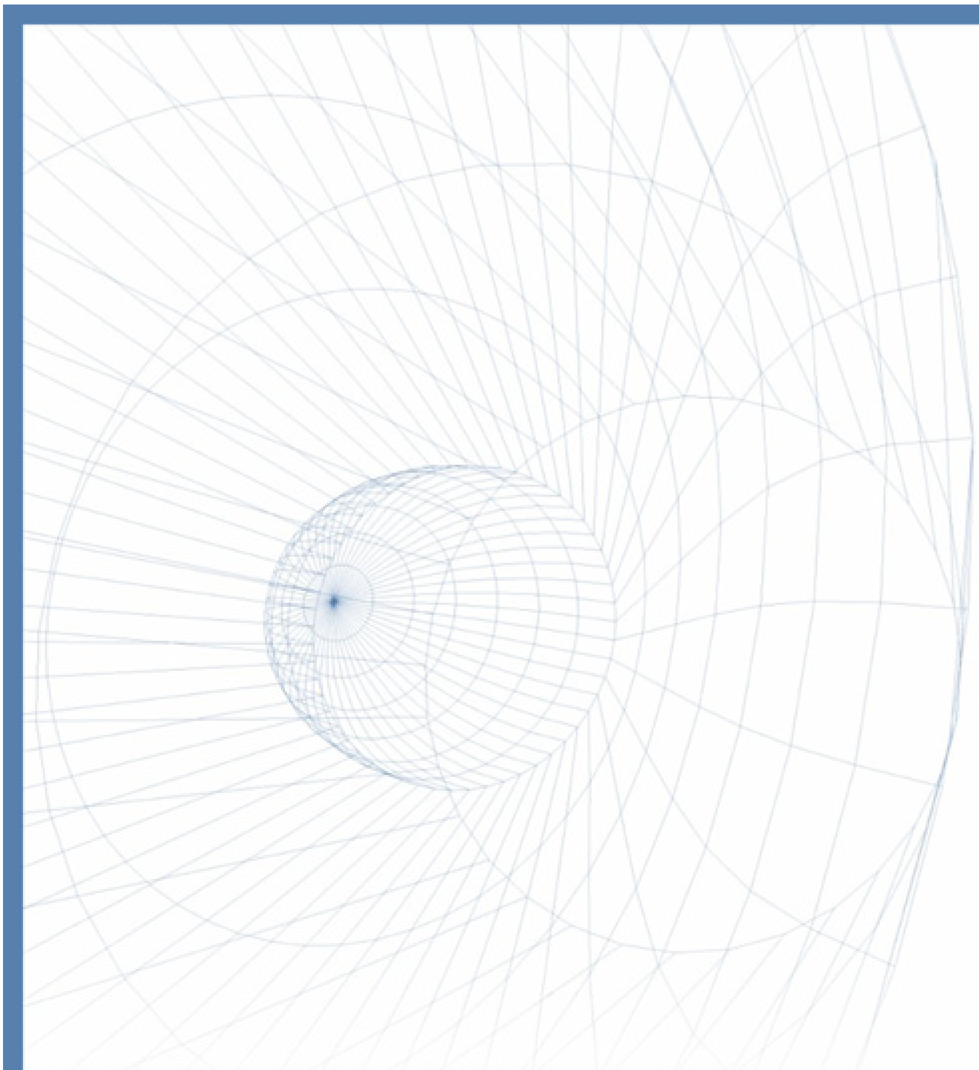


Hull in Dabacon

User Guide

www.aveva.com

AVEVA Solutions Ltd

Disclaimer

Information of a technical nature, and particulars of the product and its use, is given by AVEVA Solutions Ltd and its subsidiaries without warranty. AVEVA Solutions Ltd and its subsidiaries disclaim any and all warranties and conditions, expressed or implied, to the fullest extent permitted by law.

Neither the author nor AVEVA Solutions Ltd, or any of its subsidiaries, shall be liable to any person or entity for any actions, claims, loss or damage arising from the use or possession of any information, particulars, or errors in this publication, or any incorrect use of the product, whatsoever.

Copyright

Copyright and all other intellectual property rights in this manual and the associated software, and every part of it (including source code, object code, any data contained in it, the manual and any other documentation supplied with it) belongs to AVEVA Solutions Ltd or its subsidiaries.

All other rights are reserved to AVEVA Solutions Ltd and its subsidiaries. The information contained in this document is commercially sensitive, and shall not be copied, reproduced, stored in a retrieval system, or transmitted without the prior written permission of AVEVA Solutions Ltd Where such permission is granted, it expressly requires that this Disclaimer and Copyright notice is prominently displayed at the beginning of every copy that is made.

The manual and associated documentation may not be adapted, reproduced, or copied, in any material or electronic form, without the prior written permission of AVEVA Solutions Ltd. The user may also not reverse engineer, decompile, copy, or adapt the associated software. Neither the whole, nor part of the product described in this publication may be incorporated into any third-party software, product, machine, or system without the prior written permission of AVEVA Solutions Ltd, save as permitted by law. Any such unauthorised action is strictly prohibited, and may give rise to civil liabilities and criminal prosecution.

The AVEVA products described in this guide are to be installed and operated strictly in accordance with the terms and conditions of the respective licence agreements, and in accordance with the relevant User Documentation. Unauthorised or unlicensed use of the product is strictly prohibited.

First published September 2007

AVEVA Solutions Ltd, High Cross, Madingley Road, Cambridge, CB3 0HB, United Kingdom

Trademarks

AVEVA and Tribon are registered trademarks of AVEVA Solutions Ltd or its subsidiaries. Unauthorised use of the AVEVA or Tribon trademarks is strictly forbidden.

AVEVA product names are trademarks or registered trademarks of AVEVA Solutions Ltd or its subsidiaries, registered in the UK, Europe and other countries (worldwide).

The copyright, trade mark rights, or other intellectual property rights in any other product, its name or logo belongs to its respective owner.

AVEVA Hull in Dabacon

Contents **Page**

Hull in Dabacon

1 Hull in Dabacon

1.1 General

1.1.1 Database

All Hull data is stored in Dabacon databases. Hull Design Model data is stored in databases of type DESI (i.e. Initial Design, Structural Design and Detail Design data), while Hull Manufacturing data is stored in databases of type MANU. Other databases are also required for the Hull applications. See appendices, Marine *db setup recommendations*, for details.

For Hull: Marine db recommendations_Hull.xls

For Assembly Planning: Marine db recommendations_Assembly_Planning.xls

For Drafting: Marine db recommendations_Drafting.xls

Hull data is stored in Dabacon as a combination of binary data ('blobs') and explicitly stored data. The binary data is for internal use within the Hull applications, while the explicitly stored Dabacon data is primarily for usage within the other applications (Outfitting, Assembly Planning, Accommodation etc.), for the Common Application Framework (3D display, Explorer, Attribute queries etc.) and PML. The binary data is invisible outside the Hull modelling framework.

The binary data is synchronized with the explicit data upon Apply (see chapter *Workflow* below).

1.1.2 Claiming

Hull objects (i.e. Planar panel, Curved panel, Longitudinal etc.) are implicitly claimed when activated for update in Hull applications. Elements subordinate to these objects (e.g. stiffeners, plates) cannot be claimed individually.

Note: That the user needs to explicitly unclaim the elements to unlock them (also after **SaveWork**). If this is not done, the elements will anyhow be unclaimed when the user exits the application.

1.1.3 Updating Hull Data

Hull elements in Dabacon can be updated only by dedicated Hull application commands and functions. There are some exceptions to this rule, though. Below are some examples where Hull elements can be updated by general commands and functions:

- Add or edit User Defined Attributes (UDAs)
- Add or edit Document links
- Add or edit general Associations

1.1.4 Queries

Hull data is readable by various means, in the same way as any other data in Dabacon e.g. by **Query** command, **Attribute** command etc.

Hull Data Extraction (DEXTR) is still supported. Existing DEXTR keywords can be used (e.g. from Vitesse), but will not be extended.

1.1.5 References and Associations

Relations/topology between Hull elements is maintained by the Hull application.

References from other disciplines (e.g. Outfitting or Assembly Planning) to Hull elements follow the PDMS/Dabacon concept, i.e. as database references.

Associations may be used to create other associations than those provided by the AVEVA Marine applications.

1.1.6 Datal

Datal has been extended to support export and import of Hull elements, in addition to Outfitting elements. See *Database Management Reference Manual* for details.

1.1.7 Hull Drawings

For a description of how Hull drawings are handled, please refer to the *Drafting User Documentation*.

1.1.8 Modules and Applications

In AVEVA Marine a Module contains a number of Applications. It is possible to switch between the Applications within a Module.

Example:

The Module Hull Design includes the following Applications: Planar Modelling, Curved Modelling. Structural Design, Plate Nesting, Panel Line Control.

1.2 Hull Data in Dabacon

Hull data are stored in databases of DESI and MANU types. DESI databases are used for Hull design data of various kinds (in Initial Design, Structural Design and Detail Design). MANU databases are used for Hull Manufacturing data (Hull production parts, nestings, jigs etc.).

Hull Manufacturing elements will be seen in the Explorer, but they cannot be displayed in the 3D graphical window. When graphical display is required, the Hull Drafting canvas will be used.

For details on contents and set up of Hull databases, please see the following documentation:

- Appendices (please see section *Database*)
- Hull Project Setup - Getting started
- Data Model Reference Manual

1.2.1 Block World - Blocks

Hull Blocks can be organized into one or more Hull Block worlds.

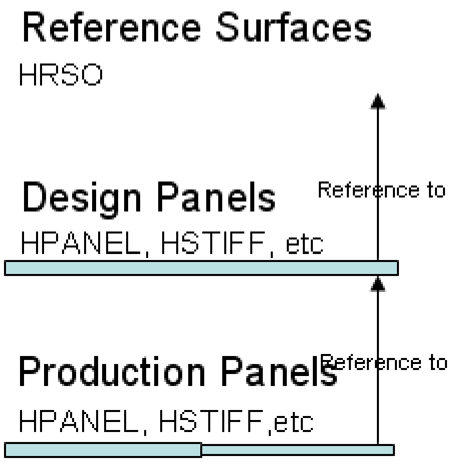
Note: That the Hull Block world element is optional, i.e. the Hull block can be the top level element. It is the customer's decision to use the Hull Block world level or not.

1.2.2 Planar Hull: Initial - Structural - Detail

For Planar Hull, there are three possible phases of model data:

1. Reference Surfaces (created in Initial Hull Design)
2. Design panels (created in Structural Hull Design)
3. Production panels (primarily used in Detail Hull Design)

Please see the picture below. See also the *Hull Documentation* for further details. The database keeps and maintains references between these three versions of the model.



It should be noted that the same element types are used for Design panels and Production panels, including their subordinate elements. There is an attribute to distinguish if the interrogated element is of one or the other type. In functions like **Clasher**, **3D Display**, **Mass Property Calculations** etc. there is an option to switch the obstruction between Design and Production view.

1.2.3 Outfitting Referring to the Hull Model

The Outfitting Designer can choose to work with any of the three model views (Reference Surfaces, Design Panels or Production Panels). The associations in the database are always made to the most stable element (stable meaning unlikely to change), which will be the Reference Surface if this exists, then the Design Panel and lastly the Production Panel.

1.2.4 Symmetry / Reflected - Starboard / Portside

Every physical Hull item is an individual element in the Hull database.

Example:

For a symmetrical Hull block AAA, there will be two blocks in the database: AAA and AAA-R. The same applies for the symmetrical panels and components within such a block.

There are attributes (see appendices in section *Database*) on the Hull element to find out its location (Starboard/Portside/OverCL), if it is Symmetrical (True/False) or if it is Reflected (True/False).

1.2.5 Geometry

The geometry (2D and 3D) is not explicitly stored in the Hull elements. Instead the Hull geometry is generated upon request, e.g. in 3D display, 2D draw, Clasher etc.

1.2.6 Catalogue

Hull applications use the Marine Steel Catalogue for specification data for profiles. The appropriate catalogue must therefore be included in the Hull MDB.

1.2.7 Name Sequence Database

The Name Sequence database (NSEQ database type) is used in Curved Hull applications, for automatic naming of longitudinals, transversals and shell seams.

1.3 Workflow

1.3.1 Terminology

Below are some of the Hull commands used for reading from and writing to the database.

Command	Action	Comment
Activate	Activate panel(s) for update.	Called ‘select’ in Curved Hull modelling.
Apply	Accept changes to the activated panels. Make the framework aware of the changes made.	Previously called ‘store’.
Deactivate	Deactivate panels and revert changes.	Previously called ‘Skip’.
Apply and Deactivate	Accept changes and deactivate the panels.	Previously called ‘store and skip’.
SaveWork	Commit changes to Dabacon. Make the changes available to other sessions.	
GetWork	Get the latest updates from Dabacon (from other sessions).	
Unclaim	Unclaim all or selected elements.	

- ### Hull Workspace

  Every Hull application uses an internal Hull workspace. The Hull workspace is a temporary ‘cache’ for Hull objects. Hull modelling uses Hull Drafting for any graphical user interaction with the model. Within a Hull application function, the Hull object is kept in the Hull workspace and any updates are not known outside this world until the user has executed the **Apply** command.

  The Deactivate (previously ‘skip’) operation means that the updated Hull objects are reverted in Hull workspace. The Dabacon session is not reverted. Therefore, once **Apply** has been executed, the change cannot be reverted.

- ### Apply - Synchronize with Dabacon Session

  When the Hull user executes the **Apply** command, the activated/updated Hull objects are synchronized with the Dabacon session, which makes the updated object known to the Explorer, the 3D Display window, for database queries, for PML etc.

- ### SaveWork - Commit Changes to Database

  To commit the changes to the database the user has to make a SaveWork. This operation works as in other AVEVA Marine applications (e.g. Outfitting), i.e. all applied changes are saved to the database.

  Note: None of the applied changes will be saved unless SaveWork is performed before exiting the application.

- ### Unclaim

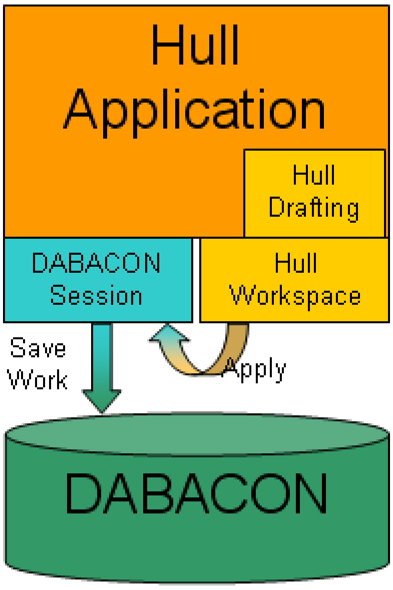
  After SaveWork, in order for other users to be able to update the same elements, the user must explicitly unclaim the elements to unlock them. For this purpose there is a function which combines SaveWork and Unclaim.

- ### GetWork - Retrieve Updates from the Database

  GetWork is used to refresh the Dabacon session with the latest updates from other users/ sessions. Hull workspace is implicitly updated, including the old command **Clean Workspace**.

1.3.2 Workflow Within Hull Design Application

Example:



1.3.3 Executing a Batch Process from Within a Hull Design Application

Before invoking a batch function, the Hull user must do SaveWork and Unclaim.

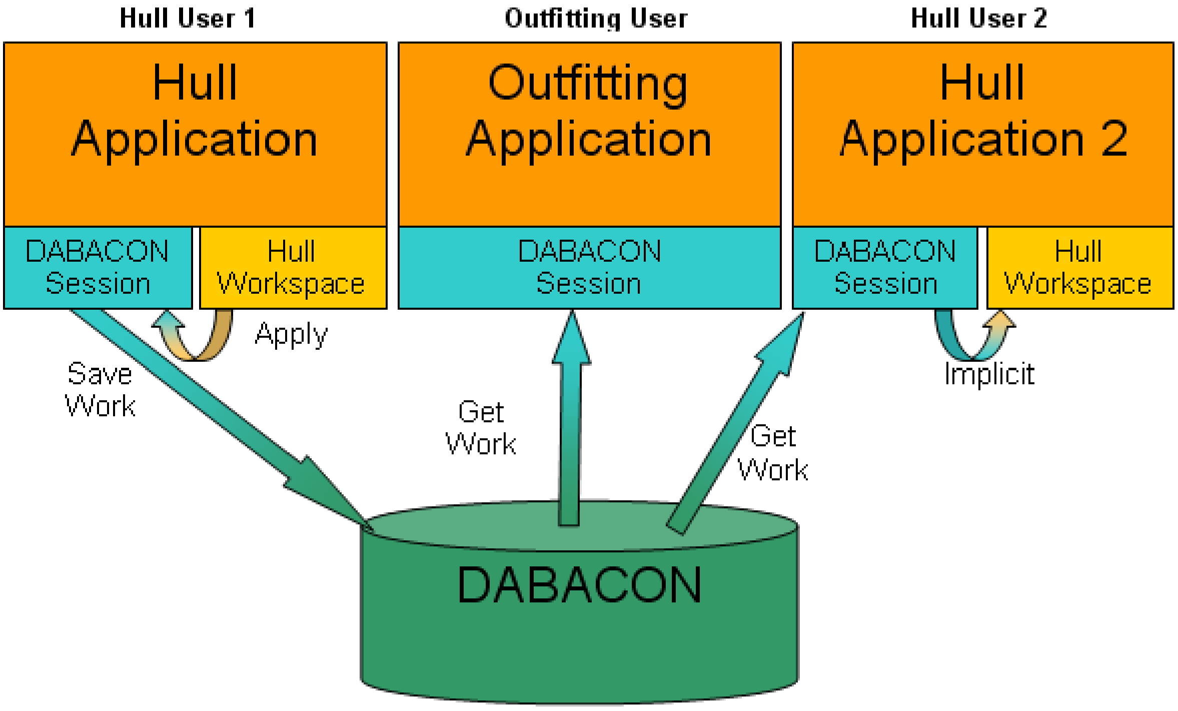
When the batch function is finished, the Hull user must do GetWork to be able to see the updates.

1.3.4 Workflow Between Applications and Users

Example, please see the picture below.

- Hull User 1 makes a change, applies the change and performes SaveWork.
- Outfitting User needs to do GetWork to see the changes from Hull User 1.
- Hull User 2 needs to do GetWork to see the changes from Hull User 1. The Hull objects and elements are, implicitly, read into Hull workspace when accessed by Hull User 2.

Hull workspace is implicitly cleaned to reflect the new status of the Dabacon session.


Hull User 1
Outfitting User
Hull User 2
Hull Application
Outfitting Application
Hull Application 2
DABACON Session
Hull Workspace
DABACON Session
DABACON Session
Hull Workspace
Apply
Implicit
Save Work
Get Work
Get Work
DABACON

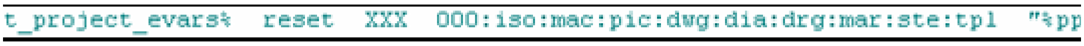
2 Hull Project Setup - Getting Started

To get started with a new Marine project, which includes Hull, follow the outline below:

1. Prepare the project environment.
2. Create the new project.
3. Set up the project environment for Hull.
4. Create the teams, databases, mdbs and users.
5. Create the top elements in the appropriate hull databases.
6. Initiate and configure the hull model.

2.1 Prepare the Project Environment

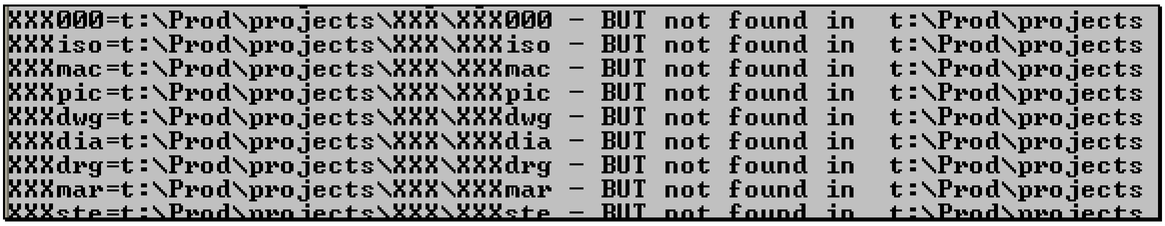
Add the new project code to the evars_projects.bat file. That should look similar like the following. For details on the evars_projects.bat please consult the *ADMIN Reference Guide*.



```
t_project_evars%  reset  XXX  000:iso:mac:pic:dwg:dia:drg:mar:ste:tpl  "%pp
```

Figure 2:1. The edited evars_projects.bat

Executing the evars_projects.bat without having the directory structure define will render the following informational messages.



```
XXX000=t:\Prod\projects\XXX\XXX000 - BUT not found in  t:\Prod\projects
XXXiso=t:\Prod\projects\XXX\XXXiso - BUT not found in  t:\Prod\projects
XXXmac=t:\Prod\projects\XXX\XXXmac - BUT not found in  t:\Prod\projects
XXXpic=t:\Prod\projects\XXX\XXXpic - BUT not found in  t:\Prod\projects
XXXdwg=t:\Prod\projects\XXX\XXXdwg - BUT not found in  t:\Prod\projects
XXXdia=t:\Prod\projects\XXX\XXXdia - BUT not found in  t:\Prod\projects
XXXdrg=t:\Prod\projects\XXX\XXXdrg - BUT not found in  t:\Prod\projects
XXXmar=t:\Prod\projects\XXX\XXXmar - BUT not found in  t:\Prod\projects
XXXste=t:\Prod\projects\XXX\XXXste - BUT not found in  t:\Prod\projects
```

Figure 2:2. Executing evars_projects.bat

The environment variable defined (XXX---) identifies the following project directories:

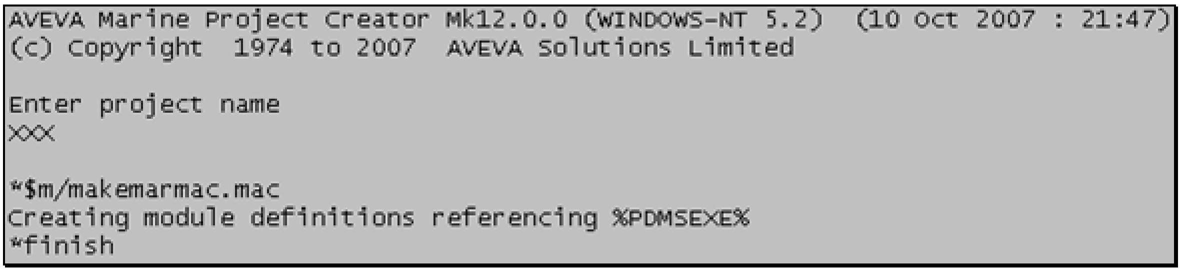
000	Database files
ISO	ISODRAFT project related data
MAC	Inter-db connection macros
PIC	DRAFT picture files

DWG	Drawings
DRG	Drawings
MAR	Hull
DIA	Marine Diagrams
STE	Visio stencils
TPL	Visio templates

2.2 Create the New Project

To create the project directory structure and initialise the databases, the make utility is used.

Run the make utility, and use the makemarmac.mac macro to create a Marine project (for details see the *ADMIN Reference Guide*). The result of the execution should be similar to the following.



```
AVEVA Marine Project Creator Mk12.0.0 (WINDOWS-NT 5.2)  (10 Oct 2007 : 21:47)
(c) Copyright  1974 to 2007  AVEVA Solutions Limited

Enter project name
XXX

*$m/makemarmac.mac
Creating module definitions referencing %PDMSEXE%
*finish
```

Figure 2:3. Running the make utility

2.3 Set Up the Project Environment for Hull

The project environment for Hull is maintained in the Marine project definition file (the D065 file) of the xxxMAR directory. It is initially created and subsequently maintained from within the Admin module.

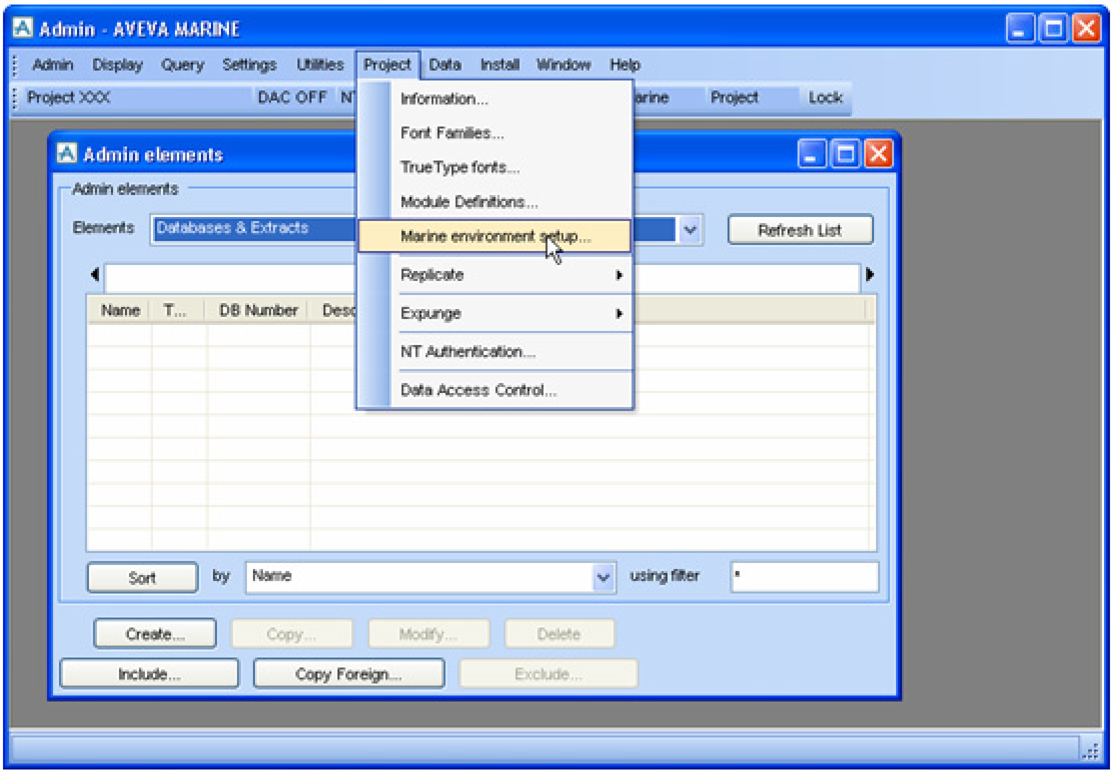
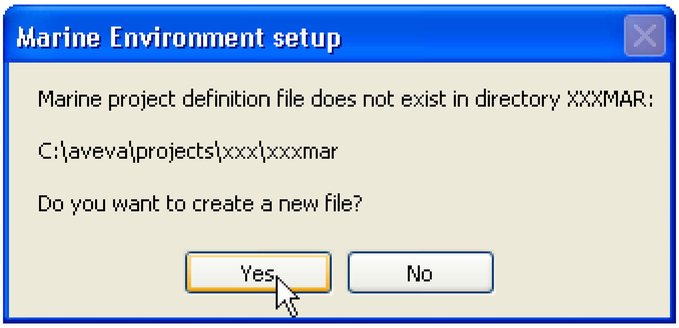

Admin - AVEVA MARINE
Admin Display Query Settings Utilities Project Data Install Window Help
Project XXX DAC OFF N arine Project Lock
Information...
Font Families...
TrueType fonts...
Module Definitions...
Marine environment setup...
Replicate
Expunge
NT Authentication...
Data Access Control...
Admin elements
Admin elements
Elements Databases & Extracts
Refresh List
Name T... DB Number Desc
Sort by Name using filter *
Create... Copy... Modify... Delete
Include... Copy Foreign... Exclude...

Figure 2:4. Initialize project environment for Hull

After starting Admin, invoke the ‘Marine Environment setup…’ and click **YES** on the initial prompt for creating a new ‘Marine project definition file’.



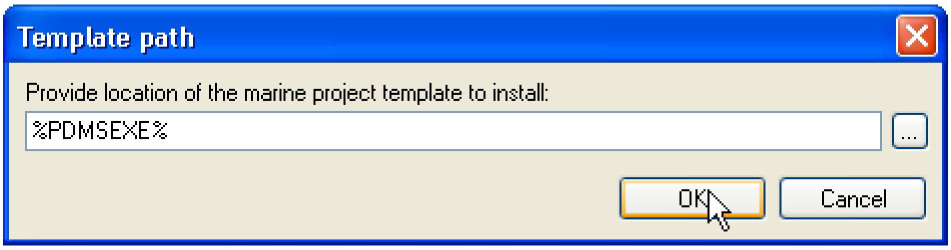
Marine project definition file does not exist in directory XXXMAR:

C:\aveva\projects\xxx\xxxmar

Do you want to create a new file?

*Figure 2:5. Click **Yes** to create the project definition file*

Select the location of the template director.

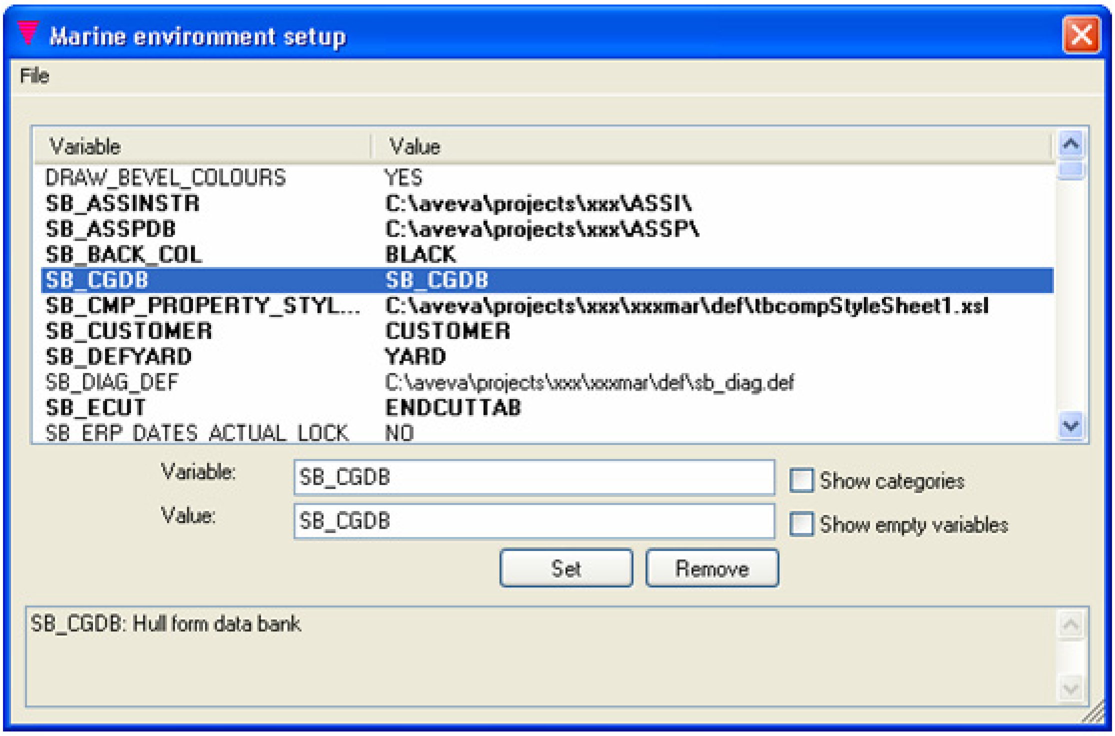


Provide location of the marine project template to install:

%PDMSEXE%

Figure 2:6. Locate the template project

The initial environment for Hull will be created for you. When finished, the Marine Environment Setup editor will start.



Variable	Value
DRAW_BEVEL_COLOURS	YES
SB_ASSINSTR	C:\aveva\projects\xxx\ASSI\
SB_ASSPDB	C:\aveva\projects\xxx\ASSP\
SB_BACK_COL	BLACK
SB_CGDB	SB_CGDB
SB_CMP_PROPERTY_STYL...	C:\aveva\projects\xxx\xxxmar\def\tbcompStyleSheet1.xsl
SB_CUSTOMER	CUSTOMER
SB_DEFYARD	YARD
SB_DIAG_DEF	C:\aveva\projects\xxx\xxxmar\def\sb_diag.def
SB_ECUT	ENDCUTTAB
SB_ERP_DATES_ACTUAL_LOCK	NO

Variable: SB_CGDB

Value: SB_CGDB

☐ Show categories

☐ Show empty variables

SB_CGDB: Hull form data bank

Figure 2:7. Marine Environment Setup editor

Note: A template directory is part of the delivery and resides in the %PDMSEXE% catalogue.

2.4 Create Teams, Databases, MDBs and Users

Before you can use your Marine project, you need to create Databases and Users. A short outline and overview of the necessary steps is laid out in this section. In-depth details are described in the *ADMIN Reference Guide*.

1. Create the Teams that will own your databases.
    - The Team-owning databases concept is described in the *ADMIN Reference Guide*.
2. Create Databases.

    - For details in database planning on which hull databases are required and recommendations on how to set up these databases, please see the appendices documentation *Marine db setup recommendations*.

      Link to appendices: *Database in Chapter Hull in Dabacon*
    - The databases will be of Marine type automatically since we are working in a Marine project.
    - Access mode for the Hull databases should be Multiwrite and Implicit claim.

3. Create MDBs.
    - Hull applications have special requirements on MDBs. Some databases are mandatory and must be present.
4. Create Users.

The following database types are used in the various Hull applications:

- Design (DESI)
    - Holds model data for the Hull Model, required by all Hull applications.
- Manufacturing (MANU)
    - Holds Manufacturing data for Hull, required by Hull Manufacturing applications.
- Name-Sequence (NSEQ)
    - Required by Curved Hull applications.
- Catalogue (CATA)
    - The Marine Steel Catalogue, required by all Hull applications.

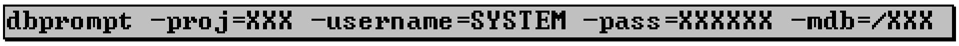
2.5 Create Hull Top Level Elements

The various Hull data is organised in the databases hierarchy under different top-level elements. For instance all RSOs are organised under an RSO top-level element of type RSO-World (RSOWLD), surfaces under a surface-world and panels are organised under Blocks. For full details on the different top-level elements, please consult the *Hull in Dabacon* overview.

The various Hull data is also distributed across multiple databases for access-rights and performance reasons. More details on the subject are available in the *ADMIN Reference Guide*.

DbPrompt is the utility to use to organise your top-level nodes. The tool can be used to interactively create and remove (empty) top-level nodes, export and import definitions on command-line and validate that all necessary top-level elements are in place.

DbPrompt should be used to manage Marine top-level nodes in the DESIgn and MANUfacturing databases.



```
dbprompt -proj=XXX -username=SYSTEM -pass=XXXXXX -mdb=/XXX
```

Figure 2:8.   Starting DbPrompt

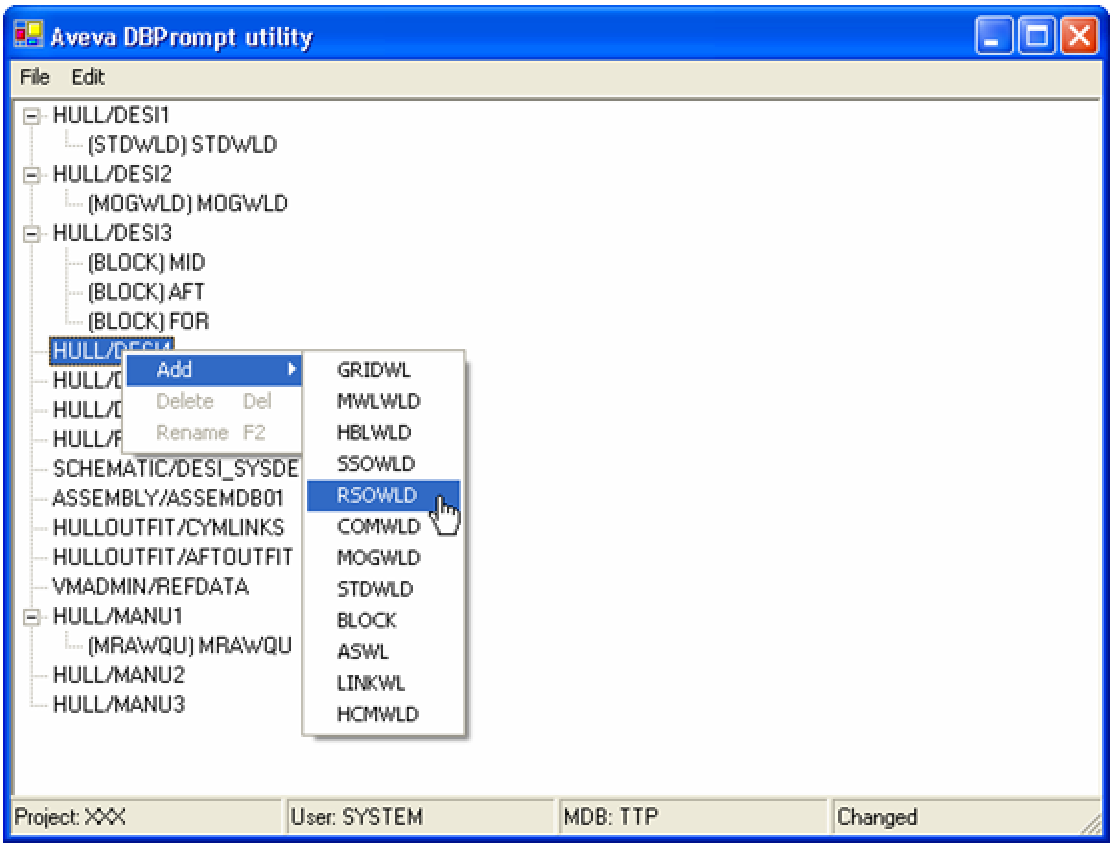
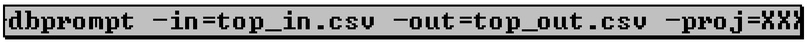


Figure 2:9. Add a top-level node (RSOWLD)

From DbPrompt, you can export and import top-level node definitions in csv-format, suitable for editing in Excel.

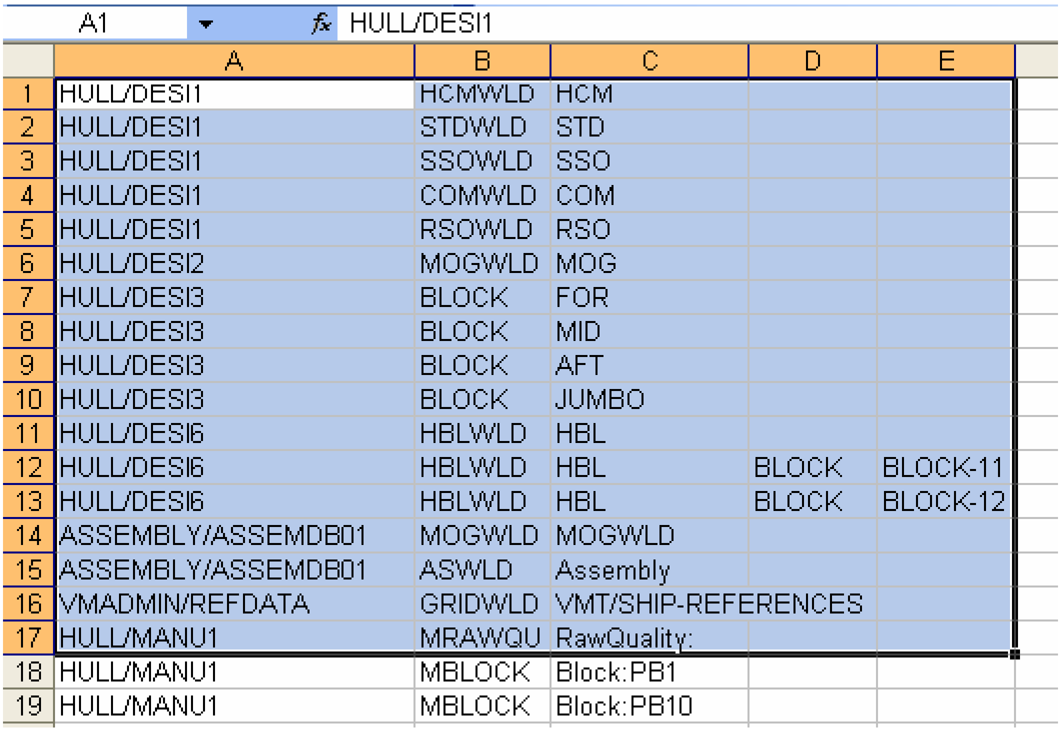
DbPrompt can also be executed on command-line to export and/or import top-level node definitions. The command is then extended by the two optional parameters -in and -out.

Note: That the -in parameter is processed before the -out, i.e. the csv-file is first imported, before the output csv-file is generated.



```
dbprompt -in=top_in.csv -out=top_out.csv -proj=XX
```

Figure 2:10. DbPrompt on command-line



HULL/DESI1

	A	B	C	D	E
1	HULL/DESI1	HCMWLD	HCM		
2	HULL/DESI1	STDWLD	STD		
3	HULL/DESI1	SSOWLD	SSO		
4	HULL/DESI1	COMWLD	COM		
5	HULL/DESI1	RSOWLD	RSO		
6	HULL/DESI2	MOGWLD	MOG		
7	HULL/DESI3	BLOCK	FOR		
8	HULL/DESI3	BLOCK	MID		
9	HULL/DESI3	BLOCK	AFT		
10	HULL/DESI3	BLOCK	JUMBO		
11	HULL/DESI6	HBLWLD	HBL		
12	HULL/DESI6	HBLWLD	HBL	BLOCK	BLOCK-11
13	HULL/DESI6	HBLWLD	HBL	BLOCK	BLOCK-12
14	ASSEMBLY/ASSEMDB01	MOGWLD	MOGWLD		
15	ASSEMBLY/ASSEMDB01	ASWLD	Assembly		
16	VMADMIN/REFDATA	GRIDWLD	VMT/SHIP-REFERENCES		
17	HULL/MANU1	MRAWQU	RawQuality:		
18	HULL/MANU1	MBLOCK	Block:PB1		
19	HULL/MANU1	MBLOCK	Block:PB10		

Figure 2:11. Editing the csv-file using Excel

2.6 Inithull

Inithull is used to create block data, to set up hull standards etc.

Please see the *Hull documentation* for details.