MechWorks PDM R16

Behavior Changes

Concurrent Engineering: Reload of externally modified components

It is now no longer possible:
- to instantly reload a modified component; the reload notification dialog is just a notification, with no ability to instantly reload the component - in order to reload the modified components, it must be used the [RELOAD] red button in the MechWorks PDM CAD Toolbar
- to automatically reload all modified components for the rest of the current session

DBWAclServer Preferred Server

Starting from R16, the *Option->Environment->DBWServer->DBWAclServer->Preferred Server Name* is mandatory. Whenever the DBWAclServer option is enabled, a check on the assignment of the Preferred Server Name is executed, and when confirming with [OK], a connectivity check is executed

DBWArm: Modifications->[X] Allow 'delete' in Open for Browsing (CHECKED = YES)

Being not possible to edit the rights of DBWArm Administrators, they now have the functionality enabled by default - so every DBWArm Admin can delete the documents with the Browser opened in 'read-only' mode

Recursive Revision Operators on Standard Parts

It is now possible to manage the revisions of Standard Parts using the Recursive Revision functionalities (Checkout/Checkin/Approve Tree).
To avoid any update of Standard Parts, it is still valid the option "*General->[X] No Updates for Standard Parts*"

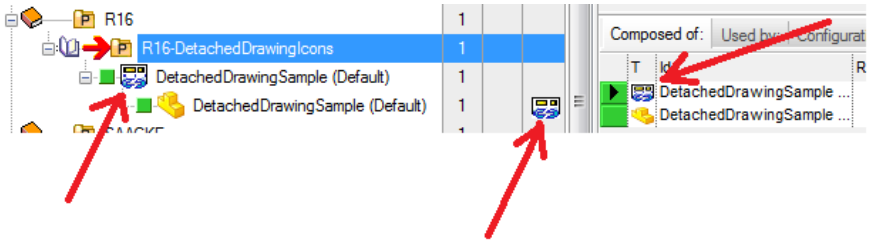
Enhancements

Browser

Detached Drawing Icon

[SolidWorks/DBWorks integration only]

The SolidWorks Detached Drawings are now displayed in the Browser with their specific Icon:

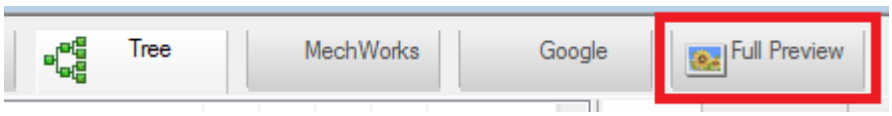


Full Preview Tab: possibility to change its position

The top-level Full Preview Tab of the Browser can be relocated in a different position by declaring it in the shared SCHEMA\DBWIeDef.txt file.
The label used for the *TITLE Full Preview* definition must be exactly the same used for the T_IDC_TAB_FULL_PREVIEW entry in the localized Message File (*.MSG) used.
For example, the following definition:

TITLE MechWorks
URL http://www.mechworks.com
TITLE Google
URL http://www.google.com
TITLE Full Preview
URL about:blank

will show the Full Preview tab at the end of the tabs:



Configurations

New Option: [X] Always ask for confirmation when deleting a non referenced Configuration

[SolidWorks/SolidEdge integrations only]

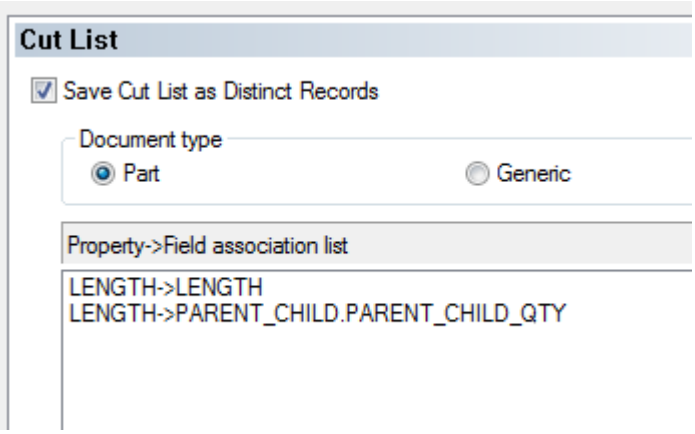
When checked, a prompt is displayed before deleting a Configuration Specific database record, if the configuration is not referenced by any other Parent Document - this option (default: checked) is a safeguard againt involuntary deletions of in-memory configurations, that are undoable in the CAD model, but not in the database.

Cut Lists

Ability to map Cut List Properties into PARENT_CHILD Fields

[SolidWorks/SolidEdge integrations only]

It is now possible to map Cut List Properties into PARENT_CHILD Fields.
The PARENT_CHILD fields must be entered manually, prefixed with the localized name of the PARENT_CHILD table followed by a dot.
Example:
Let's map the LENGTH Property into either the DOCUMENT::LENGHT and the PARENT_CHILD::PARENT_CHILD_QTY fields. The association list will be:



Data Input Form

.DYNAMIC SQL queries in LST files

It is now possible to declare the new clause .DYNAMIC in the header of a .SQL type .LST file that typically contains a macro $(...) referencing a different field, whose content will be used for driving the list of choises in the dropdown list box.
The .DYNAMIC clause forces the requery of the drop-down list box at run time; without it, the $(...) macros are resolved only one time, when the form is displayed

Example:

Let's create a CATEGORY2.LST file so that whenever the CATEGORY1 field will change, its list box will display values related to the just assigned CATEGORY1 field.
The LST file will look as follows:

```
.SQL
.KEY 0
.DYNAMIC
SELECT ID,ShortCode,Description FROM dbo.TblCodeChoices WHERE LinkToCategory=$(CATEGORY1)
```

Notes:

Without the .DYNAMIC clause, the $(CATEGORY1) macro is resolved when the form is displayed - any further change to the CATEGORY1 value will NOT affect the dropdown list box related to CATEGORY2.

With the .DYNAMIC clause, the $(CATEGORY1) macro is resolved each time the dropdown list box, related to CATEGORY2, is displayed for getting a choise, so the current value of CATEGORY1 is used for requering the database.

.LST files: .DEFAULT_IF_VISIBLE

For a specific field, it is now possible to declare a Default Value that will be applied only if the field is actually visible in the current Form (typically a Sub-Classed Form)

The definition is:
.DEFAULT_IF_VISIBLE <any valid .DEFAULT definition>

Note: the standard .DEFAULT (or the Database-View-specific .DEFAULT_A/P/G/D/0) always applies the value to the record created in the Document Table, no matter if the field itself is visible in the currently displayed Data Input Form or not.

DBInventor

Built-in support for Inventor Materials

It is now possible to get the currently assigned material without creating a SCHEMA\CustProp.txt file.
On every recalculation of the Mass Properties, the MATERIAL field will be correctly populated with the current Inventor Material.
The Environment option "[X] Use Inventor Material" has now been enabled.

DBInventor Standalone support for Documents Registration

Using the DBInventor Standalone Client, it is now possible to drag & drop an Assembly / Drawing / Part from Windows Explorer to any Project for registering the documents into the MechWorks PDM Database (the same operation can also be executed using the *RMB->Add document* functionality)
The following limitations apply:
1) the DBInventor Standalone must be installed on the same workstation where Inventor is installed
2) Level of Details / iParts / iAssemblies / Configurations are not supported - they will be registered as normal Assemblies

New option: User Interface->Tree->[X] Hide Inventor Presentations in the Tree

When checked (default: not checked), the .IPN Presentation documents will not be displayed in the Tree

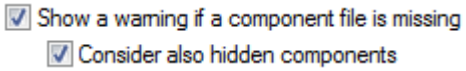
DBSolidEdge

DBSolidEdge Standalone support for Documents Registration

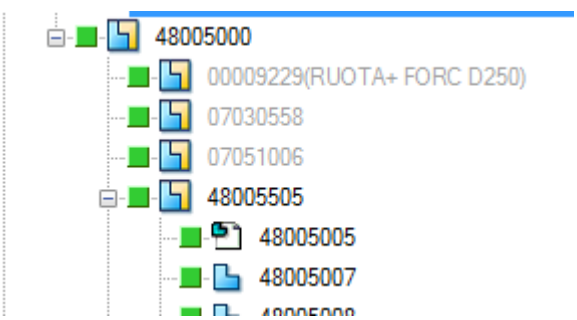
Using the DBSolidEdge Standalone Client, it is now possible to drag & drop an Assembly / Drawing / Part from Windows Explorer to any Project for registering the documents into the MechWorks PDM Database (the same operation can also be executed using the *RMB->Add document* functionality)
The following limitations apply:
1) the DBSolidEdge Standalone must be installed on the same workstation where SolidEdge is installed
2) FOA Assemblies are not supported - they will be registered as normal Assemblies

Support for missing files in Assemblies

The Open/Save->Save options have now the following options enabled:



As a consequence of this enhancement, Assemblies with files not found will be displayed with such components grayed:



DBWArm

Added new rights that can be denied

- *View Data -> [X] Open files with ...*
  When checked, it denies the ability to use the RMB->Open files with ... for a Generic Document

- *Modification -> [X] Delete files*
  When checked, it denies the ability to check the *[X] Delete the files (also revision files)* checkbox in the Document Selector for deleting records, when the Browser is opened in "update" mode

Persistence of USERS Grid Columns Widths

The columns widths assigned in the USERS Grid are now persistent across multiple sessions (the data is stored in the shared USERS\DBWARM path)

DBWERP

Image files lookup into SCHEMA\IMG\ERP

It is now possible to make custom versions of the image files:
SCHEMA\IMG\DBWERPStateInserted.BMP
SCHEMA\IMG\DBWERPStateInsertedBOM.BMP
SCHEMA\IMG\DBWERPStateInsertedOODBOM.BMP
SCHEMA\IMG\DBWERPStateInsertedOOD.BMP
by creating a copy of them under a sub-folder named SCHEMA\IMG**ERP** and modifying them accordingly.
MechWorks PDM looks first into the IMG\ERP sub-folder, then, if the image files are not found there, it will load the image files from the default SCHEMA\IMG location.

Excel BOM

Support for PARENT_CHILD / DOCUMENT field override

Suppose you want to fill the value of the field DESCRIPTION with a specific value in a given Parent Child context. If you define a field in the PARENT_CHILD table named PARENT_CHILD_DESCRIPTION, and its value is not null, then that value will replace and overwrite the value defined in the DOCUMENT table in that specific context. The same holds true for every DOCUMENT field as long as the naming convention PARENT_CHILD_FIELDNAME is respected (Please note that the PARENT_CHILD table name can be localized to every language)
The enhancement applies only to indented BOMs.

Generic Documents

Improved support for Drawing->Part/Assembly Generic Documents structures

When working with parent-child structures created using Generic Documents with the database type T altered using the SCHEMA\DBWGTYPES.TXT definition file (and of course with the option *General->More->[X] Use 'SCHEMA/DBWGTYPES.TXT' generic document's types definition file* being enabled), it is now possible, for Generic Documents marked as T='P' or T='A', to display the parent-drawings-related *Open* and *Open Master Drawing* RMB popup menus.

License Manager

DBWORKS_MAX_IDLE_TIME_BEFORE_FLOATING_LICENSE_RELEASE parameter specialization

The following new parameters has been added to the shared DBWCONN.PAR file for setting the maximum idle time (in minutes) according to the running application:

DBWORKS_MAX_IDLE_TIME_BEFORE_FLOATING_LICENSE_RELEASE_SW
DBWORKS_MAX_IDLE_TIME_BEFORE_FLOATING_LICENSE_RELEASE_SE
DBWORKS_MAX_IDLE_TIME_BEFORE_FLOATING_LICENSE_RELEASE_INV
DBWORKS_MAX_IDLE_TIME_BEFORE_FLOATING_LICENSE_RELEASE_ALONE

Example:
DBWORKS_MAX_IDLE_TIME_BEFORE_FLOATING_LICENSE_RELEASE 15
DBWORKS_MAX_IDLE_TIME_BEFORE_FLOATING_LICENSE_RELEASE_ALONE 30

Here the maximum idle time is redefined only for StandAlone (30 minutes), any other application (DBWorks, DBSolidEdge and DBInventor) refers to generic idle time value (15 minutes).

Reconnect after timeout

Using floating licenses with the DBWORKS_MAX_IDLE_TIME_BEFORE_FLOATING_LICENSE_RELEASE parameter (and/or its specializations), it is now possible to reconnect the MWPDM when the countdown finishes.

Now, if there are no activities and the countdown finishes, the MWPDM license becomes free and all MWPDM functions are disabled for the running application.
Resuming activities, the MWPDM will reconnect automatically getting the involved license and enabling all MWPDM functions.
If there are no available licenses when resuming activities, a message is provided in the ActivityConsole and the reconnect will be automatically repeated.

Link Mode

New option: [X] Link referenced Drawings/Parts even if checked-out by others

When checked (Default: not checked), it makes possible to checkin/approve drawings/parts created by other designers, but referencing the same documents.
A typical example of use is when multiple designers works for making multiple drawings of the same part.
Suppose User1 creates a Part1 and its Drawing1, and keeps it in checked-out state.
Now User2 opens the Drawing1 and creates a new Drawing2 of Part1.
Without this new option, Drawing2 and Part1 (and so also Drawing1) will always be not linked, because the check-out-by users are different.
By activating this new option, it will be possible for User1 to CHECKIN/APPROVE the Part1 and Drawing1, and have Drawing2 be checked-in/approved as well.
At the same time, User2 would be able to CHECKIN/APPROVE Drawing2, and have Part1 and Drawing1 be checked-in/approved as well.

Local Checkout Mode

Improved Local Copies mode

The Read-Only Local Copies mode has been greatly improved.

Options

New sub-option: Revision->[X] Open documents in 'view-only-mode' if released->[X] Only Drawings

When checked (Default: unchecked), it allows the opening in view-only-mode only to Drawings.
The option may be useful for cases where the correct configuration needs to be always loaded in memory.
As a use case, consider the following:
Part1 has two configurations, cfg1 and cfg2; the last saved configuration is cfg2.
Part1 is approved.
The user selects the Part1/cfg1 in the Browser and opens it.
Part1 is opened view-only, so it can't be saved-as - for such reason, the user RMB->Edit in the SolidWorks part for loading the full model in memory before the save-as.
SolidWorks uses the last saved configuration for loading the full model, so the user, even if when selecting the part in the Browser, selected the cfg1, SolidWorks has loaded the cfg2, being the last saved configuration.
This issue may cause troubles if, for example, cfg1 has an associated drawing, while cfg2 has not.
When saving-as cfg2, no drawing will be created.

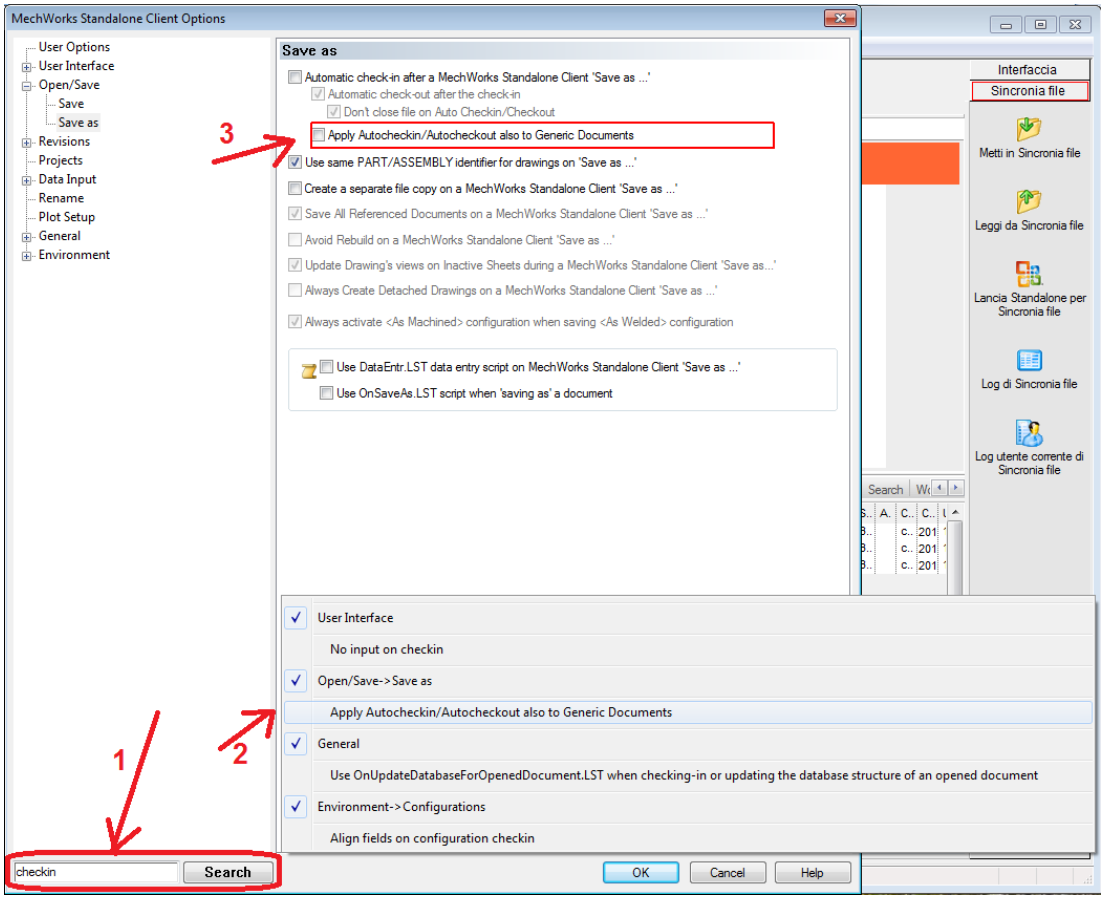
New sub-option: Revision->[X] Avoid database record editing if released->[X] Allow editing on Freeze

When checked (Default: unchecked), it allows the editing of the record when executing the 'Freeze' operation.

Search box available for searching options by name

A new search box is available in the lower left corner of the Options Dialog (see the picture below).

It allows to type a word and to search, through all the available options, for an option name matching the word being inserted (1).
Once the word has been inserted, the *[Search]* button must be pressed: a popup menu will open, with the Pages and the Option Names that matches the inserted word (2).
By selecting the proper menu entry, the specific Options Page will be activated, and the selected option will be highlighted in red (3).



PARENT_CHILD

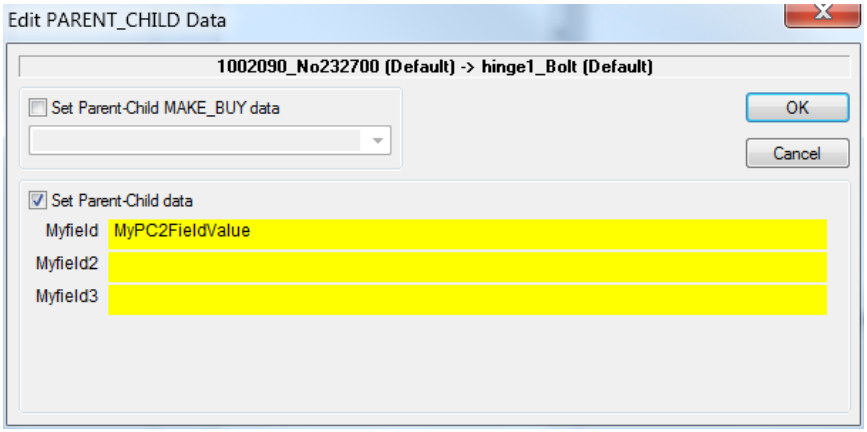
Edit PARENT_CHILD Data

It is now possible to declare the list of the PARENT_CHILD fields to be edited, by adding the section:
PARENT_CHILD
<name field 1>
<name field 2>
…
in the shared SCHEMA\Tables2.DFL definition file.

Example:

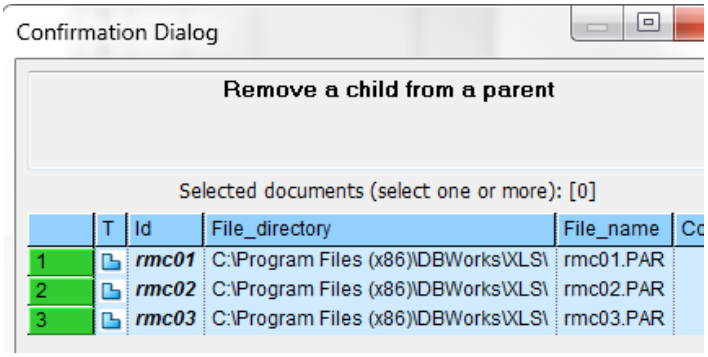
```
@VER 2
;=================================================================
; TABLES2.DFL
;
; DBWorks (C)MechWorks Default Tables definition file
;
…
…
PARENT_CHILD
        MYFIELD
        MYFIELD2
        MYFIELD3
```

The Edit PARENT_CHILD Data form will appear as:



Remove this child: improvement for multiple selected Tree Items

When multiple Tree Items are selected for being all removed from their parents, a single confirmation dialog is now displayed, allowing further selections and a single confirmation:



PDF

New Parameter File for PDF Print Options

It is now possible to create a parameter file for altering some options of the "Plot Pdf" process.
The parameter file is named DBWPdfPrintOptions.PAR, and must be located in the shared \PAR subfolder.
The first option that is implemented is:
PaperByPage <0/1> (Default: 1)
When set to 0, the option "[ ] Choose paper source by PDF-page size" will be unchecked, so the paper source will always be the same set in the printer.
An example of the file is the following:

```
;
;DBWPdfPrintOptions.PAR
;
PaperByPage 0
```

Performance

Checkin from the Browser: Model no longer closed while Drawings are checked-in

When checking-in a Model from the Browser, the Model is no longer closed while checking-in its linked Drawings.

Revision outputs or Master Drawing creation: Drawings no longer closed on approval

Drawings are no longer opened multiple times if, after a Revision Approval, Revision Outputs or Master Drawings creation were requested

Tree Expansion Performance improvements

The Tree Expansion Performance has been greatly improved for cases where:
- Existing Drawings were displayed in the Tree

- Extra Parent Child Fields columns were declared as options

Tree of Projects Performance improvements

The Projects Tree Performance has been greatly improved

VPN Mode

For working over slow network, like VPN / Cloud / etc. it is now possible to declare a special parameter in the local DBWAPPH.PAR file:

VPN_MODE_LOCAL_DRIVE C:

The intent is to make possible to connect to a remote server using a low-band-width connection (example, an ISDN connection at 128Kb), accepting a series of limitations in the User Interface.
Of course, the VPN MODE can also be used with a fast connection, for speeding up the overall User Interface for specific cases (example, when working with very large set of data).
When in VPN MODE, the Browser Title shows the label "[VPN MODE]".
Assumptions and limitations of the VPN MODE:

- Global Parameters Path must point to the shared location
- the declared VPN_MODE_LOCAL_DRIVE is the drive where an exact copy of the shared MechWorks_PDM_Server folder can be found (*Example:* if the shared folder is on Z:\MechWorks_PDM_Server, then, declaring VPN_MODE_LOCAL_DRIVE C: will assume that the same folder can be found in C:\MechWorks_PDM_Server)
- The Shared DBWAPP.PAR must be read-only
- All Shared Paths, except USERS, are automatically changed to the local installation directory (shared MechWorks_PDM_Server must be copied locally to the VPN_MODE_LOCAL_DRIVE)
- No Visual Cues
- No Tree/Grid Graphic Features depending from Database Values
- Tree Items expanded status not restored automatically when Browser opens
- No automatic refreshing of ComposedOf/UsedBy lists
- No Document Page: always open in Tree Page
- No in-Workflow documents markers: they can be accessed using the side bar queries

Preview

Added support for .AI (Adobe Illustrator) files

.AI (Adobe Illustrator) files can now be previewed either in the static preview or in the full preview window.

Released Database

New option: [X] Use Released Database for restoring Document and Parent-Child data on Undo Checkin

When enabling this new option (default: not checked), the Released Database data is used for restoring the DOCUMENT and PARENT_CHILD data of the unchecked-in document.
This means that also modifications like:
- change of quantities for real or fake components
- add/remove of Generic Documents
- editing of Database Fields in the DOCUMENT table
will be correctly restored to their last approved state.
Notes: in order to have everything working as expected, **every Child Document (included Generic Documents) must be approved before approving its Parent Document** (so for example, it is suggested to enable the *Revisions->Approval->[X] Avoid assembly approval if components are not released* and *Revisions->Approval->[X] Avoid generic document parent approval if generic document children are not released*).
If any Child Document is not approved, its related Parent-Child info will not be restored using this new option.
In few words, the new functionality applies the exact state of the Released Database for the unchecked-in Document, and **Child Documents in NEW, CHECKED-IN or CHECKED-OUT state will be removed from the Parent-Child Structure after the execution of the Undo Checkin functionality**.

Scripting

Command DBWShell("GetDocumentExternalReferences") supported by Standalone Client

The command is now supported by the Standalone Client for the three CAD file types (SolidWorks, SolidEdge, Inventor) supported by MechWorks PDM.

New Command: DBWShell("GetBrowserTreeCurrentGridInfo")

The command returns useful info from the current Grid being activated in the Browser Tree Page.
The %TMP%\DBWReslt.IN temporary file contains the info about the current Grid.
The DBWReslt.in is formatted as follows:

@CURRENT_ROW
0: no grid has been activated or no rows available in the current grid
>0: the current row in the currently activated Grid

@GRID_NAME <the name of the activated tab>
if @CURRENT_ROW > 0 , the name of the currently activated Grid.

@PREVIOUS_GRID_NAME <the name of the previously activated tab>
if @CURRENT_ROW > 0 , the name of the previously activated Grid.

<column name>=<grid value>
<column name>=<grid value>
...
<column name>=<grid value>

Notes: all the grid and column names are in upper case

Example:

```
Sub Main()
    DBWInit(True)
    DBWShell("GetBrowserTreeCurrentGridInfo")
    currentRow = DBWResult("@CURRENT_ROW")

    If currentRow > 0 Then
        DBWShell("ActivityConsoleCleanAll")
        DBWShell("ActivityConsolePrint ============================")

        gridName = DBWResult("@GRID_NAME")
        DBWShell("ActivityConsolePrint Current_grid=" & Replace(gridName," ","|") )

        previousGridName = DBWResult("@PREVIOUS_GRID_NAME")
        DBWShell("ActivityConsolePrint Previous_grid=" & Replace(previousGridName," ","|") )

        DBWShell("ActivityConsolePrint Current_row=" & currentRow)

        Set fs = CreateObject("Scripting.FileSystemObject")
        Set tfolder = fs.GetSpecialFolder(TemporaryFolder)

        Set a = fs.OpenTextFile( tfolder & "\" & "dbwreslt.in" , ForReading)

        Dim tokens

        Do While a.AtEndOfStream <> True
            line = a.ReadLine
            if line<>"" Then
                If Left(line,1) <> "@" then
                    tokens = Split( line, "=", 2, 1)
                    msg = tokens(0) & "=" & tokens(1)
                    DBWShell("ActivityConsolePrint " & Replace(msg," ","|"))
                End if
            end if
        Loop

        a.Close
```

```
    End If
End Sub
```

Command DBWShell("LoadQuery"): ability to pass a SELECT statement when in WORKINGSET modality

It is now possible to invoke the LoadQuery, in WORKINGSET modality, using a .sql file that contains the actual SELECT statement for the Working Set to be created.

Example:

Suppose to create a text file named MyQuery.SQL in the C:\TEMP\ directory.
It can now contain a statement like:

```
SELECT UNIQUE_ID FROM DOCUMENT WHERE CHECK_OUT_BY='John' AND
STATE='BEING_MODIFIED'
```

Then, the command can be invoked as

```
queryFileName = "C:\temp\MyQuery.sql"
call DBWShell("LoadQuery " & replace(queryFileName," ","|") & " WORKINGSET")
```

New Command: DBWShell("IsRecursiveCheckinApproveOperation")

The command returns a 0/1 if the PDM is currently in a recursive checkin/approve batch operation.
It also returns the info if a document is a top-level document in the checkin/approve stack.

Example:

```
…
call DBWShell("IsRecursiveCheckinApproveOperation")
if (okDBW = False) then exit sub
isRecursiveCheckinApproveOperation = DBWResult("@IS_RECURSIVE_CHECKIN_APPROVE_OPERATION")
isTopLevel = DBWResult("@IS_RECURSIVE_CHECKIN_APPROVE_OPERATION_TOP_LEVEL_DOC")
…
```

New Command: DBWShell("SetBrowserTreeCurrentGrid <tab name>")

The command changes the current grid to the one specified with its tab name (the name that appears in the Tab Control when activated).

Example:

```
Sub Main()
    DBWInit(True)
    tabName = Replace("Working Set"," ","|")
    DBWShell("SetBrowserTreeCurrentGrid " & tabName)
End Sub
```

New Command: DBWShell("ShowCreateNewBomItemDialog")

The command allows to create a new bom item related to the current selected item at tree level. The Browser must be opened (Document or Tree page).

Example:

```
Sub Main()
    DBWInit(True)
    DBWShell("ShowCreateNewBomItemDialog ")
End Sub
```

New Command: DBWShell("ShowChangeQuantityDialog")

The command allows to change the quantity related to the current selected item (assembly or part) at tree level. The Browser must be opened (Document or Tree page).

Example:

```
Sub Main()
    DBWInit(True)
    DBWShell("ShowChangeQuantityDialog")
```

```
End Sub
```

Tree Page's Custom Query Grids Popup Menus support

It is now possible to display an RMB Popup Menu when right clicking a row in a Tree Page's Custom Query Grid, created using a LST\OnCustomQueryTab_xxx.LST script.
The PDM looks now for a subfolder of the shared LST folder, named as follows:

CustomQueryTabMenus\\<name of the custom query tab>

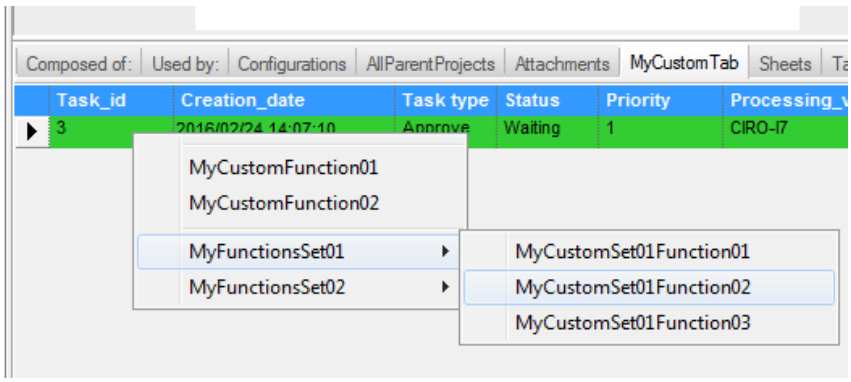
and lists in the RMB popup menu all the scripts found in that folder - the scripts names follows the same rules described in the Help Topic "Custom menu entries", in particolar the rules of the "Order sequences and separators in Scripts RMB Popup Menus" paragraph.

Example:

Suppose we have a script named
LST\OnCustomQueryTab_MyCustomTab.LST
It will create a new tab named "MyCustomTab" in the Tree Page's Grids panel.
For such tab, let's say we build the following structure of folders and scripts:

```
LST\
        CustomQueryTabMenus\
+       MyCustomTab\
            +       (01)MyCustomFunction01.vbs
            +       (02)MyCustomFunction02.vbs
            +       MyFunctionsSet01\
                    +       (01)MyCustomSet01Function01.vbs
                    +       (02)MyCustomSet01Function02.vbs
                    +       (03)MyCustomSet01Function03.vbs
            +       MyFunctionsSet02\
                    +       (01)MyCustomSet02Function01.vbs
                    +       (02)MyCustomSet02Function02.vbs
                    +       (03)MyCustomSet02Function03.vbs
```

The effect will be that when RMB on my Custom Query Grid, I will see the following popup being displayed:



Any script can now execute the
DBWShell("GetBrowserTreeCurrentGridInfo")
for getting the info of the currently selected row.

SolidWorks

Reduced number of Sheet Activations in DBWorks Variant Notes

[SolidWorks/DBWorks integration only]
The procedures that manage the DBWorks Variant Notes have been improved in order to reduce the number of Drawing's Sheet activations.
After a 'Save' or 'Refresh' operation, the Drawing is now always reset to the first Sheet.

Support for 'Promote' case in Configuration Properties

[SolidWorks/DBWorks integration only]

When the DBWorks option *General->Assemblies->[X] Force Make/Buy attribute from SW assembly cfg option:'[X] Don't show child components in BOM.'* is checked, Assemblies with the *Configuration Properties set as Child component display when used as a subassembly: (*) Promote* will be saved with MAKE_BUY=**BOM_GROUP**

Task Manager

Support for offloaded run of time demanding tasks

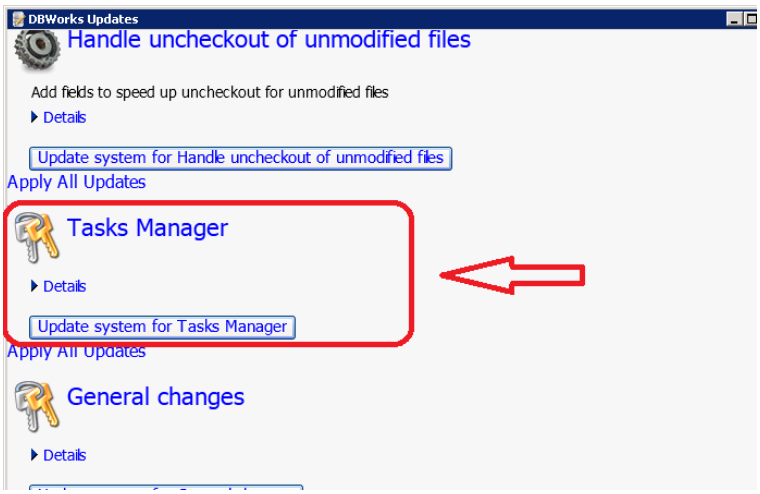
In order to allow users to run time demanding tasks in a different time without overloading their workstation (offloaded mode), a new functionality, named *Task Manager*, has been introduced.
MWPDM has now the capability to queue tasks from users' nodes into a shared system that provides their execution in specific nodes (servers) properly configured in the database, using the same conditions as they had originally at creation time. Users' machines may also be used as executing nodes, but they need to be configured too in the DB.
Users can submit to the task manager activities like *check-in*, *approve* and *plot*: during this phase, a list of available nodes (capable to run that task), will be shown to the user, ordered from lower to higher load, allowing him to choose the specific node where to queue the task. In this way, a basic load balancing policy for task execution is implemented directly by the user launching the Tasks: choosing always the top node will assign the task to the less loaded server.
Each executing node (server) will run all the queued tasks, printing out the status of each command in the Activity Console.
Documents involved in activities queued in the task manager system won't be available for edit purposes until the tasks will be completed.
Hence, for those documents, the edit/revisions context menu item in the browser tree will be disabled whereas the edit properties dialog, opened from the MWPDM menu in the CAD system, will be actually in read only mode with a fading popup alerting the user about the reason.
For the same reasons, opening the document on **SolidWorks/DBWorks** and **SolidEdge/DBSolidEdge** integration) (from both the MWPDM browser and from the CAD environment) will make them available in read only mode with a fading popup alerting the user about the reason. In **Inventor/DBInventor** integration only the popup will be shown.
Once the task is executed, the user will be shown a notification popup signaling the result of the activity being processed.

Pre-requisites: the user creates tasks to postpone actions and, regarding the execution nodes, there are two ways:

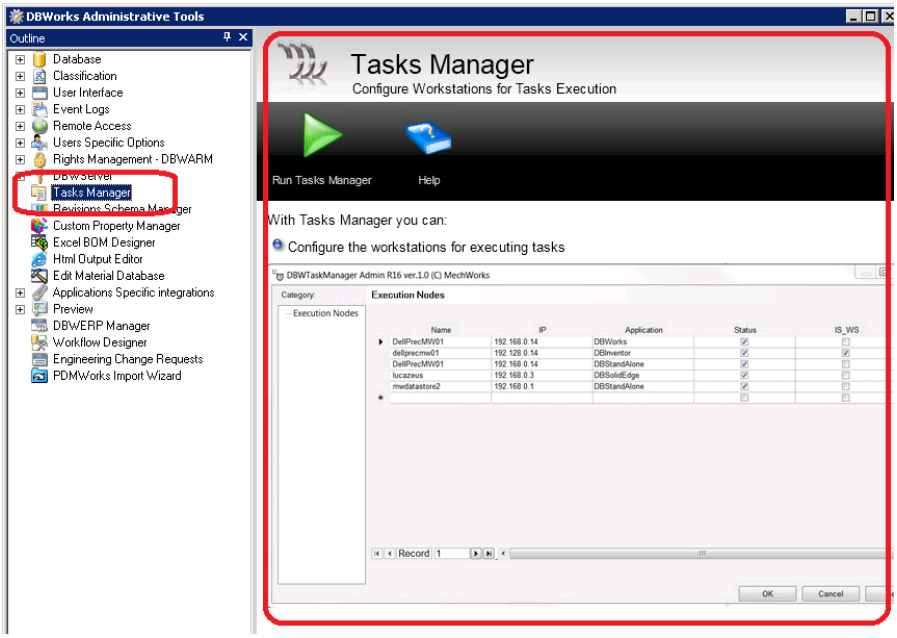
- *Client executing nodes (user machine)*: the same logged in user executes tasks on the same machine (working locally);
- *Server executing nodes (dedicated server machine)*: the logged in user executes the queued tasks on the server (working remotely).

Setup of the environment

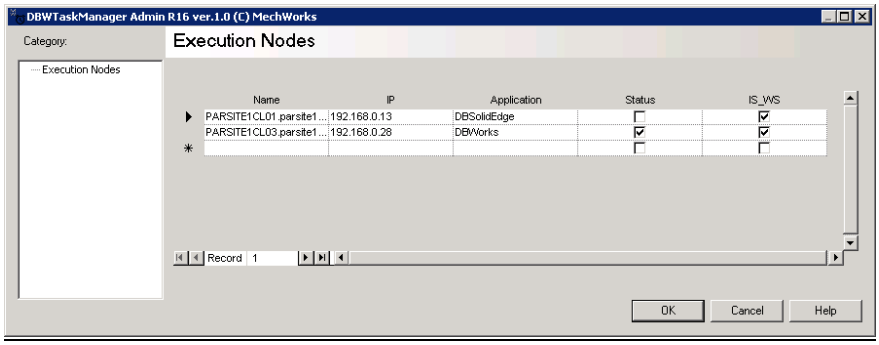
To start using the Task Manager functionalities, the DB must be properly configured. The DB Customizer tool has been changed in order to update the DB schema accordingly to the TM requirements: some new tables will be created, some more fields added to existing ones and some views regenerated:



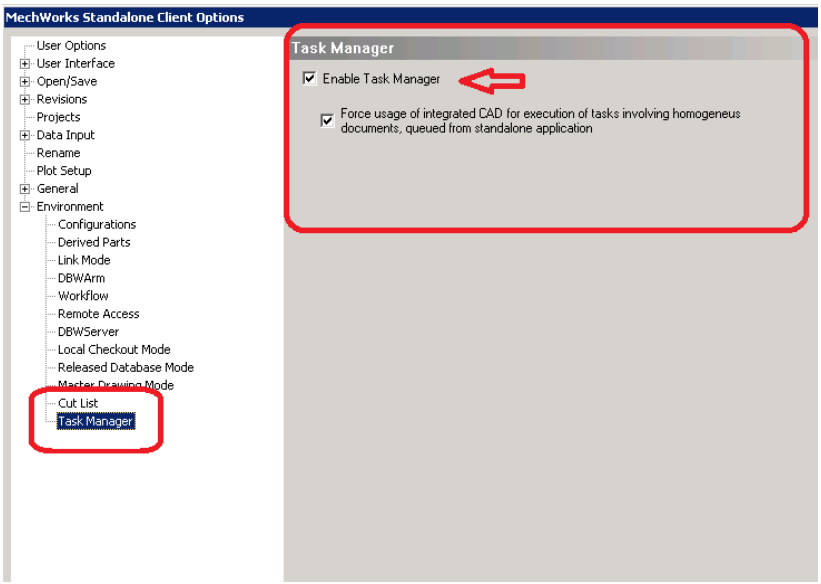
Once the DB is updated, we need to configure all the executing nodes on a new specific table, called DBW_TASK_EXEC_NODES: to achieve this a new tool has been added among the admin tools:



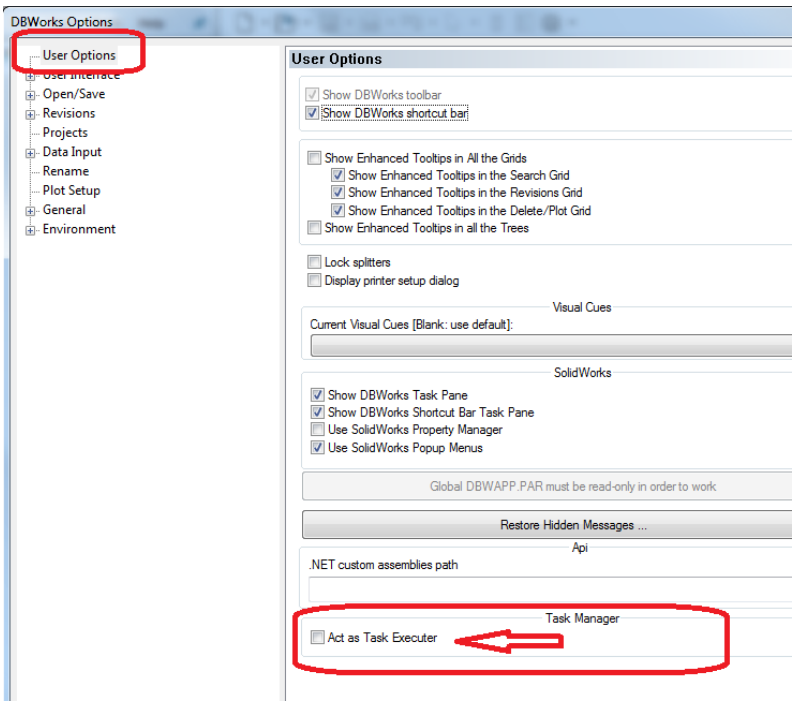
Launching this tool, you will be able to configure the executing nodes, specifying for each of it the name, its IP, the application running on it (which will be used to run tasks), whether it's enabled or not (STATUS), if it's a general working server or a specific user machine (IS_WS):



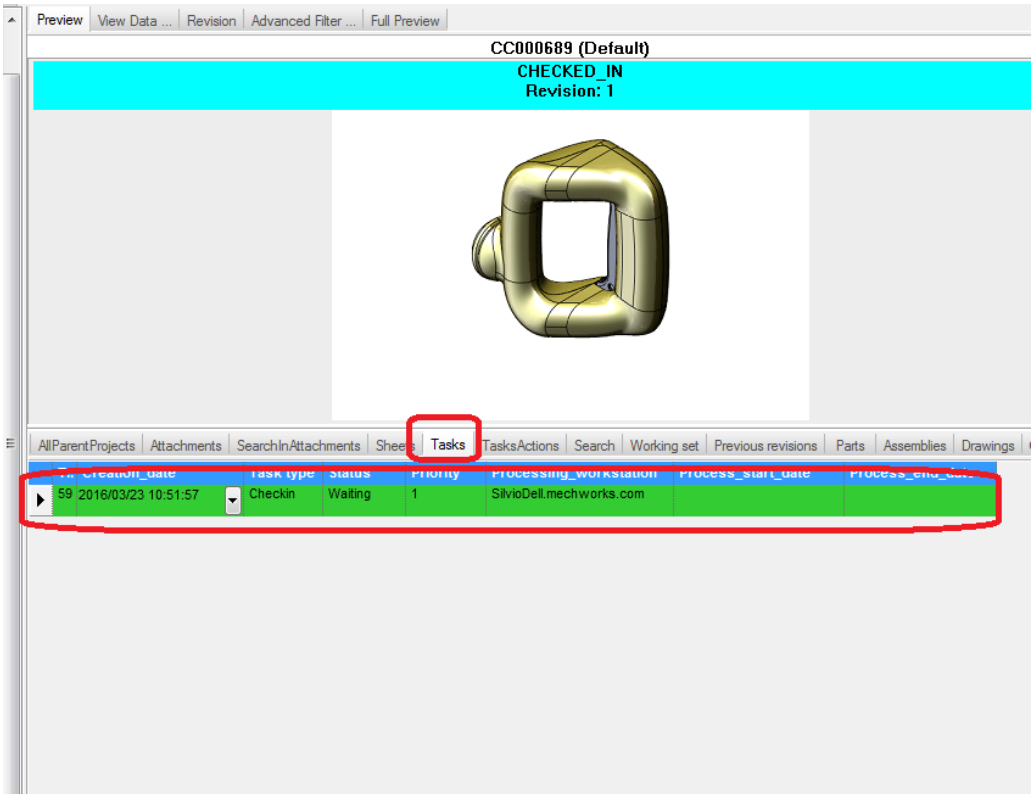
Once the system has been configured, in the DBW PDM option dialog, in the Environment section, a new panel called "Task Manager" will be shown:



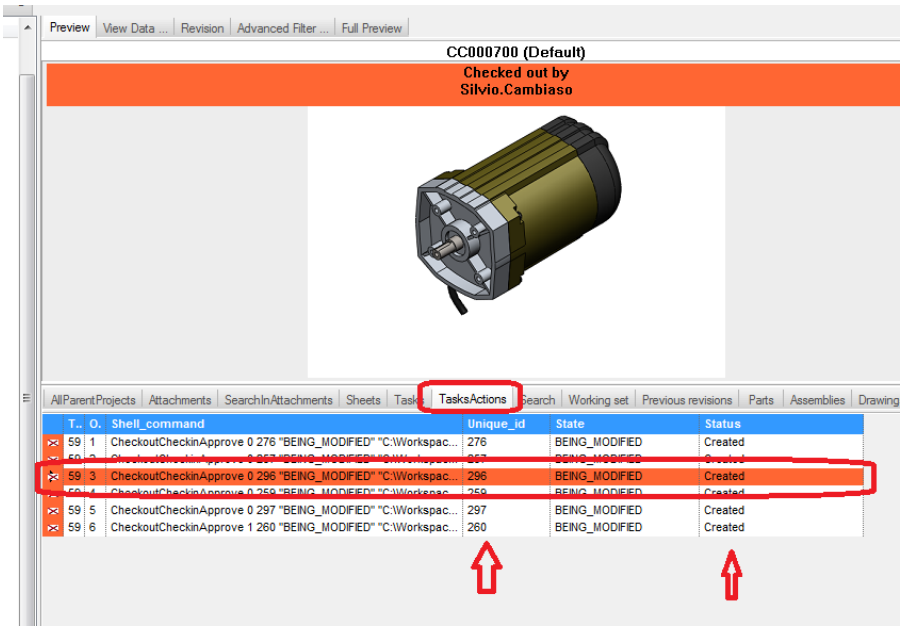
Flagging the "Enable Task Manager" will turn on the functionality.
Notes: this section will be only visible by an administrator and not by a normal user.
If a machine, running a specific CAD application, has been configured as executing node, it will be possible to get the executing dialog by setting a specific option in the User Options Panel:



After the environment has been set and the executing nodes configured, Task Management functionalities are ready to be used. You may notice this in the Browser because two new tabs will be visible, “Tasks” and “TaskActions”:

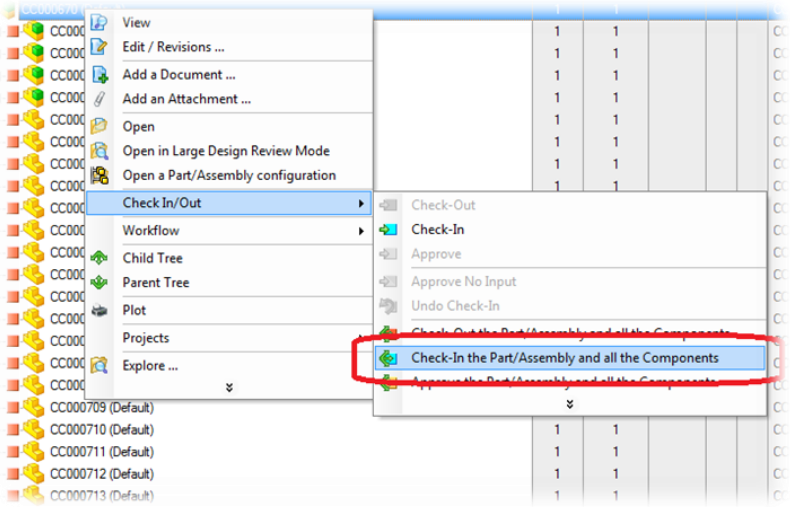


In the “Tasks” tab the user will be shown all of his tasks that have been queued but not completed (i.e. waiting to be executed, failed, paused and stopped). Clicking on one of them and them moving to the “TasksActions”, the user can see all the commands belonging to that task, with their execution state (created, executing, completed, failed) and some more details:

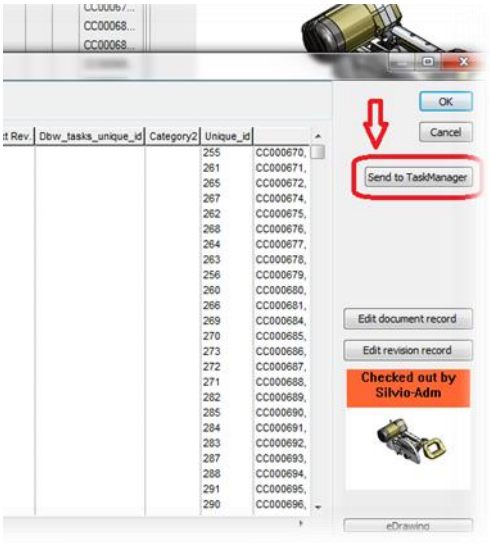


In order to keep those tabs as clear as possible, successfully completed tasks are removed from this tabs because the user can see directly on the tree from the status of the involved documents.

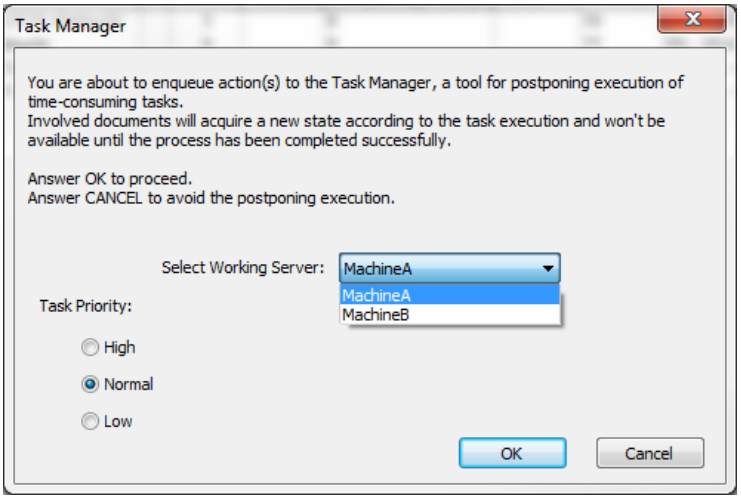
Example 1:
Offloading a check-in of a large assembly:



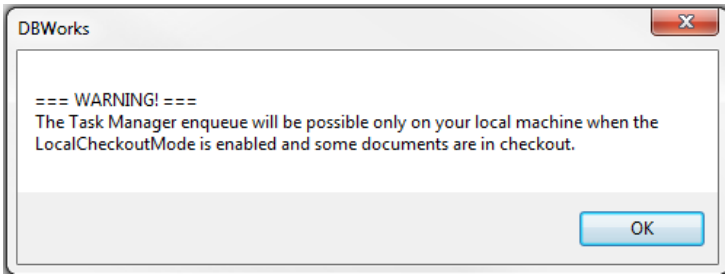
Then, inside the dialog, a new button is available, which will create a task for the task manager instead of executing it directly from that workstation:



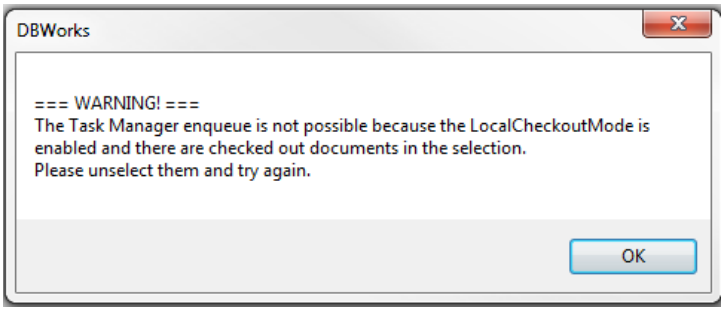
An alerting box will be shown, allowing the choice of the executing node and, only in cases of users with DBWArm Administrative roles, of the priority of the task:



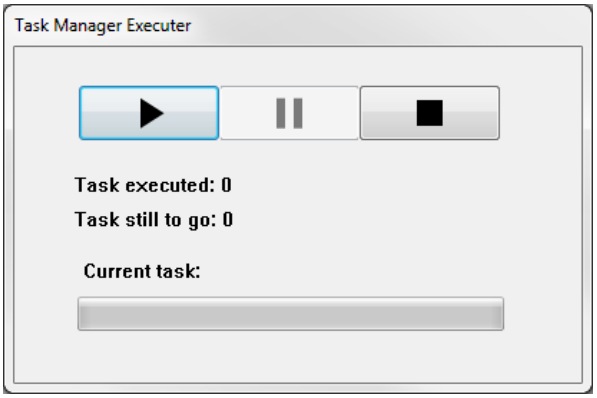
If we are in an environment with local checkout mode enabled, it's still possible to queue a task involving checked out documents, but only on local user machine, which must be configured in the system as well as the server nodes.
In such a case, the user will be shown a message box signaling that only his machine will be in the server list:



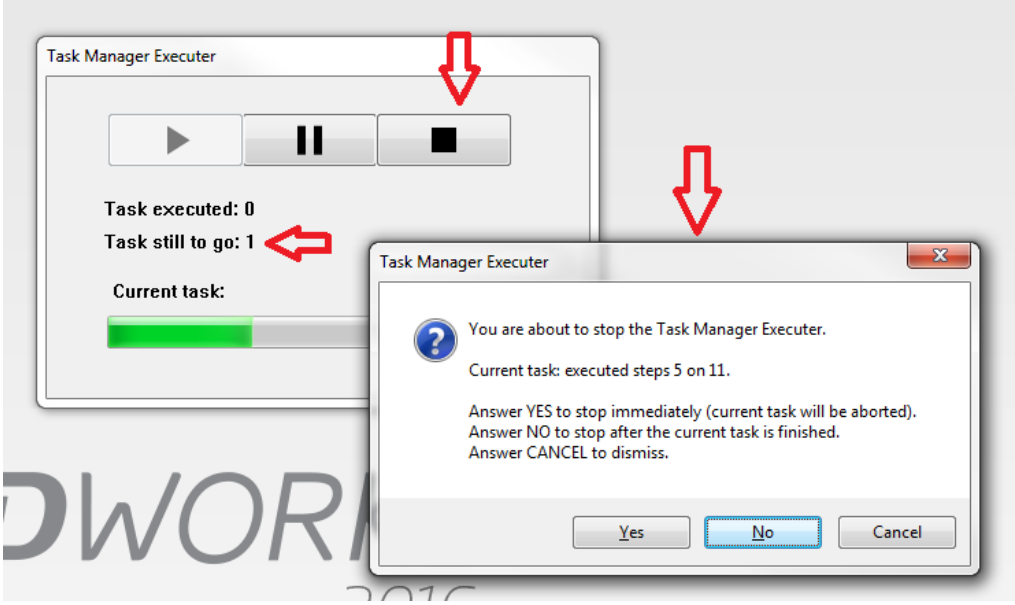
If the user local machine is not configured as executer node, another message is displayed and the queuing activity won't be allowed, unless the users de-selects the checkedout documents in his working set:



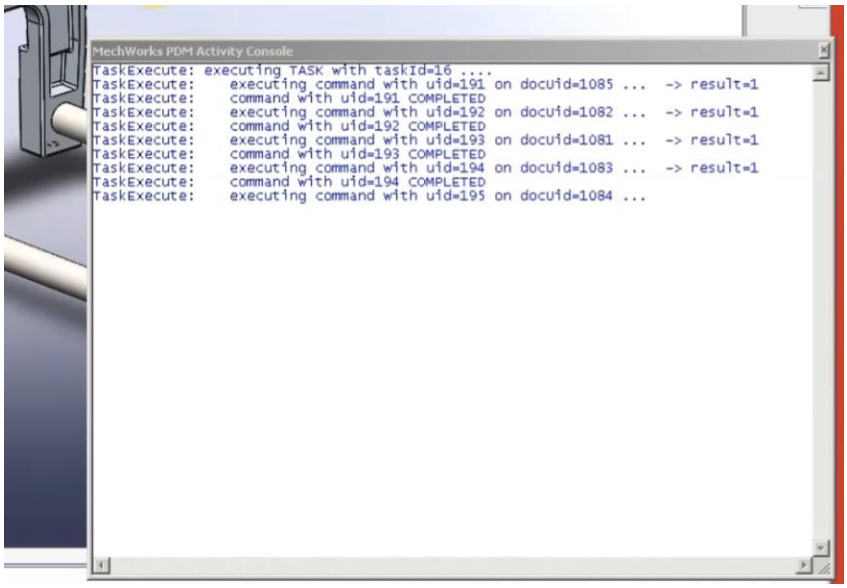
At server side, where tasks are executed, a dialog "*Executer*" is shown, displaying the typical buttons for Play, Pause and Stop:



Play and Pause are pretty self-explanatory, but clicking on Stop button will cause a message appearing if there is one task running and there are still tasks still to go:

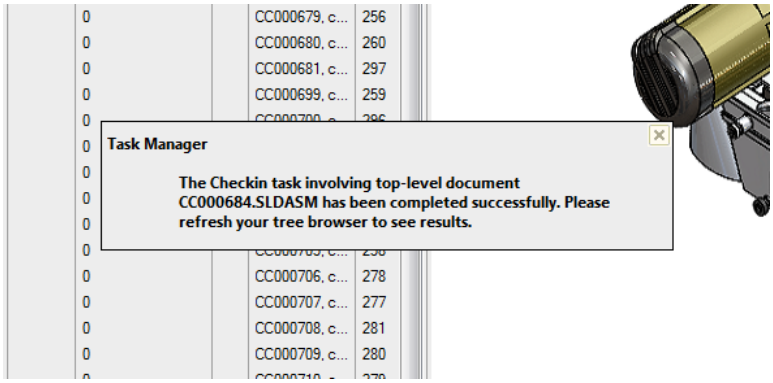


The reason of this message is that the User must be aware that something will happen to the current task according to the choice that will be done.
During the execution, the server application UI will be disabled to get the user mouse events, whereas its activity console will signal all the activities being achieved:

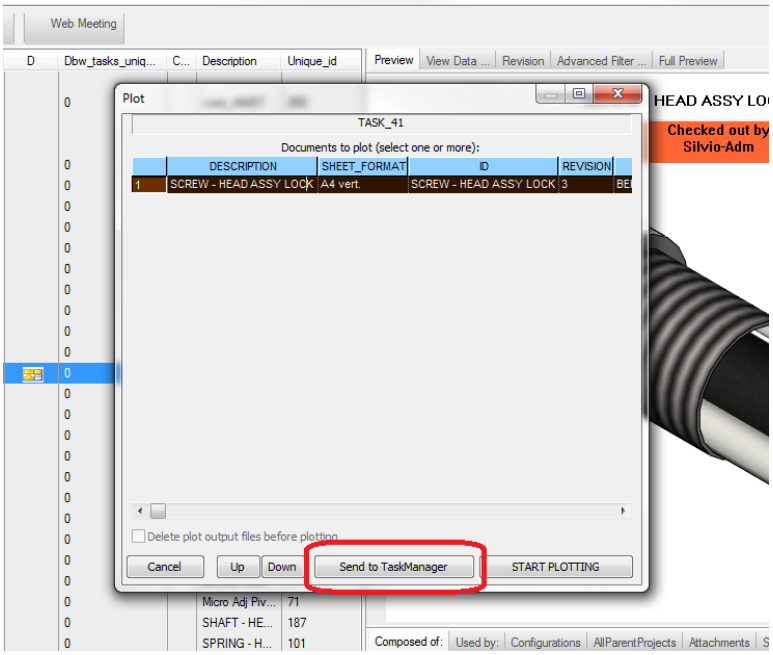


Once the pause/stop buttons are clicked, the server environment gets the input focus again.
At users' workstation, the MWPDM Browser will show up specific icons directly on the Tree, signaling the status of the task execution:

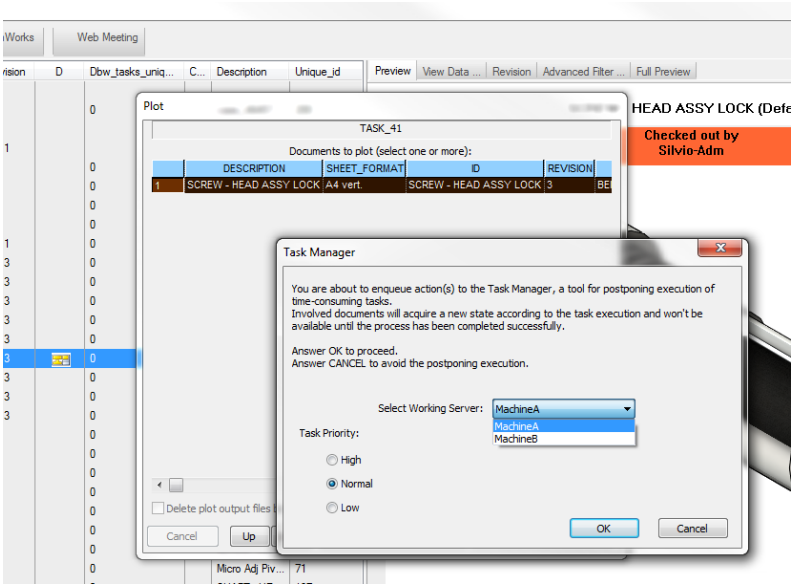
Once the task has been executed, a notification popup will be shown at users' workstation, signaling the result of the activity involving the top-level document:



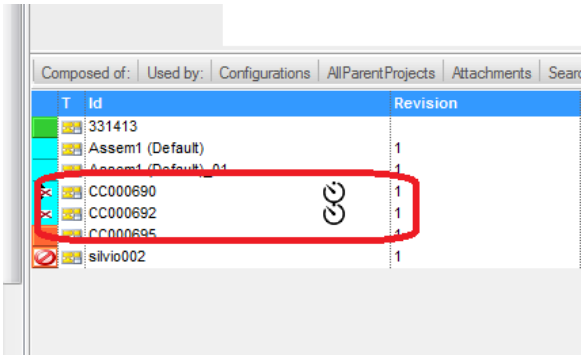
Example 2:
Offloading a plot of documents (click on plot menu item called on multiple drawings selection):



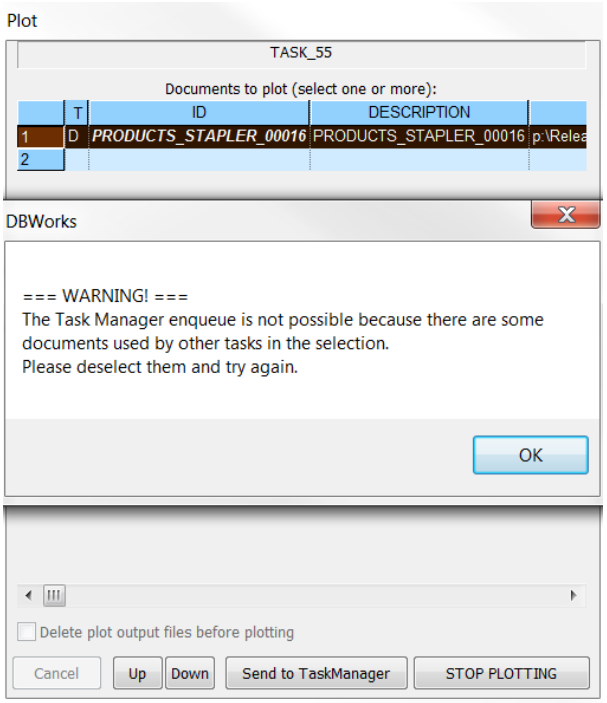
Clicking on the "Send to TaskManager" button will show the same confirmation dialog as seen in the previous example:



After the OK, the involved documents will be shown with the queued icon on the right:



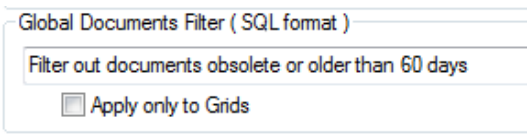
Please note that it is NOT possible to generate new tasks on documents involved in queued tasks: in such a case, the user will be notified by a message box suggesting to deselect.



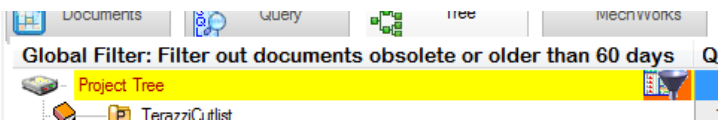
Tree

Global Documents Filter: currently selected filter displayed in the Tree Header

The currently selected Global Documents Filter:



is now displayed in the Tree Header:

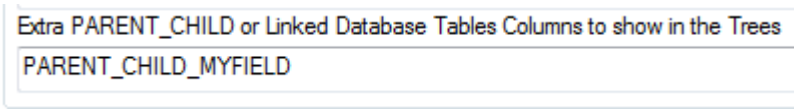


The shared PAR\DBWGDFCustomConditions.PAR is used for assigning the filter alias.

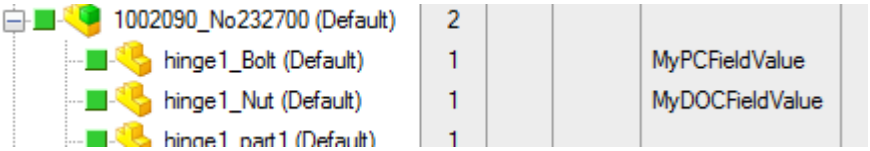
Extra Parent-Child fields: use of an existing DOCUMENT field with the same name

The Tree is now able to display the DOCUMENT table value of a field if the Parent-Child field with the same name has a null value.

Example:
Parent_Child field: PARENT_CHILD_MYFIELD
PART view field: MYFIELD
If PARENT_CHILD_MYFIELD is declared in the option "User Interface->Tree->Extra PARENT_CHILD or Linked database Tables Columns to show in the Trees":



then, if the PARENT_CHILD_MYFIELD value is null, the DOCUMENT::MYFIELD will be displayed for the Tree item; in the following picture, the "hinge1_Bolt (Default)" part has a PARENT_CHILD_MYFIELD value equal to "MyPCFieldValue", while the "hinge1_Nut (Default)" part has a PARENT_CHILD_MYFIELD value that is null, but the PART::MYFIELD value is equal to "MyDOCFieldValue"



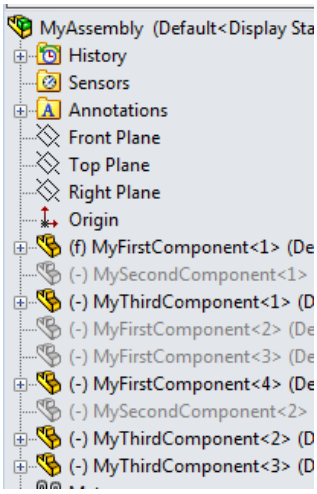
Suppressed Components color codes available in every Tree

[SolidWorks/DBWorks integration only]
When the *Options->Open/Save->Save->[X] Always save suppressed components* option is activated, and with databases updated for supporting the PARENT_CHILD_INSTANCE_ID field in the PARENT_CHILD table, it is now possible to display the same color codes for the Suppressed Components, as it was in the DBWorks Feature Manager of the Assembly is opened in SolidWorks.
The PARENT_CHILD_INSTANCE_ID contains now, for each Suppressed Component, the character star "*" as suffix:

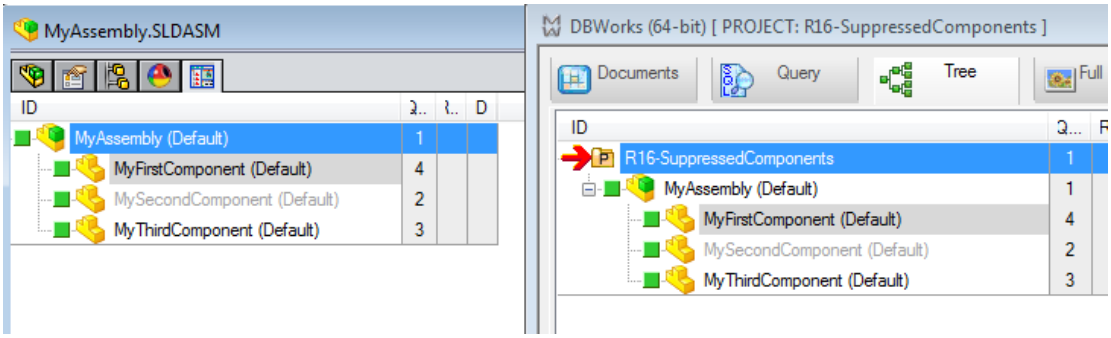
	CHILD_UNIQUE_ID	PARENT_CHILD_INDEX	PARENT_CHILD_INSTANCE_ID
1	9432	0	1
2	9432	0	2*
3	9432	0	3*
4	9432	0	4
5	9433	1	1*
6	9433	1	2*
7	9434	2	1
8	9434	2	2
9	9434	2	3

Of course, the PARENT_CHILD_INSTANCE_ID field will be updated only on next Save/Checkin action.

Example:
For the following SolidWorks Assembly:



where the MyFirstComponent has some suppressed instances, the MySecondComponent has all its instances being suppressed, and the MyThirdComponent has no suppressed instance, its Tree is now displayed everywhere as:

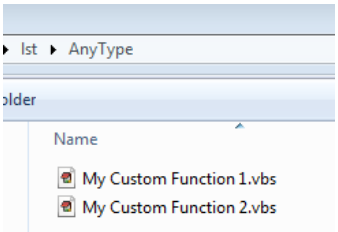


User Interface

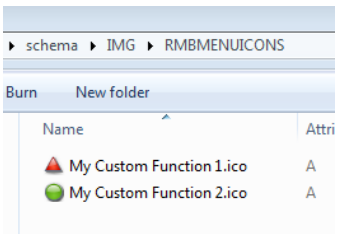
Custom Icons in Custom entries for RMB Popup Menus

It is now possible to display Custom Icons for each distinct custom entry (script) of any RMB Popup Menu. MechWorks PDM checks now if a file .ICO with the same name of the .VBS is existing in the sub-directory SCHEMA\IMG**RMBMENUICONS**

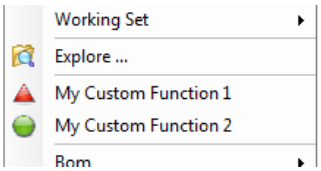
Example:
Suppose to create two new scripts in the LST\AnyType shared folder:



If you like to associate to those scripts a custom icon, you need to create the proper .ICO files, with the same names of the .vbs, under:

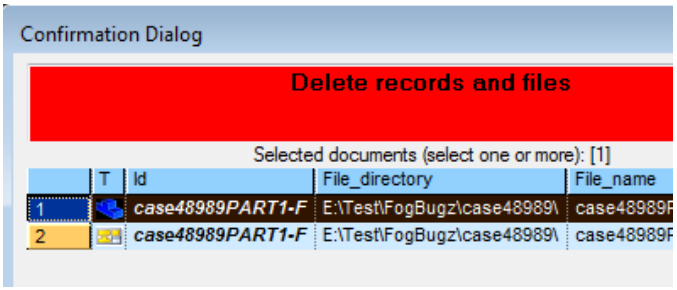


When you will display the RMB popup menu, you will then see:



Delete Documents: "[X] Delete the files" option persistent across different sessions

The *"[X] Delete the files*" option is now persistent across different MechWorks PDM sessions of the same user.
When the option is checked, a different message with a different color is now displayed in the message window of the Confirmation Dialog:



Hidden Popup Menus Definition file

It is now possible to permanently hide any RMB->Popup menu command by declaring it into a new definition file named:

DBWPopupHiddenCmds.TXT

in the SCHEMA directory.
This feature allows the MechWorks PDM Administrator to decide if only a subset of commands must be presented to the MechWorks PDM Users, independently from any other policy.

Following is a sample of the text file with the full list of allowed command identifiers. Any identifier not commented with a semicolumn (;) will hide its related command in the RMB->Popup.

```
;================================================================
; DBWPopupHiddenCmds.TXT
;
; Uncomment any of the listed commands for hiding it
; in the RMB-> popup menus
;================================================================
;ACTIVATE_REVISION
;ACTIVATE_REVISION_INCREMENT_REVISION_STATE
;ADD_A_PROJECT_TO_THE_SELECTION
;ADD_COMPONENT
;ADD_PROJECT
;ADD_SELECTION_OF_PARENT_PROJECTS
;ADD_SUB_PROJECT
;ADD_TO_PROJECT
;APPEND_CHILDREN_WORKING_SET
;APPEND_WORKING_SET
;APPROVE
;APPROVE_TREE
;ASSIGN_BOM_POSITION_NUMBER
;BOM
;BOM_BALLOONS
;BOM_OPEN_EXCEL_BOM
;BOM_OPEN_EXCEL_BOM_REVISION
;BOM_REVISION
;BRANCH_CONFIGURATION
;BUY_LIST
;CHANGE_QUANTITY
```

;CHECKIN
;CHECKIN_GENERIC
;CHECKIN_TREE
;CHECKOUT
;CHECKOUT_GENERIC
;CHECKOUT_TREE
;COLLAPSE_TREE
;COMPARE_ASBUILT_WITH_CURRENT
;COMPARE_REVISION
;CREATE_BOM_ITEM
;CREATE_CHILDREN_WORKING_SET
;CREATE_DIRTY_PARENTS_WORKING_SET
;CREATE_INVALIDATING_CHILDREN_WORKING_SET
;CREATE_NEW_BOM_ITEM
;CREATE_SELECTION_OF_PARENT_PROJECTS
;CREATE_WORKING_SET
;CURRENT_PROJECT
;DELETE_BOM_ITEM
;DELETE_LOCAL_COPY_OF_REMOTE_COMPONENT
;DELETE_LOCAL_COPY_OF_REMOTE_COMPONENT_TREE
;DOWNLOAD_REMOTE_COMPONENT
;DOWNLOAD_REMOTE_COMPONENT_TREE
;DWG_PREVIEW_REVISION
;DWGVIEW
;EDIT
;EDIT_PARENT_CHILD_RECORD_BOM_ITEM
;EDIT_RECORD_BOM_ITEM
;EVENT_LOG
;EXPAND_TREE
;EXPLORE
;FREEZE
;FREEZE_GENERIC
;GOTO
;GOTO_PROJECT
;HTML_TREE
;INCREMENT_REVISION_STATE
;INCREMENT_REVISION_STATE_GENERIC
;INSERT_GENERIC_ATTACHMENT
;INSERT_GENERIC_DOCUMENT
;LOAD_THE_PROJECT_SELECTION
;LOCAL_CHECKOUT_MODE_COPY_LOCAL_TREE
;LOCAL_CHECKOUT_MODE_CREATE_WORKING_SET_OF_LOCAL_COPIES
;LOCAL_CHECKOUT_MODE_DELETE_LOCAL_ALL
;LOCAL_CHECKOUT_MODE_DELETE_LOCAL_TREE
;MARK_AS_NEW_REFERENCE_DOCUMENT
;MARK_AS_REFERENCED_DOCUMENT
;ON_APPROVE_ON_PREVIOUS_REVISION_SCRIPT
;OPEN
;OPEN_ADVANCED
;OPEN_CONFIGURATION
;OPEN_DRAWING_SHEET
;OPEN_GENERIC
;OPEN_LIGHTWEIGHT
;OPEN_MASTER_DRAWING
;OPEN_SELECTIVE
;OPEN_VIEW_ONLY
;OPEN_WITH
;ORDER_BY
;ORDER_CHILD_INDEX
;ORDER_TREE_BY
;PLOT
;PLOT_CHILDREN
;PRINT
;PRINT_SETUP

;PROJECT_SHARE
;PROJECT_UNSHARE
;REFRESH
;REFRESH_REMOTE_COMPONENTS
;REFRESH_REMOTE_COMPONENTS_TREE
;REMOVE_ATTACHMENT
;REMOVE_CHILD
;REMOVE_FROM_PARENT
;REMOVE_FROM_PROJECT
;REMOVE_PROJECT_FROM_THE_SELECTION
;REMOVE_WORKING_SET
;RENAME
;RENAME_GENERIC
;REPLACE_BOM_ITEM
;REPLACE_SELECTED
;SAVE_THE_PROJECT_SELECTION
;SE_OPEN_ASM_LARGE
;SE_OPEN_ASM_MEDIUM
;SE_OPEN_ASM_SMALL
;SEARCH_IN_PROJECT
;SET_DBWARM_CLASS_FOR_NODE_AND_ALL_THE_CHILDREN
;SHOW_ALL_PROJECTS
;SHOW_COMPONENT
;SHOW_CURRENT_PROJECT_DOCUMENTS_ONLY
;SHOW_FOREIGN_DOCUMENTS
;SHOW_FOREIGN_PROJECTS
;SHOW_SELECTED_PROJECTS_ONLY
;SWITCH_CHILDREN_PARENTS
;TAKE_OWNERSHIP_OF_REMOTE_COMPONENT
;TAKE_OWNERSHIP_OF_REMOTE_COMPONENT_TREE
;TRANSFER_OWNERSHIP_OF_PROPRIETARY_COMPONENT
;TRANSFER_OWNERSHIP_OF_PROPRIETARY_COMPONENT_TREE
;TREE_CHILD_PROJECT
;TREE_CHILDREN
;TREE_PARENT_PROJECT
;TREE_PARENTS
;TREE_PROJECTS
;UNCHECKIN
;UNCHECKIN_GENERIC
;UNCHECKOUT
;UNCHECKOUT_GENERIC
;VIEW
;VISUALIZE_REVISION
;WORKFLOW_APPROVE_ECO_ID
;WORKFLOW_ECA
;WORKFLOW_ECR
;WORKFLOW_MULTIPLE_ECA
;WORKFLOW_MULTIPLE_ECR
;WORKFLOW_MULTIPLE_RAISE_ECR
;WORKFLOW_NEXT_STATE
;WORKFLOW_OPEN_ATTACHMENT
;WORKFLOW_PREVIOUS_STATE
;WORKFLOW_RAISE_ECR
;WORKFLOW_RESET_ECO_ID
;WORKFLOW_RESET_ECR_ID
;WORKFLOW_SHOW_USERS_FOR_STATE
;WORKING_SET_APPROVE_TREE
;WORKING_SET_CHECKIN
;WORKING_SET_CHECKIN_TREE
;WORKING_SET_CHECKOUT
;WORKING_SET_CHECKOUT_TREE
;WORKING_SET_EDIT
;WORKING_SET_EDIT_REVISION
;WORKING_SET_INCREMENT_REVISION_STATE

```
;WORKING_SET_LOAD
;WORKING_SET_LOCAL_CHECKOUT_MODE_COPY_LOCAL
;WORKING_SET_LOCAL_CHECKOUT_MODE_DELETE_LOCAL
;WORKING_SET_MASTER_DRAWING_COPY_TO_CLIPBOARD
;WORKING_SET_OPEN
;WORKING_SET_OPEN_VIEW_ONLY
;WORKING_SET_REMOTE_ACCESS_DELETE_LOCAL_COPY
;WORKING_SET_REMOTE_ACCESS_DOWNLOAD
;WORKING_SET_REMOTE_ACCESS_TAKE_OWNERSHIP
;WORKING_SET_REMOTE_ACCESS_TRANSFER_OWNERSHIP
;WORKING_SET_RESET_DBWARM_CLASS
;WORKING_SET_SAVE
;WORKING_SET_SET_DBWARM_CLASS
;WORKING_SET_SHOW_IN_QUERY_PAGE
;WORKING_SET_UNCHECKIN
;WORKING_SET_UNCHECKOUT
;WORKING_SET_WORKFLOW_ECA
;WORKING_SET_WORKFLOW_ECR
;ZOOM_100
;ZOOM_IN
;ZOOM_OUT
```

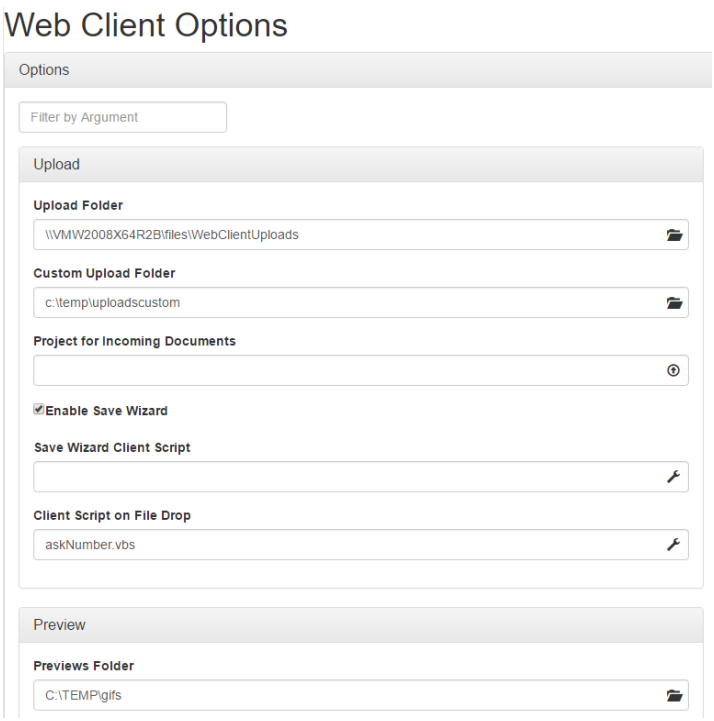
Example:
Let's hide permanently the *Open With* ... and the *Expand Tree* commands - the definition file will be created as:

```
;================================================================
; DBWPopupHiddenCmds.TXT
;
; Uncomment any of the listed commands for hiding it
; in the RMB-> popup menus
;================================================================
EXPAND_TREE
OPEN_WITH
```

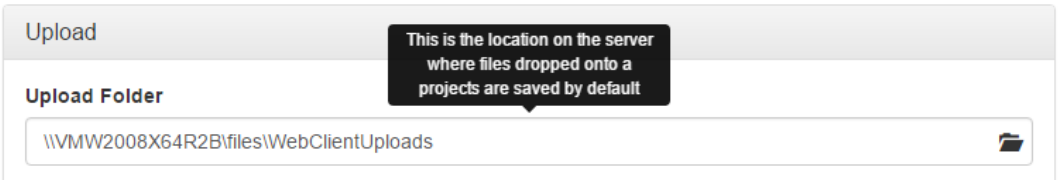
Web Client

Web Client Options

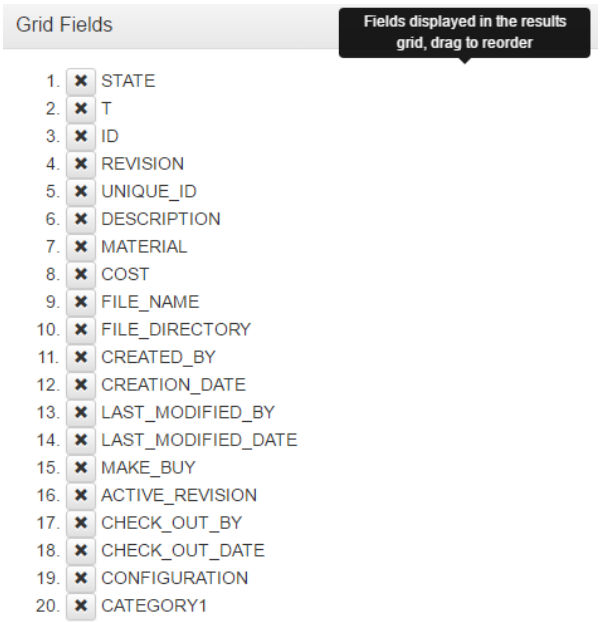
DBWorks Web Client R16 comes with new html based options available to administrator from any client.
The interface is intuitive and simple and groups the options in groups



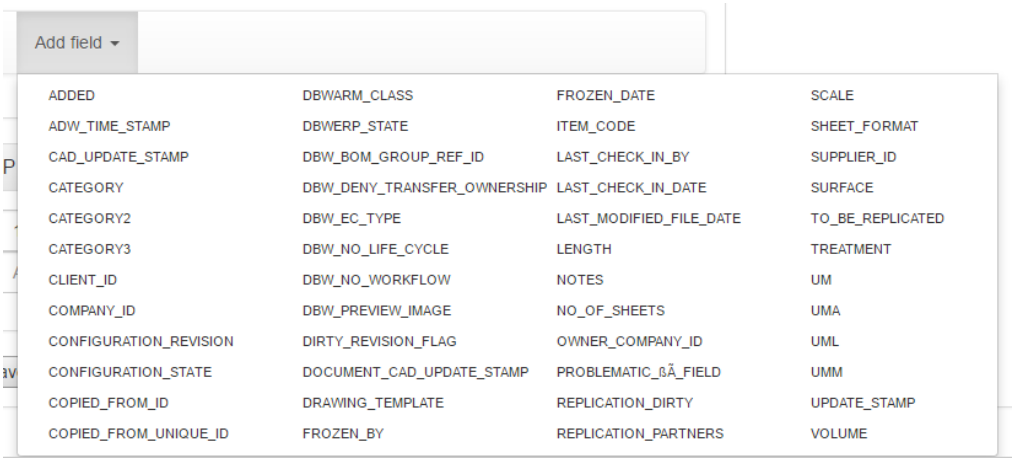
When you move the mouse over an option a popup provides useful hints on how the option affects the behavior of the software



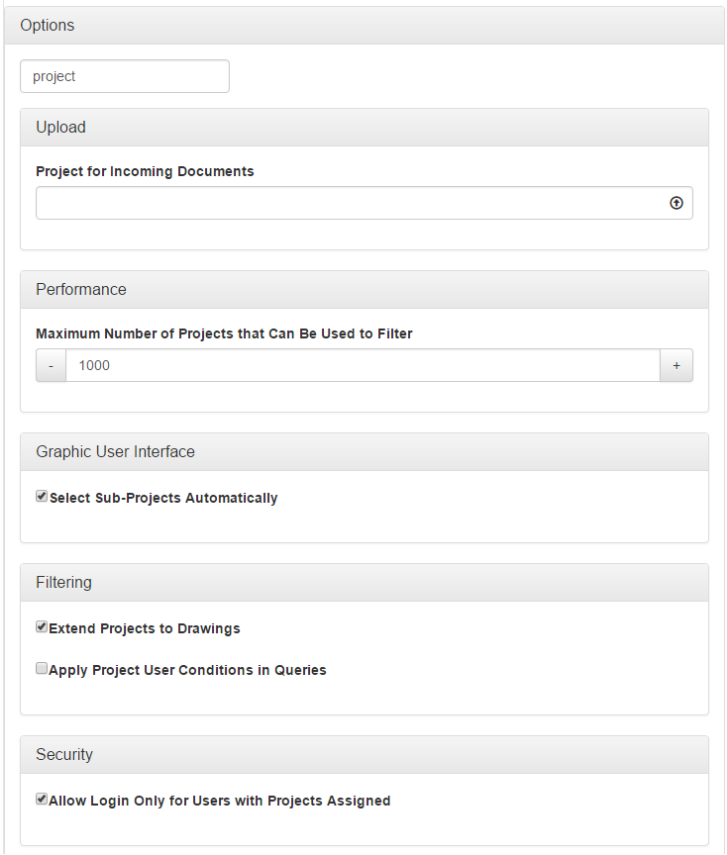
The fields in the result grids can be easily removed with a click or sorted using drag and drop



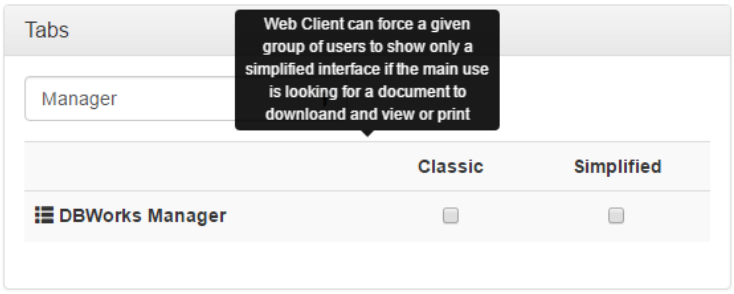
Adding a field is simple: a dropdown panel displays all the fields not still displayed at a glance



What is more important, if you are focused on changing the behavior of the software on one specific topic, like projects or performance, a search box displayed on top allows to filter quickly, displaying only the related options



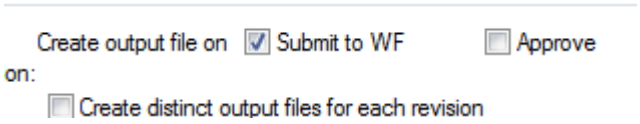
The same filtering panel applies when you want to set specific rights: type part of the group name and the list displays only the matching groups



Workflow

Ability to create Revision Output Files when submitting a Document to the Workflow

A new option has been created for being able to create a Revision Output File when submitting a Document to the Workflow.
The option changes the behavior of the existing Revision Outputs option for creating the Revision Output File at CHECKIN time.
The new option is located in the *Environment->Workflow* tab, and named:
[X] Create Revision Outputs on Submit to Workflow
When this option is activated, the Revision Outputs options display the '*Create output file on [X] Checkin*' option as '*Create output file on [X] Submit to WF*' :



Notes:
1) the Revision Output Files are created at the end of the ECO_ID creation process, because such process may be canceled by the user at any time
2) the current *LST\OnSubmitToWorkflow.LST* should be changed for avoiding the creation of files internally to the script (example eDrawings files) - in such case the Revision Output options could be changed (by customizing the *SCHEMA\DBWRevisionsOutputFileExtensions.txt*) for creating multiple files at once - *Example:* PDF+EDRW
3) if #2 is not possible, then at least it should be avoided to use the out-of-the-box *DBWCreateEDrawingAs* library procedure, because it automatically CLOSE the Document that has been opened for the creation of the EDRW file - a custom copy of such procedure should be used, with NO DOCUMENT CLOSE at the end - this will allow any further Revision Output File to be created without re-opening the previously closed Document

Submit To Workflow RMB popup menu enabled also for Checked-Out Documents

It is now possible to directly submit a Document to the Workflow from the Browser interface, even if the Document is in a Checked-Out state.
The PDM will execute a Checkin before submitting the Document to the Workflow.
We remember here that it is already possible to Checkin and Submit to Workflow from the CAD Command Bar, by keeping pressed the CTRL key while pushing the Checkin button in the Command Bar (feature added in R15)

Workflow Pane and Workflow Designer: custom label for a given process

Added the ability to define a custom label for a given process.
The label is displayed in the Workflow Pane. The ability to set its value has been added in the Workflow Designer in the properties dialog of each process.
The SCHEMA\DBWorkflowDef.txt file supports now the keyword MSG_PROCESS_LABEL:

```
BEGIN_PROCESS MODELS
…
MSG_PROCESS_LABEL <My custom label>
...
END_PROCESS
```

Obsolete Options

Environment->[X] Avoid invisible documents on checkin and on refresh variant notes
Environment->[X] Never traverse assembly structures
Environment->Remote Access->[X] Enable Local Mode
User Interface->Preview->[X] Preview of generic documents
User Interface->Preview->[X] Preview of generic documents launching the associated applications