="Name">T1</std:strval>
<std:floatval name="MainDepth">10.</std:floatval>
<std:floatval name="MainDiameter">5.0</std:floatval>
<std:floatval name="TaperedAngle">90.0</std:floatval>
<std:intval name="AnchorPoint">0</std:intval>
<std:floatval name="BottomAngle">100.0</std:floatval>
<std:floatval name="ThreadDiameter">6.0</std:floatval>
<std:floatval name="ThreadDepth">8.</std:floatval>
<std:floatval name="Pitch">1.0</std:floatval>
</std:typeval>

</std:node>

</std:node>

```

### Sample file structure for counterbored hole

```

<?xml version="1.0" ?>

<std:node name="CB1_standard" xmlns:std="http://www.dsweb.com/std">

<std:node name="Units">
<std:node name="Length">
<std:strval name="Length">mm</std:strval>
</std:node>
<std:node name="Angle">
<std:strval name="Angle">deg</std:strval>
</std:node>
</std:node>

<std:typedef name="CounterBoredHoleValues">
<std:strval name="Name">CB1</std:strval>
<std:floatval name="MainDiameter">5.0</std:floatval>
<std:floatval name="CounterBoredDiameter">10.0</std:floatval>
<std:floatval name="CounterBoredDepth">5.0</std:floatval>
<std:floatval name="BottomAngle">120.0</std:floatval>
<std:floatval name="ThreadDiameter">6.0</std:floatval>
<std:floatval name="Pitch">1.0</std:floatval>
</std:typedef>

<std:node name="CounterBoredHoleValues">
<std:typeval name="CounterBoredHoleValues">
<std:strval name="Name">CB1</std:strval>
<std:floatval name="MainDiameter">5.0</std:floatval>
<std:floatval name="CounterBoredDiameter">10.0</std:floatval>
<std:floatval name="CounterBoredDepth">5.0</std:floatval>
<std:floatval name="BottomAngle">115.0</std:floatval>
<std:floatval name="ThreadDiameter">6.0</std:floatval>
<std:floatval name="Pitch">1.0</std:floatval>
</std:typeval>

</std:node>

</std:node>

```

### Sample file structure for countersunk hole

```

<?xml version="1.0" ?>

<std:node name="CS1_standard" xmlns:std="http://www.dsweb.com/std">

<std:node name="Units">
<std:node name="Length">
<std:strval name="Length">mm</std:strval>
</std:node>
<std:node name="Angle">
<std:strval name="Angle">deg</std:strval>
</std:node>
</std:node>

<std:typedef name="CounterSunkHoleValues">
<std:strval name="Name">CS1</std:strval>
<std:floatval name="MainDiameter">5.0</std:floatval>
<std:intval name="Mode">0</std:intval>

```

```

<std:floatval name="CounterSunkDepth">5.0</std:floatval>
<std:floatval name="CounterSunkAngle">90.0</std:floatval>
<std:floatval name="CounterSunkDiameter">0.0</std:floatval>
<std:floatval name="BottomAngle">120.0</std:floatval>
<std:floatval name="ThreadDiameter">6.0</std:floatval>
<std:floatval name="Pitch">1.0</std:floatval>
</std:typedef>

```

```

<std:node name="CounterSunkHoleValues">
<std:typeval name="CounterSunkHoleValues">
<std:strval name="Name">CS1</std:strval>
<std:floatval name="MainDiameter">5.0</std:floatval>
<std:intval name="Mode">0</std:intval>
<std:floatval name="CounterSunkDepth">5.0</std:floatval>
<std:floatval name="CounterSunkAngle">90.0</std:floatval>
<std:floatval name="CounterSunkDiameter">0.0</std:floatval>
<std:floatval name="BottomAngle">120.0</std:floatval>
<std:floatval name="ThreadDiameter">6.0</std:floatval>
<std:floatval name="Pitch">1.0</std:floatval>
</std:typeval>

```

```
</std:node>
```

```
</std:node>
```

### Sample file structure for counterdrilled hole

```

<?xml version="1.0" ?>
<std:node name="CD1_standard" xmlns:std="http://www.dsweb.com/std">
<std:node name="Units">
<std:node name="Length">
<std:strval name="Length">mm</std:strval>
</std:node>
<std:node name="Angle">
<std:strval name="Angle">deg</std:strval>
</std:node>
</std:node>
<std:typedef name="CounterDrilledHoleValues">
<std:strval name="Name">CD1</std:strval>
<std:floatval name="MainDiameter">5.0</std:floatval>
<std:floatval name="CounterDrilledDiameter">10.0</std:floatval>
<std:floatval name="CounterDrilledDepth">5.0</std:floatval>
<std:floatval name="CounterDrilledAngle">90.0</std:floatval>
<std:floatval name="BottomAngle">120.0</std:floatval>
<std:floatval name="ThreadDiameter">6.0</std:floatval>
<std:floatval name="Pitch">1.0</std:floatval>
</std:typedef>
<std:node name="CounterDrilledHoleValues">
<std:typeval name="CounterDrilledHoleValues">
<std:strval name="Name">CD1</std:strval>
<std:floatval name="MainDiameter">5.0</std:floatval>
<std:floatval name="CounterDrilledDiameter">10.0</std:floatval>
<std:floatval name="CounterDrilledDepth">5.0</std:floatval>
<std:floatval name="CounterDrilledAngle">90.0</std:floatval>
<std:floatval name="BottomAngle">120.0</std:floatval>
<std:floatval name="ThreadDiameter">6.0</std:floatval>
<std:floatval name="Pitch">1.0</std:floatval>
</std:typeval>
</std:node>
</std:node>

```

## Hole Standard Parameters

Different parameters are available for each [hole](#) type. Some parameters are optional.

### Simple [hole](#)

- Description of the [standard](#): Mandatory
- Main [Hole](#) diameter: Mandatory
- Main [Hole](#) Depth: Optional
- Bottom Angle: Optional
- Thread Diameter: Optional
- Thread Depth: Optional

- Pitch: Optional

#### Tapered [hole](#)

- Description of the [standard](#): Mandatory
- Main [Hole](#) diameter: Mandatory
- Tapered angle: Mandatory
- Anchor Point: Mandatory

Note: If you set the Anchor Point value to 0, the anchor point is placed at the bottom of the [hole](#). If you set the Anchor Point value to 1, the anchor point is placed at the top of the [hole](#).

- Main [Hole](#) Depth: Optional
- Bottom Angle: Optional
- Thread Diameter: Optional
- Thread Depth: Optional
- Pitch: Optional

#### CounterBored [hole](#)

- Description of the [standard](#): Mandatory
- Main [Hole](#) diameter: Mandatory
- CounterBored Diameter: Mandatory
- CounterBored Depth: Mandatory
- Main [Hole](#) Depth: Optional
- Bottom Angle: Optional
- Thread Diameter: Optional
- Thread Depth: Optional
- Pitch: Optional

#### CounterSunk [hole](#)

- Description of the [standard](#): Mandatory
- Main [Hole](#) diameter: Mandatory
- CounterSunk Mode: Mandatory

##### Notes:

- If you set CounterSunk Mode to 0, you can define CounterSunk Depth and CounterSunk Angle.
- If you set CounterSunk Mode to 1, you can define CounterSunk Diameter and CounterSunk Depth.
- If you set CounterSunk Mode to 2, you can define CounterSunk Diameter and CounterSunk Angle.
- CounterSunk Diameter: Mandatory
- CounterSunk Depth: Mandatory
- CounterSunk Angle: Mandatory
- Main [Hole](#) Depth: Optional
- Bottom Angle: Optional
- Thread Diameter: Optional
- Thread Depth
- Pitch: Optional

#### CounterDrilled [hole](#)

- Description of the [standard](#): Mandatory
- Main [Hole](#) diameter: Mandatory
- CounterDrilled Diameter: Mandatory
- CounterDrilled Depth: Mandatory
- CounterDrilled Angle: Mandatory
- Main [Hole](#) Depth: Optional
- Bottom Angle: Optional
- Thread Diameter: Optional
- Thread Depth: Optional
- Pitch: Optional

##### Notes:

- If you do not define thread parameters in the [hole standard](#), the [hole](#) created by using the [hole standard](#) will not be threaded.
- If you do not define bottom angle in the [hole standard](#), the [hole](#) created by using the [hole standard](#) will be flat bottomed.
- The thread diameter and pitch must be set together. The thread depth cannot be set alone.

## Location of Hole Standards Files

You must store [hole standards](#) .xml files in separate directories under the following directory.

```
PartDesignFeature\CNext\resources\standard
```

You must create the following directories under the [standard](#) directory.

- Simple[Hole](#) to store simple [hole](#) files.
- Tapered[Hole](#) to store tapered [hole](#) files.
- CounterSunk[Hole](#) to store countersunk [hole](#).
- CounterDrilled[Hole](#) to store counterdrilled [hole](#).
- CounterBored[Hole](#) to store counterbored [hole](#).

You can also mention norms related to the [hole](#), such as, ISO, ASME, and ANSI, in the .xml file.

Set the environment variable as follows:

```
set CATCollectionStandard = dir1\xxx\yyy
```

## Name of Hole Standard

You can name the [standards](#) file as required. The name of the [standard](#) is the name of the source file.

## Description of Hole Standard

You must provide a name to only one description.

**3DEXPERIENCE** R2014x © 2014 Dassault Systèmes. All rights reserved.

# Creating Hole Standards

You can create new hole standards by using existing standards files.

## Before you begin:

You must have administrator rights to access and modify hole standards.

### Related Topics

Creating Holes  
More about User Defined  
Hole Standards

1. Select Me > Preferences > Standards.

The Standard Definition dialog box appears.

2. In the Category list, select a hole type.
3. In the File list, select a standard defined for the hole.

The following standards files are available under the CounterBoredHole category.

- Metric\_Cap\_Screws.xml
- Socket\_Head\_Cap\_Screws.xml

4. Expand the parent node.

The Units and Values nodes appear.

5. **Optional:** Configure units for the hole standard.

- a. Expand the Units node.
- b. Click the required node and configure the corresponding unit.

6. Configure the values for the hole standard.

- Create an instance.
  - a. Click the Values node and click Create Instance.

The Instance List dialog box appears.

- b. Type the name of the instance in the Name of new instance box.
- c. Select an instance from the Duplicated from list, duplicating which the new instance must be created.
- d. Click OK.

The instance is created. The created instance appears under the Values node.

Notes:

- A warning message appears if the standard unit is different from the unit specified for a model.
- The symbol of the new unit must comply with the symbol of the unit defined in the Units area.

**Tip:** Select Me > Preferences > Parameters and Measure > Units tab to view the Units area.

- Click the required instance and configure the corresponding values.
- Delete an instance.

- a. Click the Values node and click Remove Instance.

The Instance List dialog box appears.

- b. Select an instance to be deleted from the Delete instance list.
- c. Click OK.

The instance is deleted.

7. Click Save As New to save the standards in a new standards file.

**Important:** You can modify a standard on the fly. However, if the standard is already used and the design table is created, then the modifications do not affect the values defined in the Hole Definition and Thread Definition dialog boxes.

## In the Knowledge Base

About updating hole standards

About using hole and thread standards