Control Simulator Innovations - CSI
Graphical User Interface (GUI)
Operations Manual

Manual Revision 1.5

Table of contents

The Graphical interface

The main window

When the application is launched, the user is presented with the following interface:

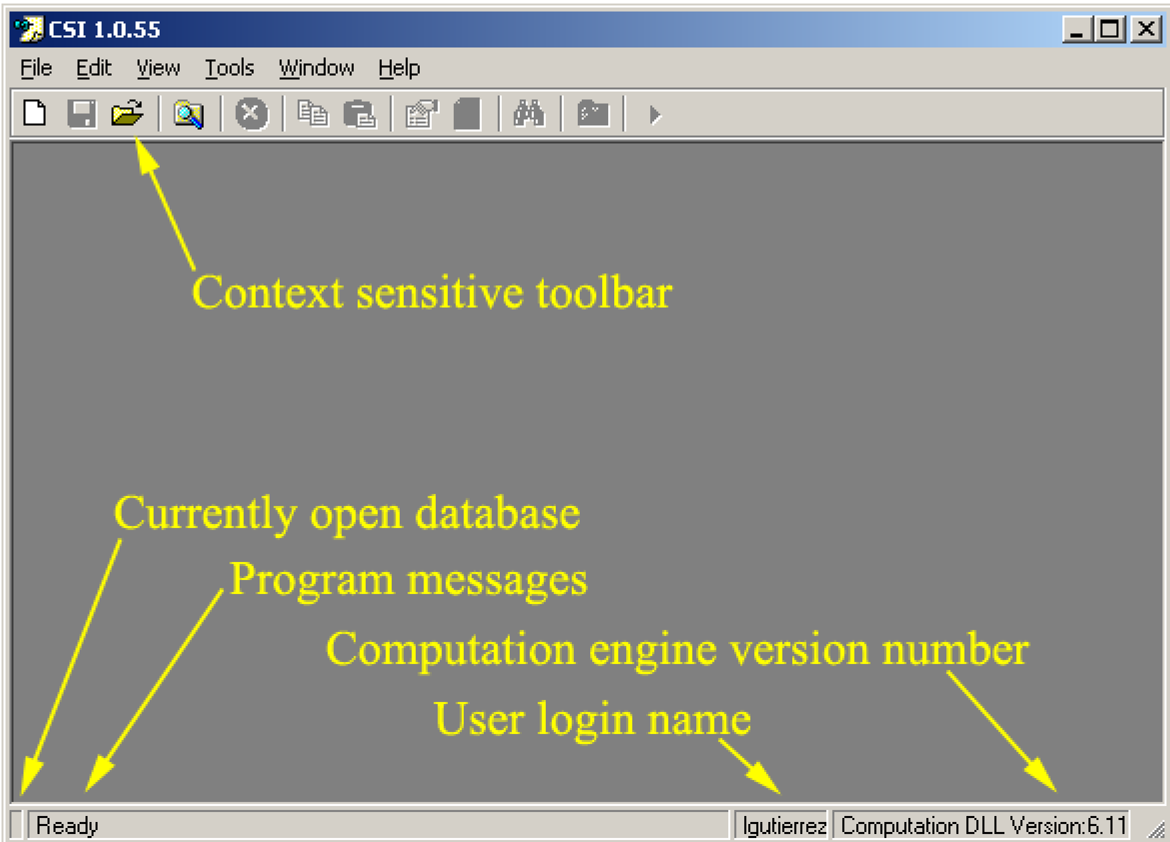


Figure 1

At the bottom of the interface, a status bar displays the name of the currently active database, the current status of the program, the Windows™ login name and the version of the computation engine DLL file.

The context sensitive toolbar

At the top of the program GUI, the context sensitive toolbar gives the user access to all of the functionality of the program. The behavior of each button changes according to the currently open window as described below.

Figure 2

Listed from left to right

- New
  - If no database is currently open, it allows the user to create a new database
  - If a database is currently open, the "Create new" dialog box appears to allow the user to create new components or assemblies
- Save
  - If pressed while the active window is an assembly schematic, the state of the currently open assembly (assembly name, comments, component locations and logical links) along with all the included components and their properties are saved.
  - If pressed while the active window is a chart, the state of the chart is saved (curve visibility status, color, pen width and line type)
- Open: allows the user to open an existing database
- Output: opens output dialog box show here:
  - This window opens the last computation-dll-generated output files using one of three different applications:
    - Default text editor
    - Microsoft Word
    - Microsoft Excel
    - Or disable a particular output file from being opened
  - The three files that can be opened from here are:
    - Header file: General analysis information consisting of all input data.
    - Summary file: Includes the header file plus all computed results.
    - Output file: Comma delimited numeric file including only the computed results and column headings.

  For an explanation and more information on these files, see Running the analysis

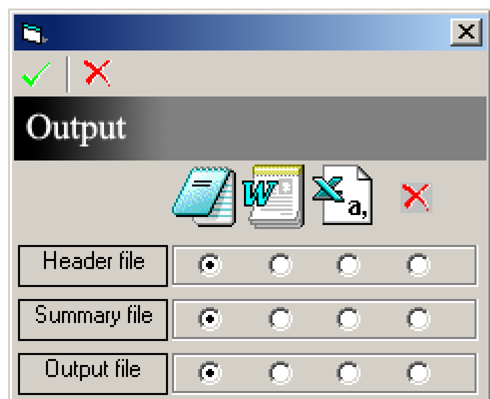


Figure 3

- Delete: Deletes the currently selected item (assembly, component, instance, logical link)
- Copy: Copies the current selection to the clipboard

  Note: CSI 6.0 uses an internal clipboard (not available to other applications) for copying and pasting component and assemblies within the application. However, if this option is used from a chart window, CSI places two things in the Windows clipboard: a windows metafile formatted (WMF) picture of the chart area plus the complete data set that generated such chart (see viewing results for more information on why and how to use these features).
- Paste: Pastes the contents of the internal clipboard. CSI uses an internal clipboard for copying and pasting components, instances and assemblies within the application. The contents of this clipboard are not accessible to other applications.
- Properties: Opens the appropriate editing window for the currently selected item.
- Reload: Replaces the information in the active window with the information currently saved in the database. This is most useful in cases where more than one user is working on the same database. Because this option reloads the information contained in the database, your current changes to the active window are lost.

- Find: Opens the component listing window where the user can apply various filters to find a particular component definition
- Go up: Opens the active window's "parent" window
  - If called from the assembly listing, component listing or assembly schematic, this option opens the main database interface.
  - If called from a chart window, this option opens the associated assembly schematic
- Analyze: Runs an analysis of the currently open assembly

The components palette

At least one database must be open for this toolbar to be visible. Drag and drop components from this toolbar to create new components/instances (see instances vs. components)

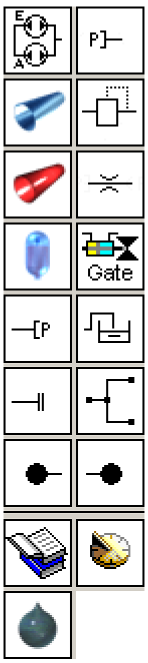
	Pump	Constant pressure inlet
	Hard tubing	Regulator
	Thermoplastic hose	Restriction
	Accumulator	Valve operator
	Constant pressure discharge	Sea chest
	Blocked	Branch
	Inlet port	Outlet port
	Document	Timing
	Fluid	

Figure 4

Components dragged from this palette can only be dropped onto the hydraulic schematic area of the assembly window, the components listing, assembly listing or database interface but not onto existing components. For a complete description of these components, see the Components reference.

Note that when you drag a component from this palette you are creating a new component having default data that you will need to change to the data you want. If the component that you want has already been defined and is in the component database, it is better to drag the component from the database rather than the components palette. See the section on adding existing components to an assembly for more details.

Working with databases

Quick Overview

A complete project, including components and assemblies covering many different analysis cases are stored in a single file called a database[1]. Database files have names of the form *.mdb.

There are two database files, Template.mdb and Units.mdb, that have special meanings that you should be aware of.

Units.mdb stores the engineering unit conversion tables. When "opening a database", never open this file. There is a special CSI main menu function for editing the unit conversion factors, should you ever feel the need to do so. Normally, you should just ignore Units.mdb.

Template.mdb is used as the starting pint for the creation of a new project database from scratch. Normally you should ignore this file. The only reason to open Template.mdb is if you want to enter your own components that you want to always be there when you create a new project. This is explained in the Creating databases section that follows. ***NOTE: If you customize Template.mdb, be sure and make a backup of your customized copy. If you install CSI again, the default template will be installed in place of the one on your computer, wiping out your work.***

A database called Demonstration.mdb is included with CSI. It contains a number of components and assemblies to illustrate working cases that are ready to run. While learning to use CSI, this is a good file to practice on, since all of the cases work "right out of the box", and you can alter them and see the result. It is a good idea to preserve a copy of the "factory" Demonstration.mdb file, so that you can easily restore it to the default state if you want to.

Finally, if you are making the transition from CS4 (the DOS version), you may want to make use of files on the CD in the CSI-CS4 Comparisons folder. There you will find CS4 versions of some of the demonstration cases allowing you to see the same system modeled in the DOS and Windows versions.

Finally, there are some restrictions on the circuits that can be modeled. See the section Allowable Assembly Schematics for more details.

Creating databases

To create a new database, do one of the following:

- Select *File* > *Create Database*
- Press *Ctrl+Ins*
- If no database is currently open, click the ***New*** icon from the toolbar.

CSI copies a template database called *Template.mdb* to create new database, therefore any component and assembly definition saved within this file will be inherited by the newly created database. This is a good place to store your commonly used components so that they will be immediately available to new projects. To do this, simply open Template.mdb as described in the next section and make your changes. Thereafter, new projects will have your changes in them when they are created.

Note: CSI cannot create a copy of an open file; therefore, the *Template.mdb* database must be closed for the Create Database command to work.

[1] It is, in fact an Microsoft Access-compatible database structure. But you should not make changes with Access. If you do, CSI may not be able to make sense of the resulting file. Do all editing from within CSI.

Opening an existing database

To open an existing database, do one of the following:

- Select *File* > *Open* database
- Drag and drop the database from Windows Explorer onto gray area of the main application window
- Click the open button on the context-sensitive toolbar
- Select it from the list of *Most-Recently-Used (MRU)* files under the *File* menu

Do not try to open Units.mdb. Do not open Template.mdb unless you are customizing it for later use in creating new projects. See the Quick overview section above for more details.

CSI verifies the validity of the selected file and if successful presents the main database interface shown in figure 5.

Database header data

The first window that a user sees after opening a database is the main database interface. An example is shown below: Usually, the name in the figure below should be the project name and the comments should be comments that apply generally to all simulation cases in the project.

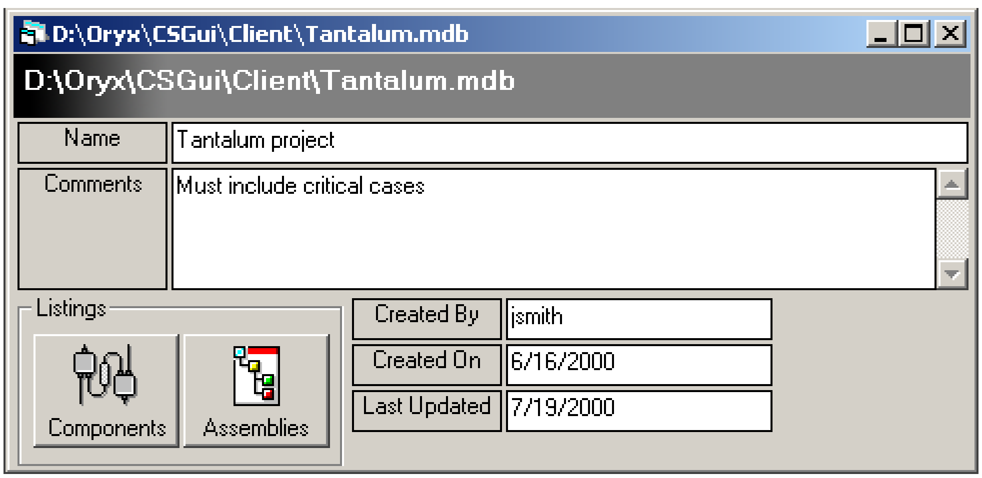


Figure 5

A descriptive name and comments can be given to a database in this window. The *Created By*, *Created On* and *Last Updated fields* are read-only. The *Created By* property is taken from the user's login name; is the user is not required to login, CSI uses login name "*user*".

Note: CSI databases are handled as documents, hence, users must explicitly save changes by clicking the Save button on the main toolbar, selecting File > Save, or pressing Ctrl+S. (If the user made any changes and attempts to close a database, CSI prompts the user if changes are to be saved.

The database interface also provides the gateway to the component listing and the assembly listing.

- Click the *Components* button to open the components listing
- Click the Assemblies button to open the assemblies listing

As mentioned above, CSI uses the *Template.mdb* database to create new databases, so if you just created a new database, you will find the same components and assemblies that are present in the *Template.mdb* database. This happens because new databases are created by creating a copy of the Template.mdb database and renaming it to the selected filename.

Working with components

Creating new components

New components can be created using any one of the following methods (they all assume you have at least one database open):

1) Drag and drop the component from the component palette onto one of the following:
   a) The database main interface
   b) The component listing window
   c) The assembly listing window
   d) The hydraulic schematic portion of an assembly window
2) Click the ***New*** icon on the context sensitive toolbar
3) Select *File* > *New*

Method 1 opens the appropriate component dialog box where component properties may be entered. Enter the desired component properties and press the green check mark. To cancel press the red cross

Figure 6

The second two methods open the Create new item dialog box. This is shown below:

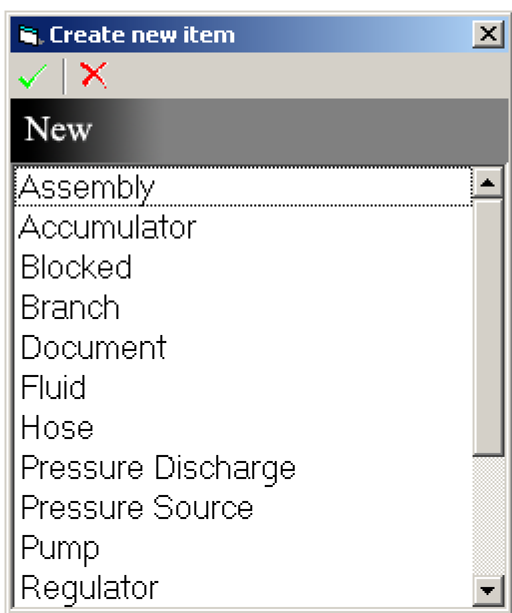


Figure 7

Select the desired component type and press the green check mark to accept. As in method 1 above, the appropriate component window opens and allows you to enter component properties. The general structure of the component dialog box is explained below:

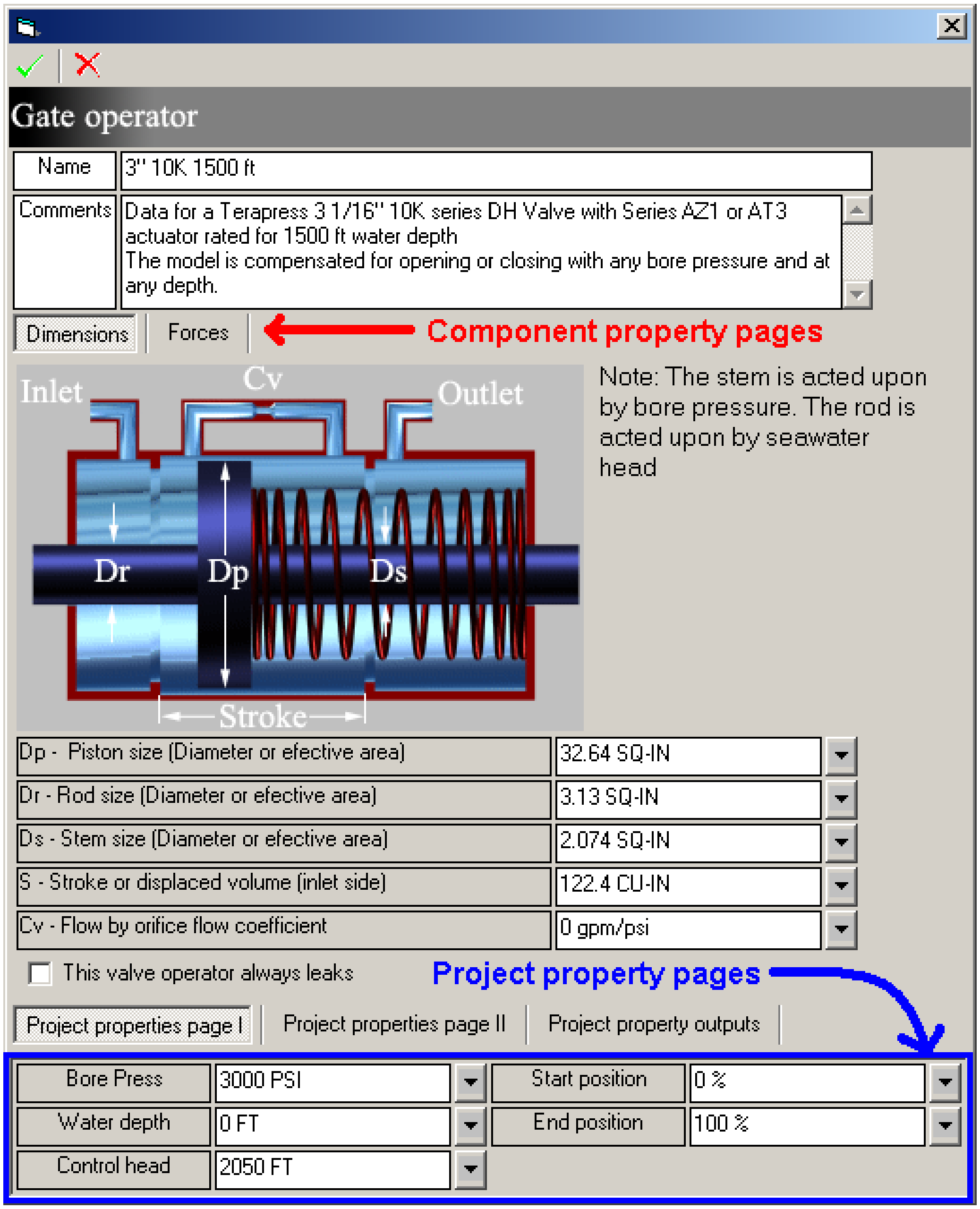


Figure 8

All components have at least a name and comments associated with them. Immediately below the name and comments fields are the component intrinsic properties' fields. Depending on the number of component properties, the dialog box may span several pages. Users may "flip" through the pages by clicking the tabs directly below the comments' box.

Components that are added to assemblies may have an extra set of properties. These are called "project properties" because they may change from project to project as opposed to intrinsic properties that are constant for a component definition.

Examples intrinsic properties are:
- The OD of a particular type of tubing
- The empty volume of an accumulator
- The bulk modulus of a fluid

Examples project properties are:
- The length of a tube
- The initial conditions of an accumulator
- The temperature of a fluid

Project properties are show below the component's intrinsic properties, and like intrinsic properties, these may span several pages. "Flip" pages by clicking the second set of tabs (if present). As a visual aid, project properties are always presented inside an embossed rectangle.

Modifying existing components

This section explains how to modify the basic definition of an existing component. To modify components within an assembly refer to working with assembly components

Open the component listing by pressing the *Components* button on the main database interface. Find the component you want to modify (see Finding components) then double-click it to open the appropriate component dialog box. Because it is the actual component definition that is being edited, only the intrinsic properties will be shown. (i.e., the project properties' area, if the component can have one, will not be visible)

Edit the desired properties and accept by clicking the green check mark. Cancel by clicking the red cross. Some component properties are required and others are optional. CSI will not accept component definitions with incomplete data. If a required field is left blank or its data is invalid, CSI will highlight the aforementioned field and prompt the user to enter valid data.

Selecting component property units.

Users may select an array of different units to specify component properties. The typical component property field is depicted below:

Figure 9

The component property field is composed and behaves as follows:

- Field name: Shows the name of the property to be edited. A short explanation of the property is shown as a tool-tip text (place the mouse over the field name to view the tip)
- Field value: The property value is entered here. This text box behaves as follows:
  - When first activated (via a mouse click or the tab key, for instance), the numeric value gets automatically selected (if separated from the units by a space)
  - Double-click it to select the complete entry
  - Numeric values must be separated from units by at least one space
  - Entries are NOT case sensitive (2000 psi is the same as 2000 PSI)
- Unit conversion button: When clicked, CSI displays a popup menu with the property value converted to all other defined units (if they can be converted). Otherwise it displays a list of the available units for the property with a numeric value of zero. An entry may be selected directly from the menu or the menu may be displayed for reference only. Some properties accept entries of different types such as an outer diameter or an area. CSI will not attempt to convert values between such dissimilar types.

For more information about the units used by CSI see Working with the units database

Deleting existing components

This section explains how to delete existing component definitions. To modify components within an assembly refer to editing assembly components

Open the component listing by pressing the *Components* button on the main database interface. Find (see Finding components) and select the component(s) you want to delete and simply press the delete key or click the delete button on the main toolbar.

Duplicating components

CSI provides different ways to duplicate components. This feature becomes very useful if a component similar to an existing one must be defined. The user can duplicate the component then simply modify the desired properties. Proceed as follows:

- Find the desired component in the components' listing window (for more information see finding components) and select it.

Right click on the desired component and select *Duplicate* on the resulting popup menu. CSI creates a copy of the selected component called *Copy of ComponentName* where *ComponentName* is the name of the original component.

Finding components

From the database main interface, press the *Components* button. This will open the components listing window show below:

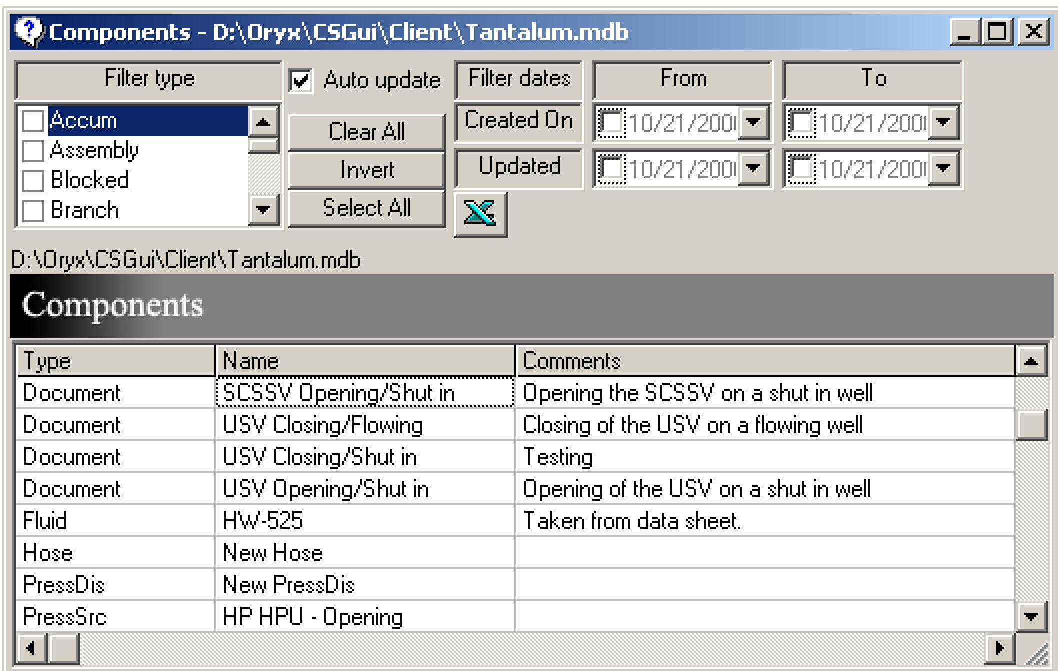


Figure 10

This interface has the following controls:

- Filter type: Filters the component list by type of component
- Auto update: This is useful with large databases. If this is deselected, the user is required to select the “Reload” button for the filters to be applied; otherwise, the list is updated every time the filter selections change. This option is selected by default.
- Clear all: Clears all the component *Type* filters so that all components are displayed
- Invert: Inverts the component *Type* filter selection
- Select all: Applies all of the component *Type* filters. Since not selecting any *Type* filter is the same not filtering at all, “Select All” has the same effect as selecting none.
- Filter dates: Use this filter to further narrow the show list by component creation or modification date.

- Send selection to Excel (icon shown below): This option copies the selected cells to an Excel sheet.

Figure 11

Working with assemblies

Creating a new assembly

Open the database where the new assembly is to be created then do one of the following:

- Click the ***New*** button on the main toolbar
- Select *File* > *New* (or press *Ctrl+N*)

The *Create new item* window pops up (see figure 7). Select assembly and click the green check. A new assembly schematic like the one shown below opens.

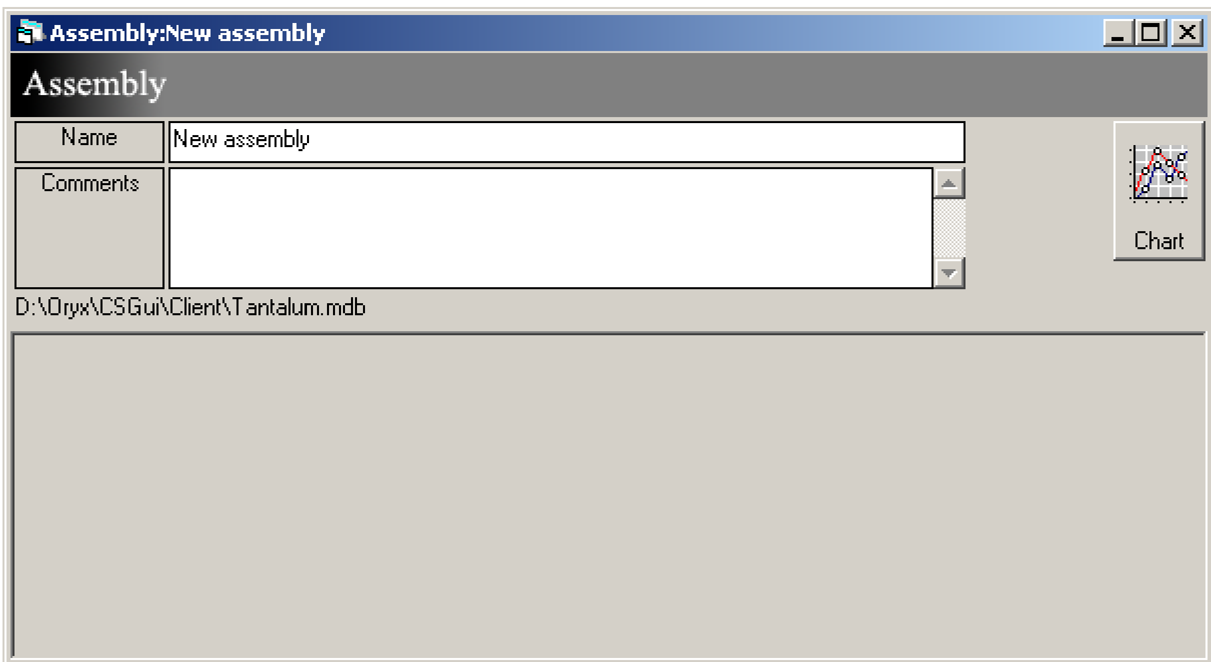


Figure 12

Enter a name and comments as desired.

Note: CSI was designed to handle assemblies as documents (see earlier note on this), therefore, users must click the save button for the changes to be written to the database.

The embossed rectangle is the hydraulic schematic area where components and subassemblies may be entered to define a hydraulic circuit. The button labeled *Chart* opens the charting application for the current assembly.

The blank area immediately to the left of the chart button is used for special components that are required if an assembly is to be analyzed by the computation engine. (See Analyzing assemblies)

Opening existing assemblies

To open an existing assembly, first open the assembly listing by pressing the Assemblies button on the main database interface. The assembly-listing window is shown below:

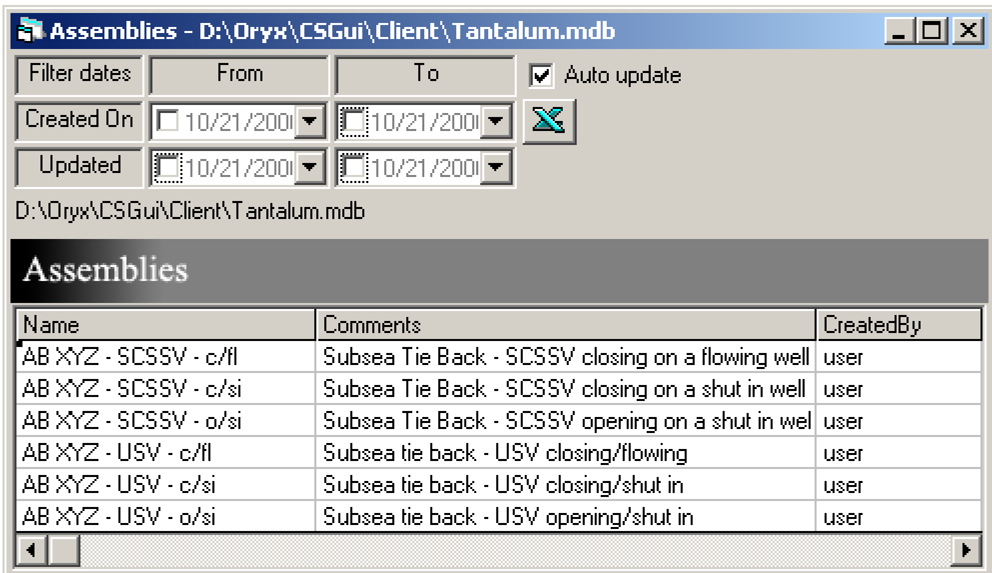


Figure 13

The assemblies' list may be filtered by creation date or last modification date. Simply double-click the assembly row to open the assembly.

Adding new components to assemblies

To add a new component to an assembly, drag and drop the desired component icon from the component palette onto the hydraulic schematic area of the assembly. CSI creates a new blank component and opens the appropriate dialog box

Adding existing components to assemblies

Component definitions can be added to an assembly using one of the two methods described below:

- Find (see finding components) and select the component definitions you want to include in the assembly then select the *Copy* option under the *Edit* menu or press *Ctrl+C*. Open the target assembly window (see opening assemblies). Click on the gray area of the hydraulic schematic to reset the current selection if required then select Paste under the Edit menu or press *Ctrl+V*. The selected components will be added to the assembly.

- Open the target assembly window. Find and select the component definitions you want to include in the assembly. Notice the small black dot that appears to the top right of the leftmost-topmost selected cell. Manage both the assembly window and the component list window such that both the hydraulic schematic area and the aforementioned dot are visible on the screen. Place the mouse pointer directly on top of this black dot. The mouse will change shape as depicted on the figure below:

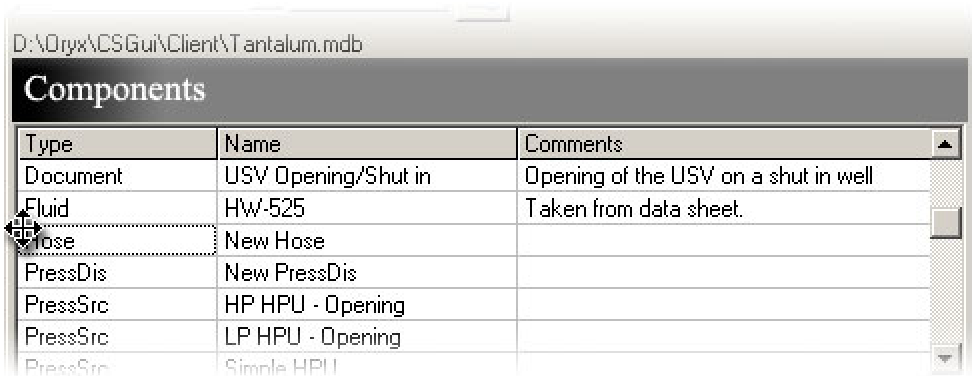


Figure 14

Click and drag the black dot (this is called the drag-drop handle) from the component list window onto the hydraulic are of the assembly window. While dragging, the mouse will change to according to the following rules:

- If only one component is being dragged, it will show the type of component
- If more than one component is being dragged, it will show the “sub-assembly” icon.

On systems with limited screen size where placing two screens side by side is difficult it becomes useful to start the drag-n-drop operation with the component list maximized. Having started the drag-n-drop operation, repeatedly press *Alt+Tab* until the desired assembly activates. The dragged selection can now be dropped onto the hydraulic schematic area of the assembly window.

Note: CSI uses three special types of components (fluid, document, and timing) in the assemblies that are to be analyzed (see running the analysis). Although these components are displayed by CSI in the gray area between the Name field and the Chart button, they too must be dropped onto the hydraulic schematic area of an assembly.

Adding components to an assembly from within.

Components may also be added from within the assembly. This is a useful feature if a component exists somewhere in the active assembly and must be used somewhere else within the same assembly. Instead of drag-n-dropping it or copy-n-pasting it from the component listing and re-entering project properties, the steps described below can be followed.

- Open the assembly where the component is to be duplicated
- Select the component or components that are to be duplicated (see selecting components/ assemblies/ logical links)
- While holding both the control and shift keys down (*Ctrl+Shft*) click and drag the selected component(s) onto the gray area of the hydraulic schematic.

The following window appears:

Figure 15

Select whether you want to create copies or instances of the selected components (see next section).

An alternative method is:

- Open the assembly where the component is to be duplicated
- Select the component or components that are to be duplicated.
- Select the *Copy* option under the *Edit* menu
- Click anywhere on the gray area of the hydraulic schematic to reset the selection
- Select the *Paste* option under the *Edit* menu

The window shown above appears. Select whether you want to create copies or instances of the selected components.

The next section provides a detailed explanation of the differences between copies and instances.

Creating component copies vs. instances

In order to describe the difference between creating copies and creating instances, the definition of an *instance* must be understood.

An instance is the collection of intrinsic properties of a component definition plus the collection of project related properties of a component, if any. Any time a component is added to an assembly from the components' palette or the components listing, an instance of that component is being created and project properties are associated with it. If the same component definition is added from the components' listing, an additional instance of the same component is created, but both instances reference the same component definition. If an intrinsic property of the component definition is modified, the property will be modified in every instance in every assembly within the same database.

For example, a tubing component definition is created with an outer diameter of 0.375 in and wall thickness of 0.049 in. The component is saved and instances of it are inserted in assemblies A, B and C with the following lines lengths assigned: 5 mi., 7 mi. and 4.5 mi. The tubing component definition is then opened from the components listing and its wall thickness is changed to 0.05 in. and saved. The next time assembly A, B, or C is opened, the tubing component will reflect the new wall thickness of 0.05 in. The change of the wall thickness in the component list caused a change in all instances of the component.

In the picture below, the inner rectangles represent a tubing component definition (not all properties are shown) that has been inserted in one or more assemblies to create instances of said component.

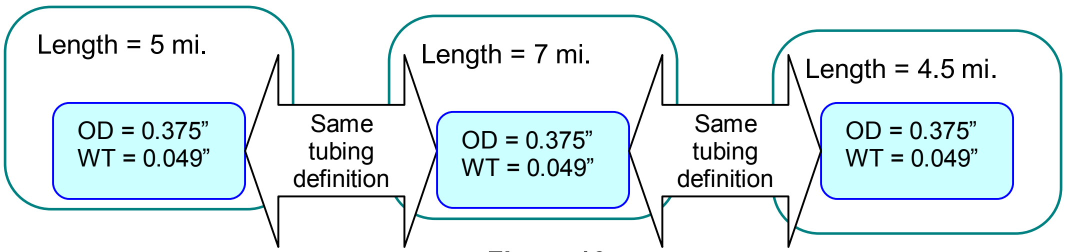
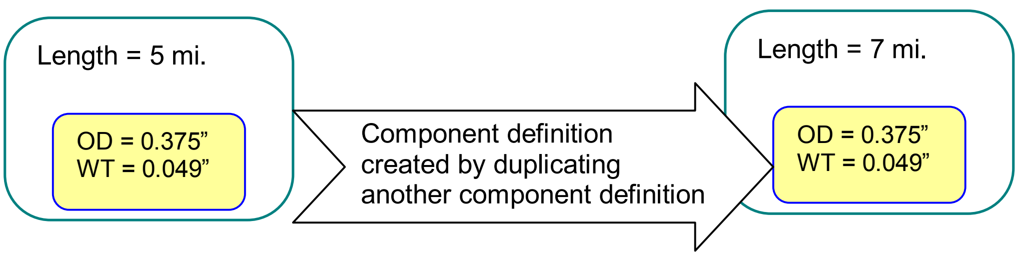


Figure 16

If a duplicate component is created however (see duplicating components), a completely new component definition (detached from the original definition) will be created in the component list. Changing, an intrinsic property in one component will not change the properties in the other because they are separate components. In the picture below, the inner rectangle on the left represents a component definition that has been duplicated to create the component definition represented by the inner rectangle on the right. Changing intrinsic properties in the first component will not change intrinsic properties on the second component.



When a component definition is duplicated, a ***copy*** of the component is being created.

CSI detachment of intrinsic properties (directly associated with the component definition) from project properties (associated with an instance of a component definition) allows for an effective way to keep component definitions up to date.

Replacing assembly components

The Control Simulator GUI provides a way to quickly modify assembly configurations by dragging and dropping (or copying and pasting) components onto existing components. This feature is most useful when a single component or subassembly is to be replaced with another component or subassembly, but the user wants to preserve some properties. For example, consider a case where several hydraulic hoses are to be evaluated for a certain application. The user can create an assembly with all the logical links in place (for an explanation of logical links and how to set them up see Building the hydraulic circuit) and enter all the appropriate properties for all the hydraulic components, including the first hose being analyzed. The user can them duplicate the assembly (for an explanation of assembly duplication see Duplicating assemblies) instead of creating the second analysis case from scratch. Then rather than removing the original hose component, inserting the new one and reconnecting with logical links, the user can simply drag-n-drop the next hose component from the components listing onto the old hose component in the duplicate assembly. The hose component will inherit all the project properties (such as length) of the original hose component, but the instance will reference the newly dropped hose component. The process is depicted below:

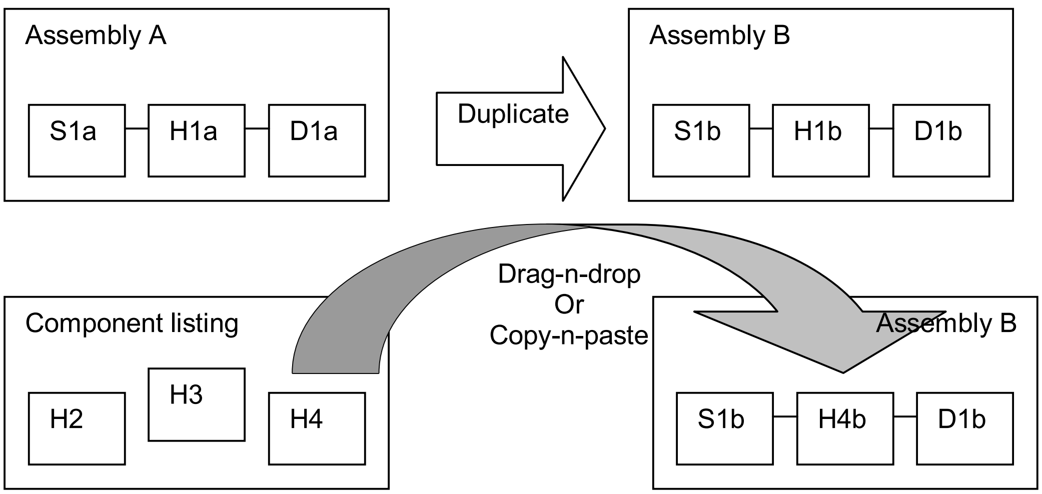


Figure 17

The hose component H4 would replace the component referred to by H1b, effectively creating instance H4b, but the project properties (such as the length of the hose) of instance H1b would be inherited by instance H4b (for an explanation of instances see Creating instances vs. copies)

Note: To replace a component using the copy-n-paste procedure, first select the component that will replace the target component, then choose *Copy* under the *Edit* menu (or press *Ctrl+C*), then open the assembly where the target component is and select the component to replace. Now select *Paste* under the *Edit* menu.

Selecting components/subassemblies/logical links

In order to manipulate components within assemblies, the user must first make selections as appropriate. This section describes how components, sub-assemblies and logical links are selected in CSI. Because this section applies to components, sub-assemblies and logical links within an assembly, they all will be referred to as *objects*. Selections may contain one or more objects.

Click an object to select it
Click an object while depressing the *Ctrl* key to add it to the current selection
Click an object while depressing the *Alt* key to remove from the current selection

CSI shows selected components and subassemblies with embossed icons. Logical links turn white when selected. The picture below shows a selected valve operator component.

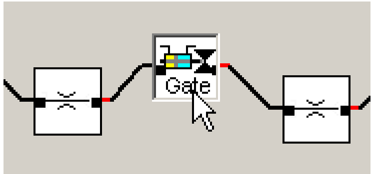


Figure 18

The figure below shows a selected logical link. (For more information on logical links, see Building the hydraulic circuit)

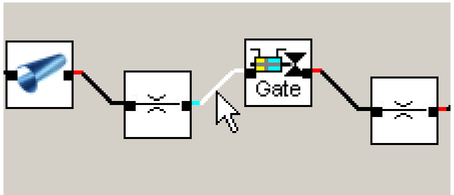


Figure 19

Users can create selections in CSI by dragging a selection rectangle (selection rectangles do not apply to logical links). Simply click and drag with the left mouse button anywhere on the hydraulic schematic to begin a selection rectangle. Any component fully enclosed by the selection rectangle will be selected or deselected as explained below:

- Click and drag resets the current selection and creates a new one. CSI displays the selection rectangle in black.
- Click and drag while depressing the *Ctrl* key adds the enclosed components to the current selection. CSI displays the selection rectangle in blue.
- Click and drag while depressing the *Alt* key removes the enclosed components from the current selection. CSI displays the selection rectangle in red.

In the image below, CSI will reset the current selection (the valve operator) and select the three components enclosed by the selection rectangle. It will select the valve operator and both restrictions at either side of the valve operator, but not the tubing component because the selection rectangle does not fully enclose it.

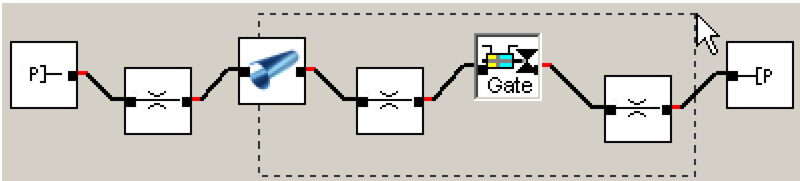


Figure 20

Editing assembly components.

To edit properties of any component within an assembly, simply double click the component icon and enter properties in the resulting dialog box. Some components have project dependent properties, that is, properties that are not intrinsic to the component but potentially change from project to project such as the length of a hose of the precharge of an accumulator. In these cases, CSI requires the user to enter valid values for such properties.

Specifying component output options.

CSI allows the user to specify which analysis variables to output. An output specification is always treated as a component project property. The image below shows the typical arrangement of an output specification control

Figure 21

- Include: The check box controls whether the variable is to be written to the output file or not. The variable will be written only if the box is checked.
- Unit spec. Either type the unit to use for output or select the unit from the pop up menu that appears when the unit selector button (next to the units' spec) is pressed.
- Precision: Select the output precision with the pull-down list immediate to the right of the unit selector button.

Building the hydraulic circuit

The hydraulic circuit is defined by creating logical links between the assembly components (and sub-assemblies if applicable. For more information see using assemblies as building blocks). To create the logical links follow the steps below:

- Set CSI to "linking-mode" by clicking the "Linking" button, just under the fluid icon in the upper right of the assembly window. When it says "Linking: Active" you may draw links as follows..
- Click near the desired component's output port and drag while the linking button says Linking: Active. CSI will find the closest output port and will create a "floating link" like the one by the mouse cursor shown below:

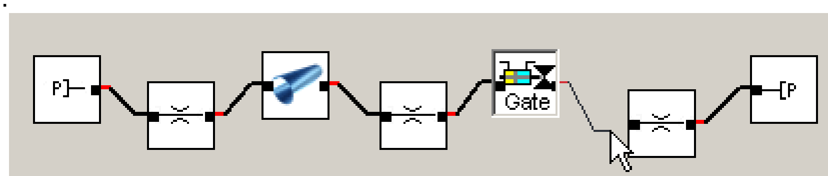


Figure 22

- Continue dragging until the mouse is over the desired inlet port of the downstream component then click the Linking button again to change it to "Linking: Inactive".
- NOTE: older versions of CSI do not have a Linking button. You had to hold the shift key down to draw the links. That method of drawing links no longer works.

Logical links can only be created in downstream direction, i.e. beginning at an outlet port and ending on an inlet port. Outlet and inlet ports are to the right and left of the icon respectively. CSI shows the upstream portion of a logical link in red.

Copying assemblies across different databases.

The steps for copying assemblies across databases are listed below:

- Open the source database assembly listing
- Open the target database assembly listing.
- Select the assembly to be copied on the source assembly listing
- Place the mouse over the drag-n-drop handle (the black dot at the top-left of the active cell), then click and drag with the left mouse button.

- Drop the selected assembly onto the target assembly listing. CSI will copy the selected assembly along with all the dependent components. If the selected assembly included subassemblies, the entire assembly is “flattened-out”; that is, the components in the subassembly are brought into the same level as the top-level assembly.

Duplicating assemblies.

CSI provides a way to duplicate assemblies. This feature becomes very useful in the following cases:

- Different scenarios/cases are to be analyzed for a single assembly.
- The user need to create an assembly that closely resembles an assembly that was previously defined in CSI

To duplicate an assembly, simply open the assembly listing, find the assembly you want duplicate and select it with a single click (double-clicking the assembly will cause CSI to open it). Right-click the selected assembly and select “duplicate” in the resulting pop up menu. The following dialog box appears:

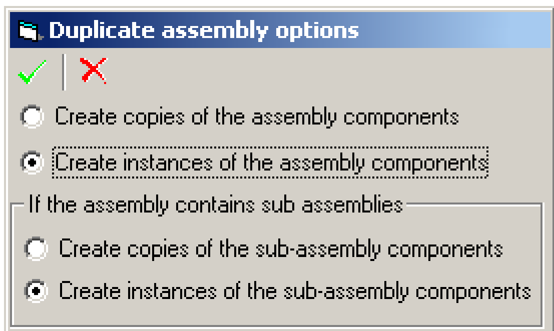


Figure 23

Select whether you want to create copies or instances of the assembly components (and sub-assembly components if present) and click the check mark. If in doubt, create instances. For more information on choosing component copies versus instances see Creating component copies vs. instances. For more information on subassemblies see Using component subassemblies to create complex assemblies.

Allowable Assembly Schematics

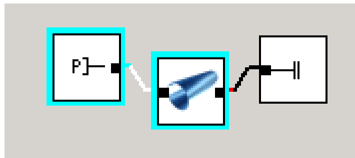
The assembly editor will allow you to construct almost any equipment arrangement you can think of. However, the current computation engine (based on earlier versions of The Control Simulator) is restricted to certain predefined component arrangements. These restrictions will eventually be removed, but, for now, you have to deal with them.

At the highest level, allowable arrangements must be of the form **source -> line -> discharge.** Discharges may be followed by additional **line -> discharge** pairs for up to eight lines. A line is simply tubing or a hose. The source can be thought of as the equipment located topside. The discharge is the equipment located subsea. The source and discharge need further clarification.

Fixing Errors

Sources and discharges must be composed of components connected in certain predefined ways. The following figures provide maps showing which components may follow other components. In each figure, you must start with a component on the left edge of the figure (Having no line to the left of the box) and proceed to the right along the lines until you reach a component having no line to the right.

Figure 24 – A structural error in an assembly



If you construct an arrangement that is not allowed, the assembly editor will give you an error message and a warning that the structure is not that of a valid assembly. It also highlights the components that are in error. In the example in Figure 25, a pressure source is followed directly by tubing. If you look at Figure 26, you will see that a pressure source must be followed by a restriction, which may then be followed by tubing.

To fix the error in Figure 25, you would delete the link between the pressure source and the line (click on the link to select it as in figure 19, and then press the delete key). Then you would add a restriction to the assembly. Finally, you would connect the pressure source to the restriction and then the restriction to the line.

Source Map

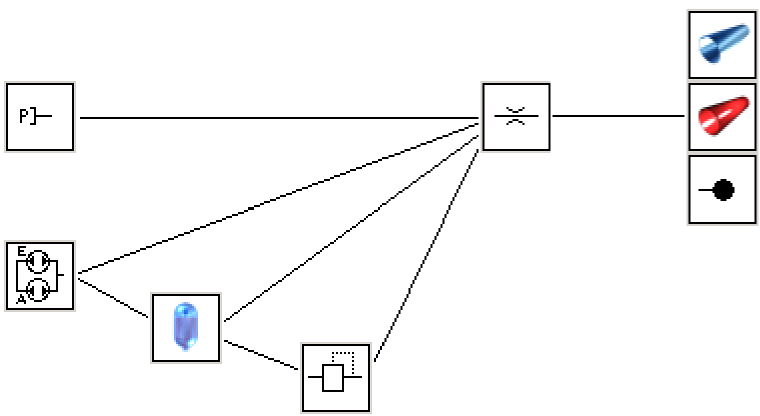


Figure 25 - Map of allowable component connections at the source (topside equipment)

Discharge Map

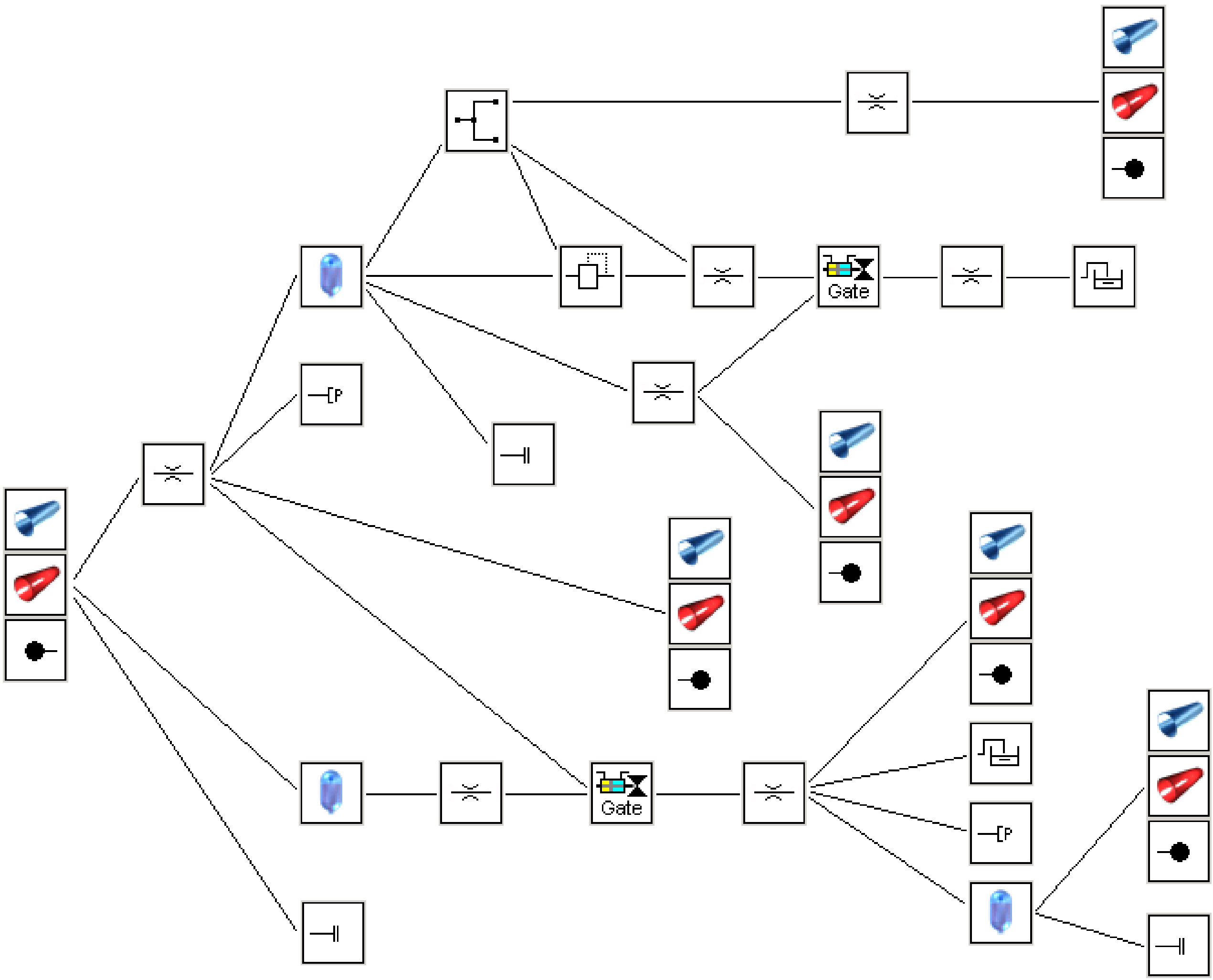


Figure 26 - Map of allowed component connections in the discharge (subsea equipment). Note that if you include the branch component, both outlet ports must be complete.

Analyzing Assemblies

Running the analysis

There are certain requirements that an assembly must meet in order to be analyzed by CSI. These requirements are listed below:

1. The top-level assembly must contain one *Document* component, one *Fluid* component and one *Timing* component.
2. The top-level assembly must contain at least one component in the hydraulic schematic (the computation engine DLL requires at least a source connected to a line and discharge).
3. The hydraulic circuit must not loop back on itself, that is, a component's outlet port cannot connect to a component upstream from it
4. The hydraulic schematic must conform to one of the mathematical models supported by the computation engine DLL. (For more information, refer to the *Valid Assemblies section*)
5. Only one hydraulic circuit may be included in the assembly schematic.
6. All components must have all their properties properly initialized.

To run an analysis, open the top-level assembly and press the *Play* button on the main toolbar (a little triangle pointing to the right). CSI runs a series of tests to ensure that requirements 1, 2, 3 & 5 above are met. Requirement 6 is checked during the definition of the components themselves. Because the supported mathematical models will vary with versions of the computation engine DLL, it is the computation engine DLL that checks for the validity of the hydraulic circuit. If the complete hydraulic circuit falls within the definition of those supported by the current version of the computation engine, the circuit is analyzed and the results written to the following three files:

- *Header file*: Analysis information file (your input data only) as read by the computation DLL. This file is best suited for creating reports with Microsoft Word.
- *Summary file*: Includes the header file plus results tables (computed output values). This file is best suited for creating reports with Microsoft Word or viewing with Notepad.
- *Output file*: Comma delimited numeric file. This file is best suited for creating charts in Microsoft Excel or other spreadsheet applications.

The files listed above are temporary files that are written into the folder along with the program executable file, CSI.EXE. These files are overwritten each time an analysis is run. If you want to save the results, it is usually best to save the file from Word or Excel.

In addition a more permanent file named *asemblyname*_summary.txt is written into the folder that contains the database file, *databasename*.mdb. This file is only overwritten when an analysis is run on the same assembly again. It is identical in content to the Summary file described above.

Viewing analysis results.

After a successful analysis run, CSI automatically displays the Output dialog box (see Figure 3) where the user can choose from three different applications for viewing the results files. The user may choose not to open an external application by selecting the options under the red cross. In this case the built-in charting utility can be used to view and chart the results. The charting utility is described in the next section.

Using the charting utility

To access the built-in charting utility in CSI, open the assembly for which results are to be charted and press the *Chart* button. This will open a window like the one shown below:

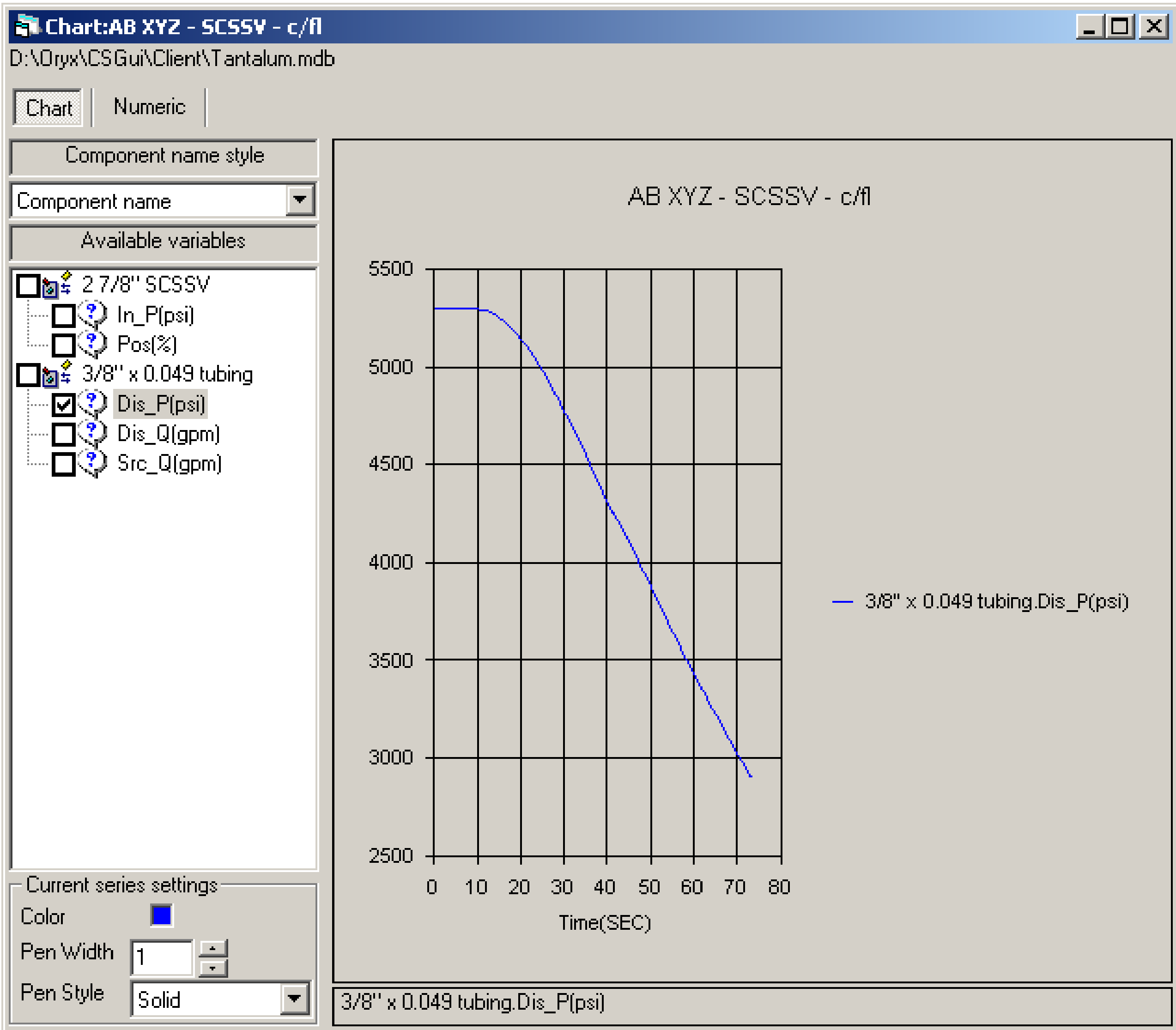


Figure 27

The top of the window shows the database where the applicable assembly is saved. Immediately below the filename, two tabs labeled *Chart* and *Numeric* allow the user to access the two views in the charting utility.

The chart view

The chart view allows the user to create charts with the available curves for the selected assembly. (To see how to make component output available to the charting utility see Specifying component output options)

The chart view has the following controls:

- Component name style: Allows the user to display the component name (both on the chart legend and the variables hierarchy tree) in one of the following formats:
  - Component name: This is the name the user gave the component via the appropriate component dialog box
  - Sequence name: This is name CSI gave the component while analysis was taken place. It is in the form *Type_n* where *Type* is the component type identifier (accum, tubing, etc.) and *n* is a sequence number.
  - Component + sequence: A concatenation of the two names explained above

- Available variables: This displays the variables available to the charting utility for the selected assembly. Checking the boxes to the left of the variable name causes will make the associated curves to appear on the chart. Checking the boxes to the left of a component icon will cause all the variables associated with said component to appear on the chart.

- Current series setting: Allows the user to modify the properties of the active chart curve. To activate a curve, either click on it directly on the chart (if visible), or select it on the *Available variables* hierarchy tree. The name of the active curve will appear within the box just below the chart area.
  - Color: Click the colored box to change the color for the active curve.
  - Pen width: Modifies the weight (width) of the active curve
  - Pen style: Modifies the style of the active curve (solid, dotted, etc.)

  To view the value of an individual point on any visible curve, first make the curve active as described above, and then click the desired point (double-clicking the point will not work, it's one click to activate the curve, wait a bit, the click to activate a point). The abscissa and ordinate values will show within the box just below the charting area.

- Chart: This area shows the chart itself. The title of the chart is the same as the name of the assembly. The legend always displays the chart names as *ComponentName.Curve*. Because the axes always expand and contract to display the full numerical range of all the selected curves, displaying multiple curves with widely differing numerical ranges will make one of them invisible or hard to read. For example, selecting a pressure curve with range 0-2000 psi together with a flow curve with range 0-5 gpm will display the pressure curve correctly but the Y-axis will show the flow curve as a nearly flat line near the X-axis.

The numeric view

Use this view to either browse the numeric data or to send it to Microsoft Excel as explained in the *Advanced topics* section of this manual.

Advanced topics

Creating a library of components

Because copying component definitions and assemblies across databases is a simple drag-n-drop or copy-n-paste operation, creating and using a library of components and assemblies in CSI is a simple task. A library of components is a regular Control Simulator database that is designated as a central repository for component definitions and assemblies that are commonly used by a user or a group of users.

To create a library of components, simply create a new CSI database and save it to a location that is readily available to all the users that need to have access to the component library, such as a shared network directory. Then add components just as you would for a project database.

To use a component definition, open both the library database and the project database and copy component definitions by dragging-n-dropping components and assemblies from the library to the project library as needed. (For more information on copying component and assembly definitions across databases see Working with assemblies)

Using component subassemblies to create complex assemblies

CSI provides a method for grouping components as subassemblies that can be inserted into assemblies for creating higher order assemblies. The technique explained here can also be used to create sequences of components that are considered “standard” in your organization. For example: If Pump XYZ can be represented with a pressure source component followed by a restriction component, you can group the two components into a subassembly and later insert the subassembly into another assembly rather than having to enter the insert the individual components again creating logical links between them.

Exporting analysis results to Microsoft Office

CSI provides various methods for exporting data to the Microsoft Office suite for further analysis and report generation. These are explained below:

- Press the *output* button (see the context-sensitive toolbar) on the main toolbar. The resulting dialog allows the user to send any or all of the three files that are generated during an assembly analysis. Simply select the desired application to send the file to. Both header and summary file are better suited for Microsoft Word whereas the output file is better suited for Microsoft Excel.

Note: The header, summary and output files are temporary files that get rewritten every time an analysis is run; hence, this option applies only to the assembly analyzed last.

- Open the assembly for which results are to be exported to and press the *Chart* button. The chart window opens. Click the *Numeric* tab and press the export to Excel button.

- Open the assembly for which results are to be exported to and press the *Chart* button. The chart window opens. Change chart settings as desired (pen color, pen width, etc.) and select the Copy option under the edit menu or press *Ctrl+C*. This operation places two sets of data onto the Windows clipboard: The chart image as a windows metafile and the numeric data that generated said chart. Open the target document in the desired application (Word or Excel) and select paste. The numeric data gets pasted by default. To paste the windows metafile image, select paste special and select image in the resulting dialog box. The format in which this operation copies data to Excel varies from that in the first and second option in that the former creates one column for time information and one column for each of the existing parameters and the latter creates a series of time-parameter columns.

Table as output by option 1:

Time	*Param1*	*Param2*	*Paramn*

Table as output by option 2:

Time	*Param1*	*Time*	*Param2*	*Time*	*Paramn*

XML output

XML stands for extensible markup language. XML is designed 'to make it easy and straightforward to use SGML on the Web: easy to define document types, easy to author and manage SGML-defined documents, and easy to transmit and share them across the Web.'

Just prior to calling the computation engine DLL, CSI generates an XML formatted file that shows the hierarchy of the total assembly as it is to be treated during analysis. This file can be viewed using an Internet browser such as Microsoft Internet Explorer or Netscape Navigator. The file will show the dependency of components (logical links and branching) of all the assembly components.

A full explanation of XML is beyond the scope of this manual. For more information about XML, visit http://www.w3.org/TR/REC-xml

Working with the units database

Even though CSI comes with a multitude of predefined units for specifying component properties and computation output, the user may want to expand or modify the units' database used by the Control Simulator. All of the unit definitions are saved in the Units.mdb database.

CSI provides a user-friendly interface for working with the units.mdb database. This interface can be accessed by selecting the *Units editor* option under the Tools menu.

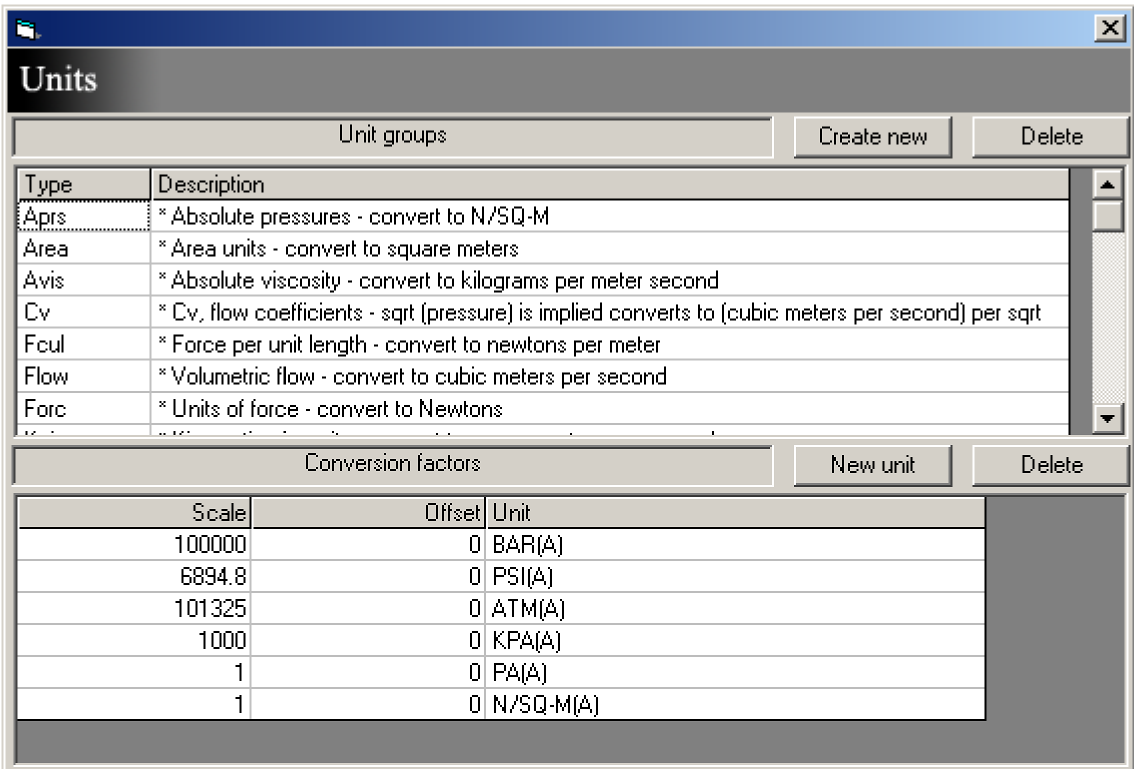


Figure 28

The top portion of the dialog shows the unit groups. Selecting a row on this grid will cause the applicable conversion factors to show on the grid below. CSI uses the displayed scale and factor values to convert values to and from the MKS system (CSI uses the MKS system for internal computations) as per the formulae show below:

- MKS Unit = Scale * (User unit + Offset)
- User Unit = (MKS unit / Scale) – Offset

Working with unit groups

Although CSI does yet not allow users to specify which of types of units a component's property can accept, CSI provides a way to create new unit groups as framework for future enhancements. Open the unit group editor by pressing the *Create new* button or by double-clicking a unit group line. The unit group definition window is shown below:

Figure 29

Simply enter a 4-character alphanumeric type identifier and a description and select the check mark. CSI will check for identifier uniqueness. The description must begin with * and should identify the type of unit and the MKS unit that is used internally.

Note: Component property unit types are hard-coded in CSI, therefore, unit group identifiers (type) should never be modified. The mechanism for modifying identifiers is presented here for future compatibility only.

Working with unit definitions

This is likely to be the only portion of the units database interface to be normally used. To create or modify a unit definition, first select the unit group in the units groups list, then press new unit button or double-click an existing unit definition. The unit definition dialog box window appears:

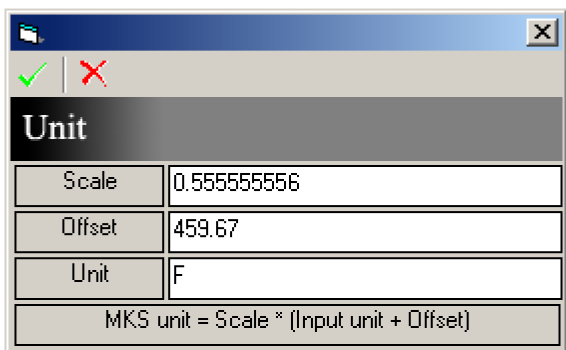


Figure 30

Users must enter a scale factor, an offset and a unit identifier. The dialog also shows the formula that CSI uses to convert numeric values to and from the MKS unit system. For example, the internal MKS unit for temperature is K (Kelvin). If you input 40 F, it is converted as follows:

K = scale * (40 + offset)
277.594 = 0.55555555556 * (40 + 459.67)

NOTE: It is assumed that the existing unit conversions are correct. If you find an error, please do not just correct it. Please let us know about the error! Also, if you think a conversion is wrong, check it out very carefully before changing it. They have been checked repeatedly.

Component reference

This section describes the properties and the default values of all the hydraulic components. For each component we show the properties screen as it appears to the operator, and a table showing the internal representation of those same properties. Some properties in the table are only visible to the computation engine and cannot be modified directly through the graphical user interface (for instance, adding a line to the comments' section modifies the QtyComments property). The tables are used to provide explanations of selected properties.

The properties shown in gray in the tables are called project properties and can only be edited within the context of an assembly (they are not intrinsic component properties). Please note that project properties do not have default values and that the following tables show project property values only for the purpose of exemplification.

The accumulator component

The accumulator component dialog box is show below:

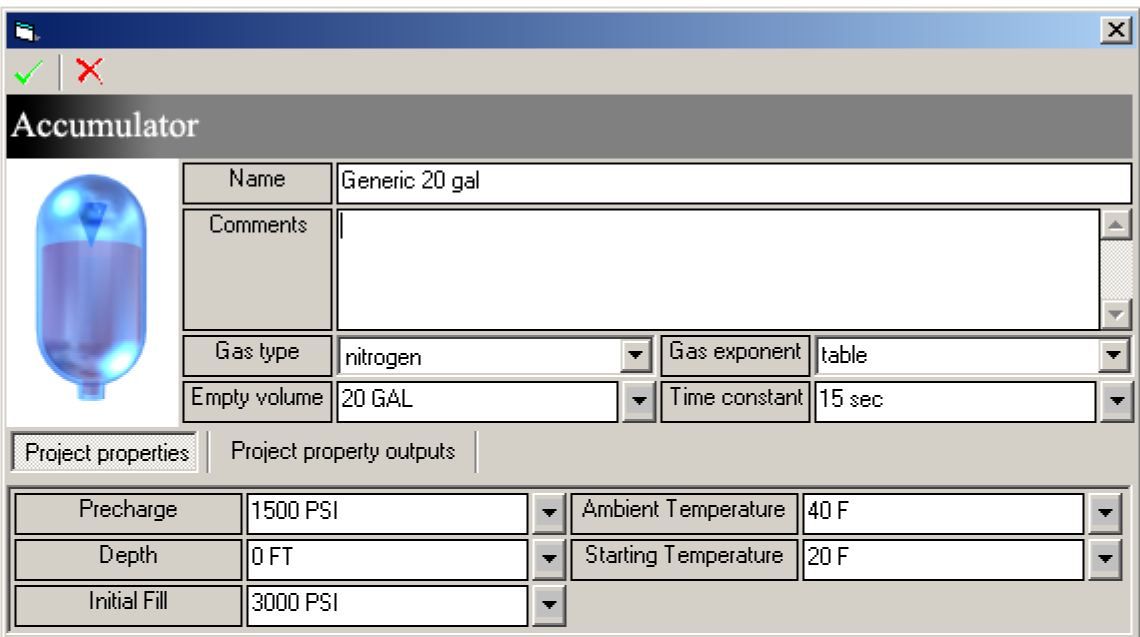


Figure 31

Property	Description
[accum_1]	Unique identifier for this component
Cn=""	Comment line where *n* is an integer >=1
QtyComments=n	
Type_Of_Gas=nitrogen	Gas can be nitrogen, helium or ideal. Note: above about 5000 psia, “ideal” is seriously in error when compared to the real gas.
Gas_Exponent=table	This can be “table” or a real number. For the most accurate results use “table”
Empty_Volume=10 gal	Gas volume when the accumulator is empty of fluid. This is the actual volume of gas present when the accumulator is precharged, which is usually different from the “nominal” volume of the bottle.
Time_Constant=45 sec	This parameter is a measure of the rate at which the temperature of the gas approaches the ambient temperature. 45 seconds has proven to be accurate for cylindrical float and piston type accumulators, with either helium or nitrogen. In the absence of better data, use 45 seconds.
Precharge=1500 psi	This is the precharge pressure at the ambient temperature. If an absolute pressure unit like psi(a) is not used, then the hydraulic fluid depth head is added to get the absolute pressure.
Depth=1000 ft	Depth below the HPU
Ambient_T=40 F	The gas always tries to settle back to this temperature.
Starting_T=20 F	Temperature of the precharge gas at the start of the simulation
Initial_Fill=3000 psi	This the pressure (or volume) of fluid in the accumulator at the start of the simulation. *Important:* If you want to start with an empty accumulator, do not use a pressure here… use 0 gal or some other volume unit. If you specify a pressure, round off errors may result in a tiny squirt of fluid that you do not expect.
QtyOutputs=4	
O1=Gas_P,y,psi,-1	(Output) Internal pressure. May be absolute or gauge (relative to fluid hydraulic head).
O2=Gas_T,n,F,-1	(Output) Gas average temperature as a function of time. “Average” means over the internal volume at a particular point in time.
O3=Gas_n,n,,-1	(Output) Gas law exponent. If you specify an exponent this will be boringly constant. If you use the table, the exponent will change with temperature and pressure.
O4=Fluid,y,gal,-1	(Output) Amount of fluid in the accumulator

The blocked component

When put on the end of a line, this component does not allow any fluid to exit the line through that end. The blocked component dialog box is shown below:

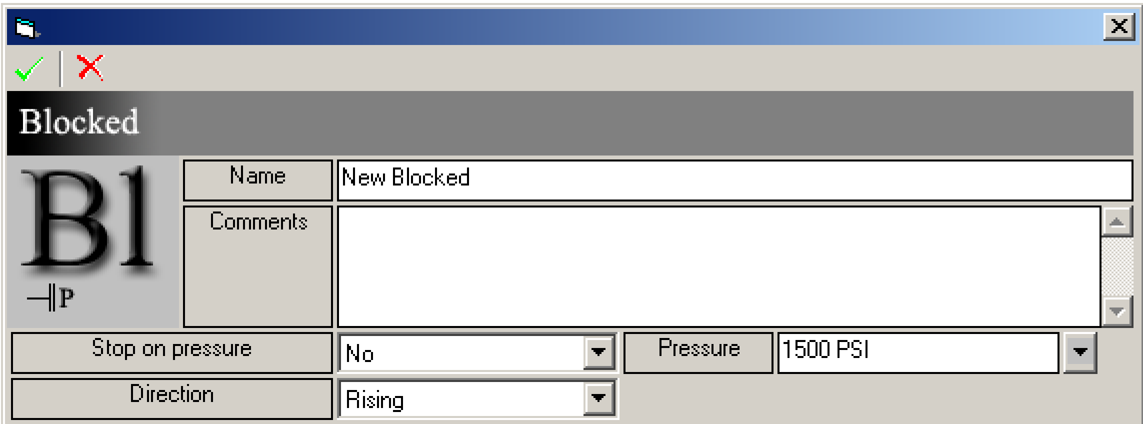


Figure 32

Property	Description
[[Blocked_1]	Unique identifier for this component
Cn = ""	Comment line where *n* is an integer >=1
QtyComments = n	
StopOnPressure=yes	You can have the simulation stop when the pressure at this blocked end passes through a reference pressure. If you answer no, then pressure is ignored and the simulation will end base on time alone.
Direction=rising	If you are stopping on the basis of pressure, this determines whether the simulation is stopped when the pressure is above or below the reference point.
Pressure=1500 psi	This is the reference pressure for stopping the simulation.

The branch component

The branch is used to connect more than one line to the output of a discharge device. The branch component does not have a component properties' editor.

Property	Description
[Branch_1]	Unique identifier for this component
Cn=""	Comment line where *n* is an integer >=1
QtyComments=n	
Qty_Branches=2	This value cannot be modified

The document component

The document component dialog box is show below:

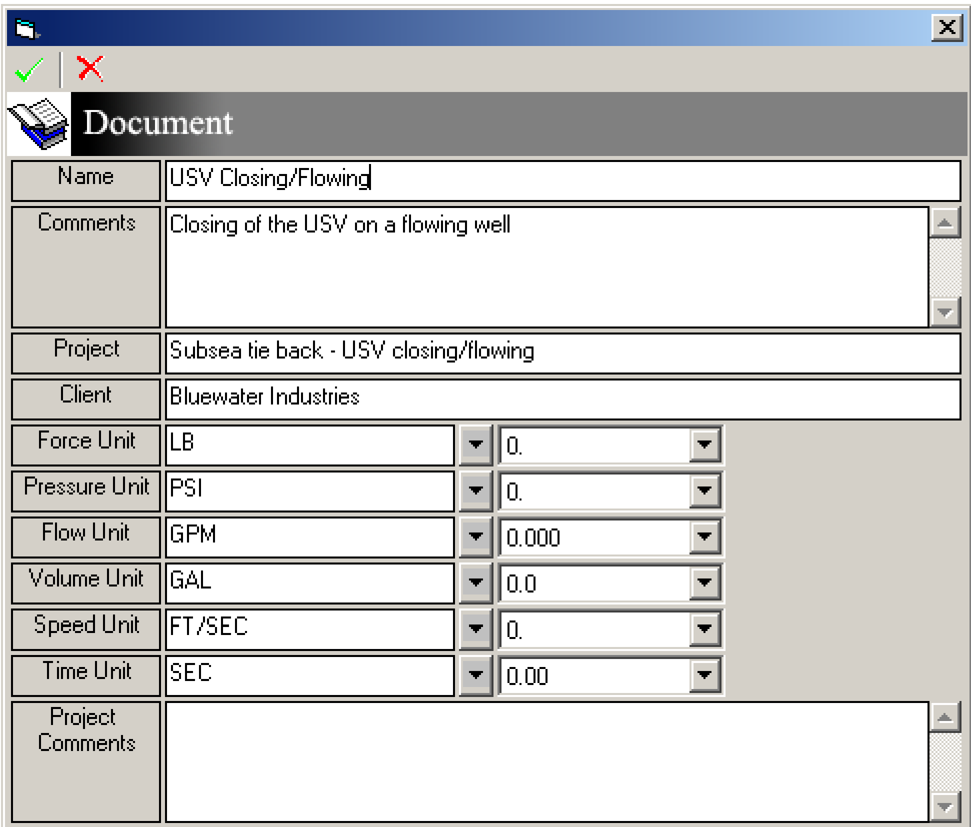


Figure 33

Property	Description
[Document_1]	Unique identifier for this component
Cn=""	Comment line where *n* is an integer >=1
QtyComments=n	
Project="Titanic"	A line identifying the project
Client="Oil Gulf"	A line identifying the client
PressureUnit=psi,-1	(See explanation below)
FlowUnit=gpm,-1	(See explanation below)
VolumeUnit=gal,-1	(See explanation below)
SpeedUnit=ft/sec,-1	(See explanation below)
TimeUnit=sec,-1	(See explanation below)

This component holds general descriptions and explanations related to a simulation. There must be one, and only one of these in each assembly that can be run. The computation engine uses the units for any computed quantity that does not otherwise have a unit assigned to it.

The fluid component

The fluid component dialog box is show below:

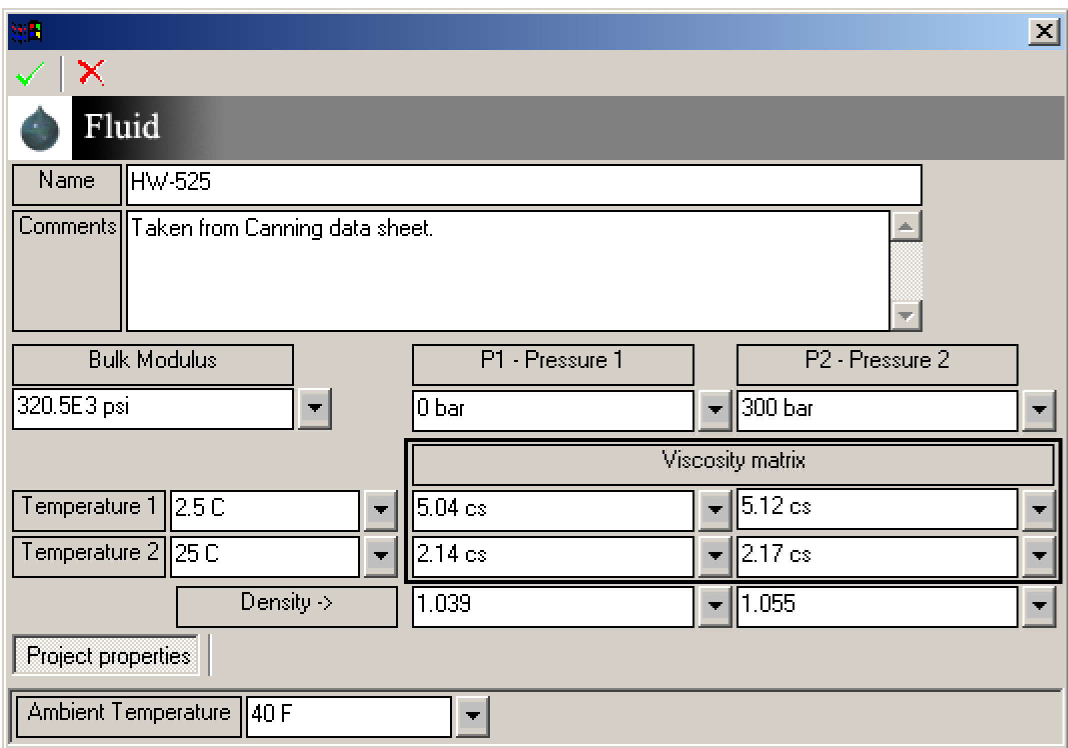


Figure 34

Property	Description
[Fluid_1]	Unique identifier for this component
Cn = ""	Comment line where *n* is an integer >=1
QtyComments = n	
T1=2.5 C	Two temperature points to allow interpolating a viscosity at the operating temperature. Temperature must be >0 absolute. A logarithmic interpolation is used, not linear.
T2=25 C	
P1=0 bar	Two pressure points to allow interpolating a viscosity and density at the operating pressure.
P2=300 bar	
V_at_P1T1=4.38 cs	Viscosity matrix for interpolating a fluid viscosity at the working temperature and pressure.
V_at_P2T1=4.45 cs	
V_at_P1T2=1.93 cs	
V_at_P2T2=1.96 cs	
Density_at_P1=1.011	Two density points to allow interpolating a density at the operating pressure.
Density_at_P2=1.027	
Bulk_Modulus=315.378E3 psi	This is the mathematical inverse of the fluid compressibility.
Ta=40 F	The ambient temperature is used to compute a Viscosity for all fluid calculations.

This component holds fluid properties. There must be one of these in each assembly that can be run.

The hose component

The hose component dialog box is show below. At the end of this section is a discussion on how to choose the various hose parameters.

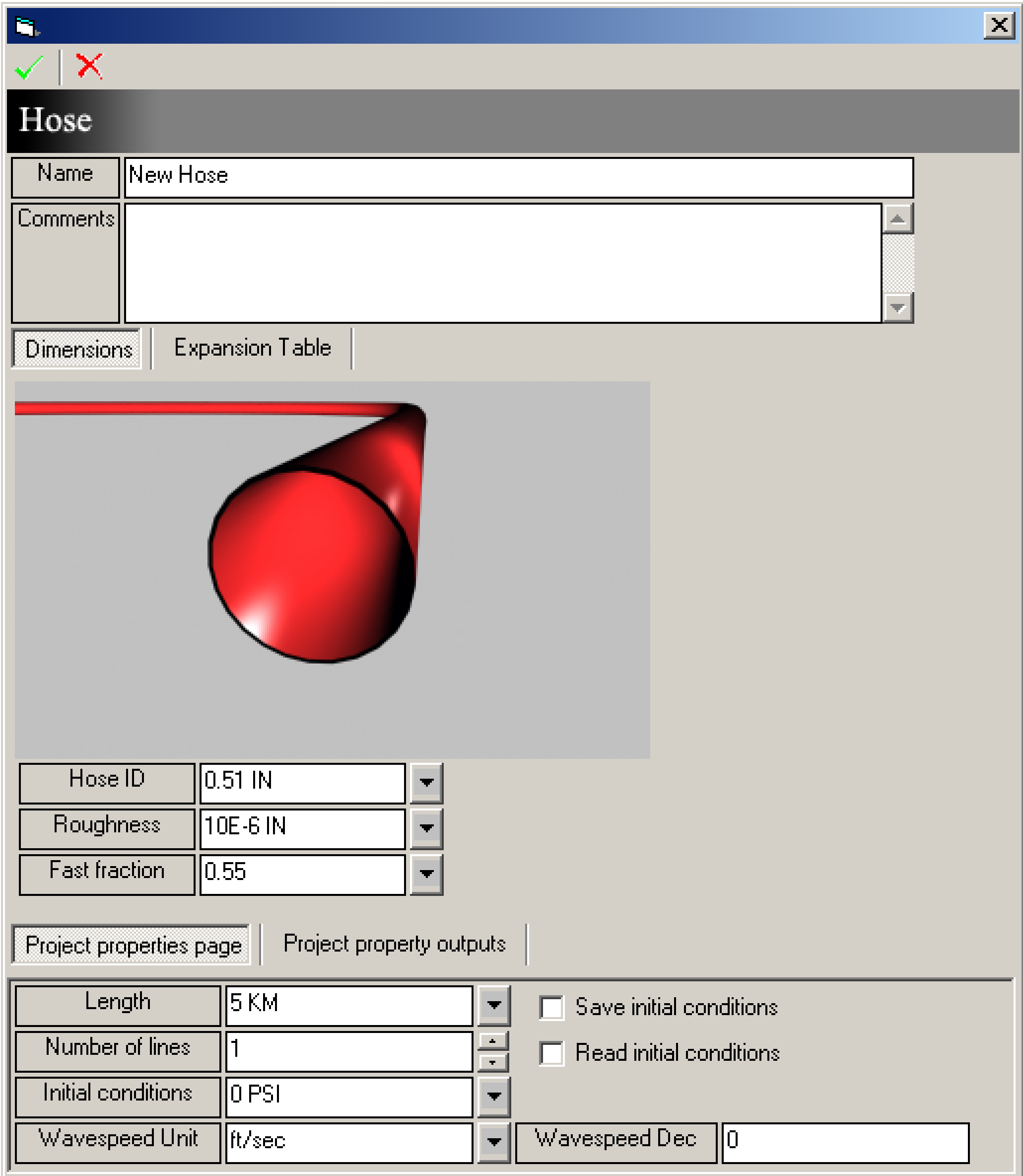


Figure 35

Property	Description
[hose_1]	Unique identifier for this component
Cn = ""	Comment line where *n* is an integer >=1
QtyComments = n	
ID=0.51 IN	Hose internal diameter
Roughness=10E-6 IN	Absolute roughness of the internal surface of the hose
FastFraction=0.55	This is the ratio of the viscous expansion that occurs with the fast time constant to the total viscous expansion.
ExpansionUnit=CC/FT	Can also be % (applies to the expansion table)
PressureUnit=PSI	(Applies to the expansion table)
TimeUnit=SEC	(Applies to the expansion table)
P1=0.13, 25, 0.14700, 60, 80	Data points describing the volumetric expansion as a function of pressure. There can be any number of points. The format is: Px=exp, pres, ratio, fast, slow. *Exp* is the expansion (increase in volume due to internal pressure) *Pres* is the pressure inside the hose relative to outside *Ratio* is the ratio of elastic to total diameter change *Fast* is the fast viscoelastic time constant *Slow* is the slow viscoelastic time constant
P2=0.23, 50, 0.15600, 60, 80	
P3=0.36, 100, 0.18300, 60, 80	
P4=0.53, 200, 0.22100, 60, 80	
P5=0.74, 400, 0.27600, 60, 80	
P6=1, 800, 0.35000, 60, 80	
P7=1.39, 1600, 0.43600, 60, 80	
P8=1.71, 2400, 0.49300, 60, 80	
QtyPoints=8	Number of expansion table points (computed automatically)
Length=5 km	Length of hose
QtyLines=1	Number of identical lines in parallel
Wavespeed_Unit=ft/sec	Unit of measure to use when displaying the calculated wave speed.
Wavespeed_Dec=0	Number or decimals for displaying wave speed
Initial_Cndx=1500 psi	Initial pressure or flow in the hose component
Save_Cndx=y	Save final conditions for use in the next simulation
Read_Cndx=y	Read final conditions from the previous simulation
QtyOutputs=8	
O1=Src_P,y,psi,-1	Pressure at source end of this line
O2=Src_Q,y,gpm,-1	Flow rate at source
O3=Src_Qtot,n,gal,-1	Totalized flow at source
O4=Dis_P,y,psi,-1	Pressure at discharge end of this line
O5=Dis_Q,y,gpm,-1	Flow rate at discharge
O6=Dis_Qtot,n,gal,-1	Totalized flow at discharge
O7=Re,n,,-1	Reynolds's number at the source end
O8=f,n,,-1	Friction factor at the source end

Expansion table points are entered using the second page of the hose component dialog box

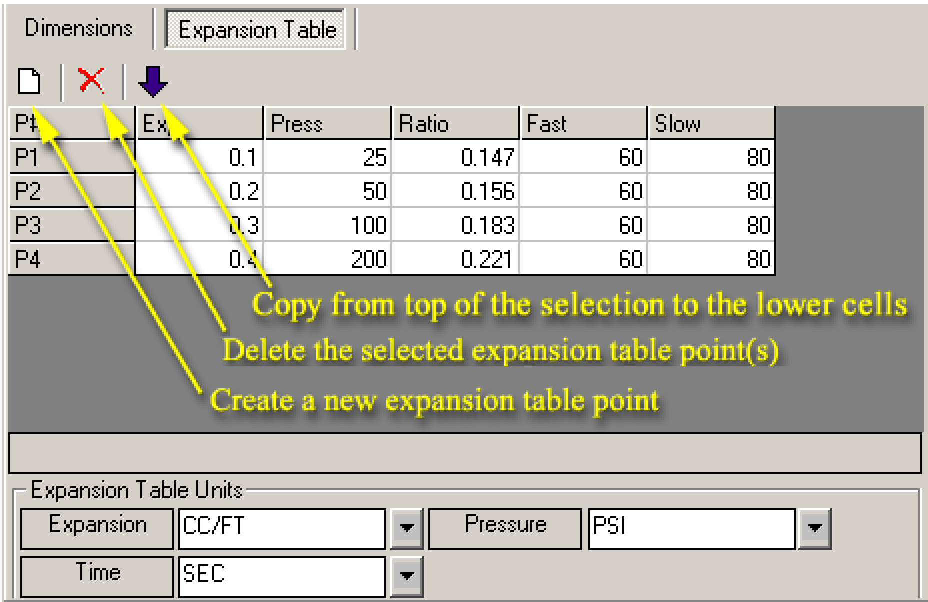


Figure 36

Suggestions for building a hose file

Most likely, the only parameters you will be able to get from the hose manufacturer are the Hose ID, volumetric expansion (as a table or a curve) and (maybe) internal roughness. The remaining parameters will have to be estimated.

At the conference *Umbilicals, the Future*, organized by the Society for Underwater Technology and held in London, 14 December 1995, a paper was delivered which reported on the measurement of viscoelastic properties of hose. “The Dynamic Response of Thermoplastic Hoses” by P.S. McCarthy and P.H. Knight contains a wealth of objective data on hoses and is the basis for some pretty good guesses at the difficult to measure viscous parameters. From this paper, fast fraction is estimated at 0.55, the fast time constant at 30 to 40 seconds, and the slow time constant at 400 seconds.

"Ratio" affects the speed of sound. A ratio of 1 indicates perfectly elastic expansion and yields excessively low values for the speed of sound. Normally, the speed of sound in a hose is low at low pressure and increases smoothly as pressure increases. Here is the process I go through to make a well behaved hose file.

The hose manufacturer normally gives Pres and Exp as the expansion curve for the hose. Make a plot of the typical expansion data as would actually be measured by the manufacturer. The curve should be smooth and get flatter as pressure is increased. Any bumps or oscillations should be taken out. Then, using the default ratio of 0.4 look at the calculated speed of sound that results in the simulation. Speed of sound should increase monotonically with pressure from something like 700 to 1500 ft/sec near 0 pressure to something like 2500 to 2700 ft/sec at operating pressure. I have not personally observed anything over 3000 ft/sec.

NOTE: Any real, accurate test data you have should take precedence over these estimations.

The inlet port component

The inlet port component uses a generic dialog box as show below:

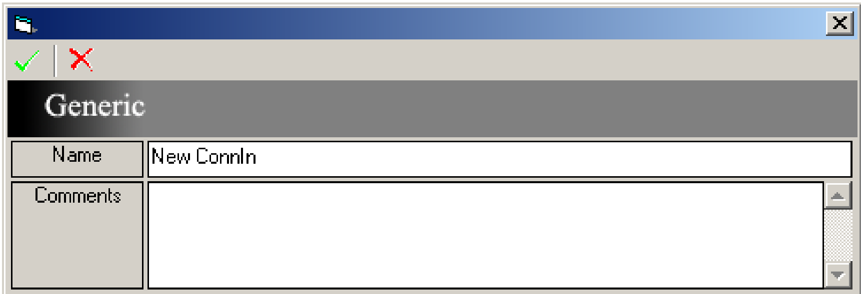


Figure 37

Property	Description
[ConnIn_1]	Unique identifier for this component
Cn = ""	Comment line where *n* is an integer >=1
QtyComments = n	

This component is used for creating subassemblies and utilizing these as building blocks for creating more complex assemblies. A port component will represent an inlet port in the subassembly icon once entered in the parent assembly. Components or assemblies that precede this one can connect to this inlet port.

The outlet port component

The inlet port component uses a generic dialog box as show below:

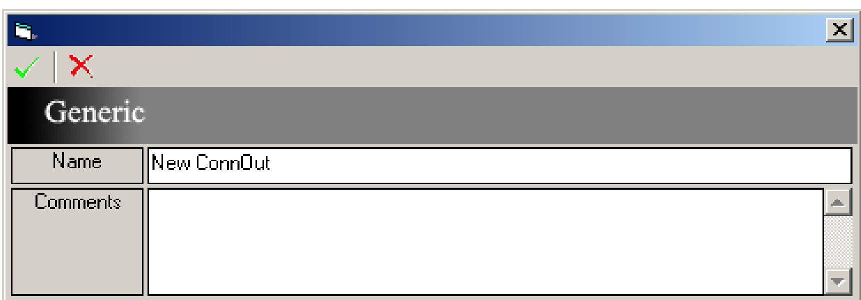


Figure 38

Property	Description
[ConnOut_1]	Unique identifier for this component
Cn = ""	Comment line where *n* is an integer >=1
QtyComments = n	

This component is used for creating subassemblies and utilizing these as building blocks for creating more complex assemblies. A port component will represent an outlet port in the subassembly icon once entered in the parent assembly. This port is used to connect an assembly to components or assemblies that follow this one.

The pressure discharge component

The pressure discharge component supplies or establishes a pressure at the discharge end of a line. The pressure discharge component dialog box is show below:

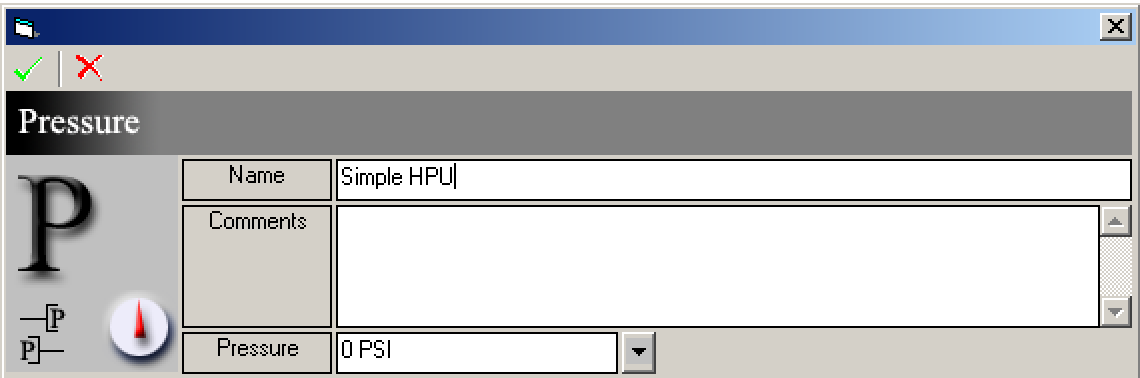


Figure 39

Property	Description
[PressDis_1]	Unique identifier for this component
Cn=""	Comment line where *n* is an integer >=1
QtyComments=n	
Pressure=1500 psi	

The pressure source component

The pressure source component supplies or establishes a pressure at the source end of a line. The pressure source component uses the same dialog box as the pressure source component.

Property	Description
[PressSrc_1]	Unique identifier for this component
Cn=""	Comment line where *n* is an integer >=1
QtyComments=n	
Pressure=1500 psi	

The pump component

The pump component represents two pumps, one electric, and one air driven. Both feed the same output. Both pumps, one pump or neither pump might be running at any particular time. The pump component dialog box is show below:

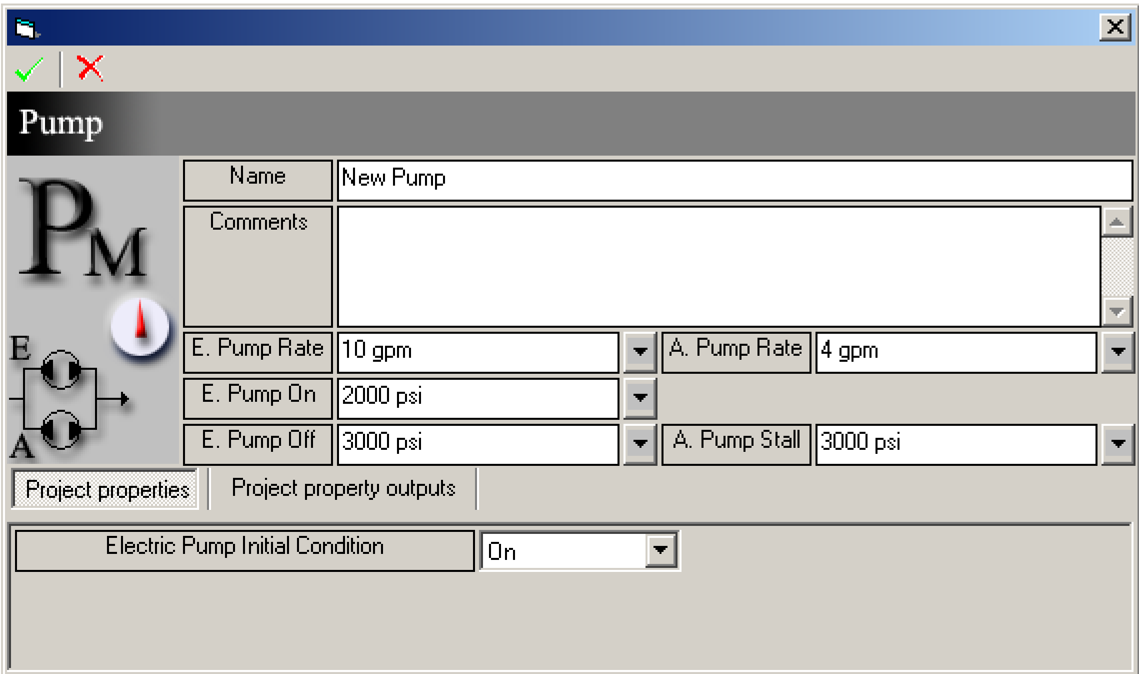


Figure 40

Property	Description
[pump_1]	Unique identifier for this component
Cn = ""	Comment line where *n* is an integer >=1
QtyComments = n	
E_Pump_Rate=10 gpm	Constant flow pump rate put out by the electric pump when on
E_Pump_On=2000 psi	Pump low limit on switch
E_Pump_Off=3000 psi	Pump high limit off switch.
A_Pump_Stall=3000 psi	Pressure at outlet which will stall the air driven pump (pressure at which its flow rate is reduced to zero)
A_Pump_Rate=4 gpm	Flow from the air driven pump with no outlet pressure present. This is the absolute maximum flow the air pump can manage.
E_Pump_Init=on	initial condition of the electric pump
A_Pump_Init=on	initial condition of the air pump
QtyOutputs=3	
O1=Flow,y,gpm,-1	(Output) Total flow rate for both pumps
O2=E_Flow,n,gpm,-1	(Output) Electric pump flow rate
O3=A_Flow,n,gpm,-1	(Output) Air pump flow rate

The regulator component

The regulator component dialog box is show below:

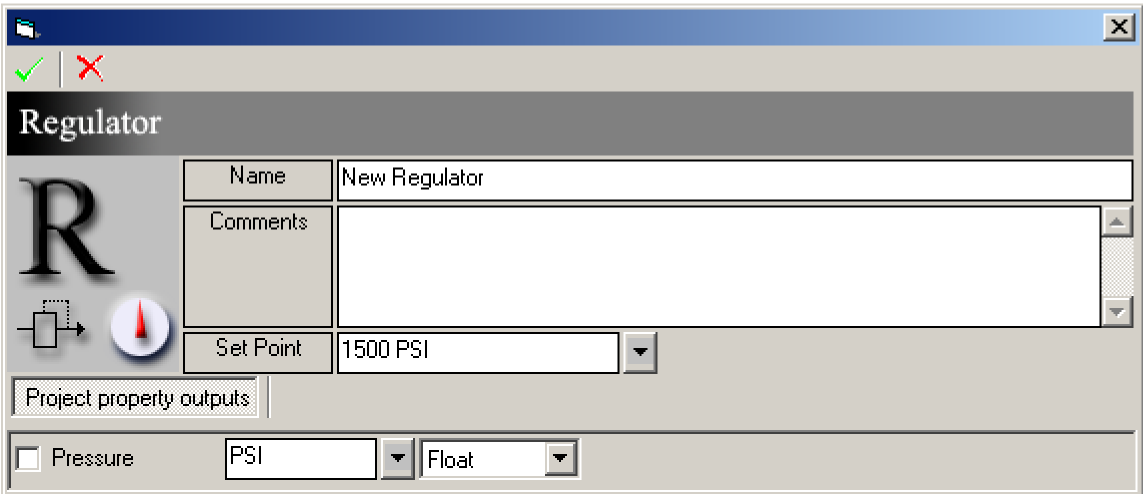


Figure 41

Property	Description
[regulator_1]	Unique identifier for this component
Cn = ""	Comment line where *n* is an integer >=1
QtyComments = n	
Set_Point=1500 psi	If the pressure at the regulator inlet is at or above the set point, then the outlet pressure will be equal to the set point. If the inlet pressure is less than the set point, then the outlet pressure will equal the inlet pressure.
QtyOutputs=1	
O1=Pressure,n,psi,-1	(Output) Pressure

The restriction component

A restriction is defined to be a square-law-type restriction followed by a length of tubing. You may choose the length of the tubing to be zero. If it is not zero, then the simulation computes an equivalent Cv for the tubing and combines it with the value of the Cv property. If you only want the tubing part, the Cv property can be made negligible by entering a Cv 5 to 10 times the Cv of the tubing. The restriction component dialog box is show below:

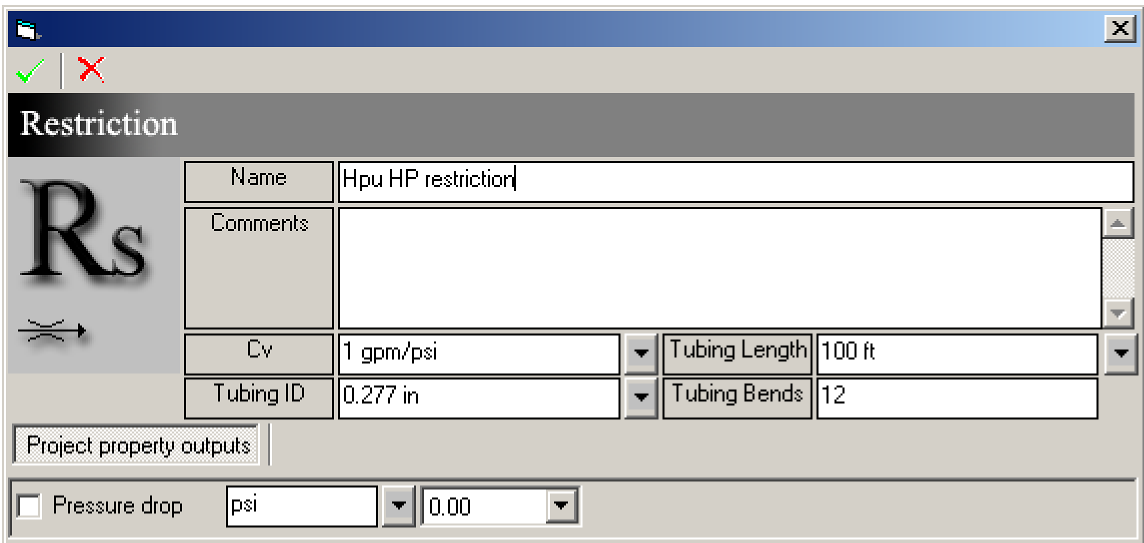


Figure 42

Property	Description
[restriction_1]	Unique identifier for this component
Cn = ""	Comment line where *n* is an integer >=1
QtyComments = n	
Cv=0.5 gpm/psi	The restriction value itself
TubingLength=0 ft	Length may be zero
TubingID=0.5 in	
TubingBends=12	
QtyOutputs=1	
O1=dp,n,psi,2	(Output) Pressure drop

The sea chest component

The sea chest component uses a generic dialog box as show below:

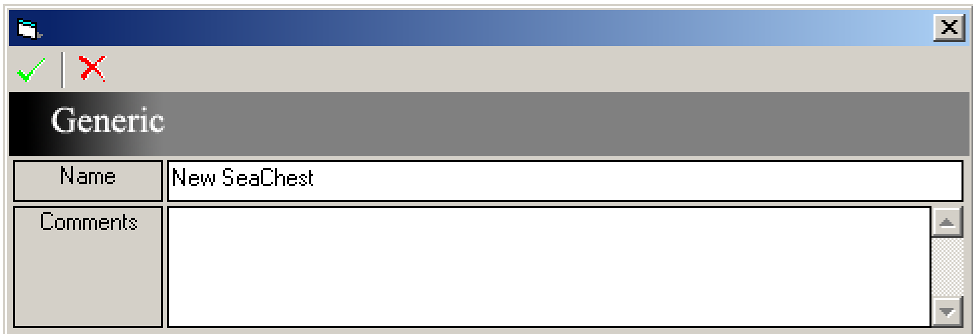


Figure 43

Property	Description
[SeaChest_1]	Unique identifier for this component
Cn = ""	Comment line where *n* is an integer >=1
QtyComments = n	

The timing component

The timing component dialog box is show below:

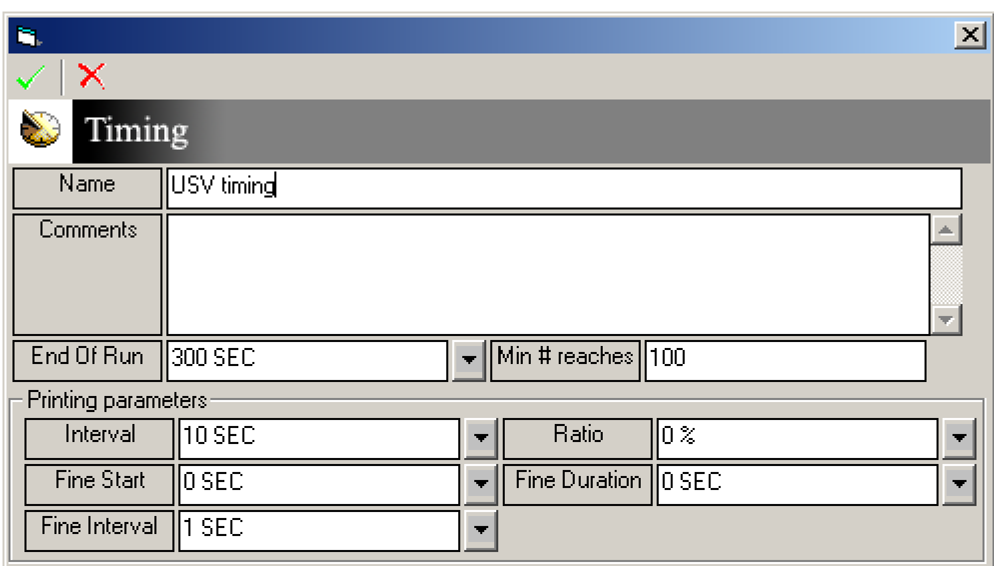


Figure 44

Property	Description
[timing_1]	Unique identifier for this component
Cn = ""	Comment line where *n* is an integer >=1
QtyComments = n	
End_of_run=60 SEC	Analysis end time. Unless the simulation is ended earlier by the condition of a discharge component, the simulation will end when this much simulated time has elapsed.
Reaches=50	Divide line into at least this many sections. Simulations run faster with fewer reaches, but the accuracy is degraded. Using 50 is generally good for systems with one line. If you are in doubt, try a number, then double it. If the results do not change much, then use the smaller number. Also, some kinds of numerical instability can be resolved by using more reaches.
Print_Interval=1 SEC	Controls the simulated time interval between output samples. See text.
Print_Ratio=0 %	This is used to make the output samples follow a quasi-logarithmic sample interval. See text
Fine_Start=0 SEC	It is sometimes useful to have a large Print_Interval to minimize the number of points in the output, but still print with a small interval during some important event. This is the use of fine printing. You specify a start time, end time and fine print interval.
Fine_Duration=0 SEC	
Fine_Interval=0 SEC	Setting Fine_Interval to 0 disables fine printing.

Note: The Print_Interval in combination with the Print_Ratio determines how often a sample is printed.

As long as Print_Ratio=0, then Print_Interval is the desired time interval between printed outputs. The actual interval between printed outputs will usually be close to this, but the interval will not be exact due to the fact that internal calculations are made at a fixed time interval (based on speed of sound and Reaches) and the print interval must be a whole number multiple of the fixed calculation time interval.

If Print_Ratio > 0, then the interval between printed points will increase with simulated time. Whenever Time*Print_Ratio > Print_Interval, the actual print interval will be Time*Print_Ratio. Otherwise, the interval between printed points will be Print_Interval.

Example: assume Print_Ratio=10% and there is a printed output at time=100 sec. Then the next sample will be printed at 110 sec (T + 10%*T).

The tubing component

The tubing component dialog box is show below:

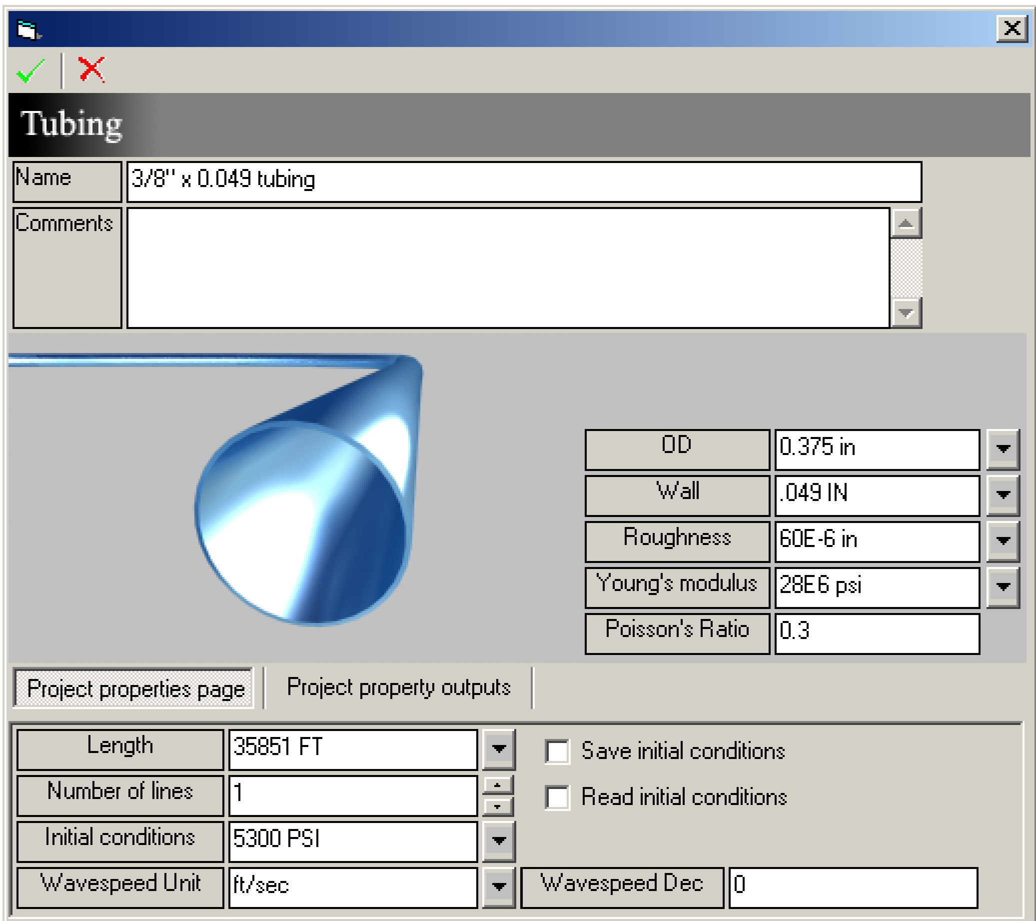


Figure 45

Property	Description
[tubing_1]	Unique identifier for this component
Cn = ""	Comment line where *n* is an integer >=1
QtyComments = n	
OD=0.5 in	Outside diameter of the tubing
Wall=0.032 IN	Tall thickness
Youngs_modulus=28E6 psi	Modulus of the wall material (use 28E6 psi for stainless, 30E6 psi for carbon steel, unless you have better data)
Poisson_ratio=0.3	For any metal tube, use 0.3 unless you have better data
Roughness=5E-6 in	
Length=5 km	
QtyLines=1	Number of identical lines in parallel
Wavespeed_Unit=ft/sec	
Wavespeed_Dec=0	
Initial_Cndx=0 psi	Pressure or flow in the line at the start of the simulation
Save_Cndx=y	Save final conditions for use in the next simulation
Read_Cndx=y	Read the final conditions from the previous simulation
QtyOutputs=8	
O1=Src_P,y,psi,0	(Output) Pressure at source end of the tubing
O2=Src_Q,y,gpm,3	(Output) Flow rate at source
O3=Src_Qtot,n,gal,3	(Output) Totalized flow at source
O4=Dis_P,y,psi,0	(Output) Pressure at discharge end of the tubing
O5=Dis_Q,y,gpm,3	(Output) Flow rate at discharge
O6=Dis_Qtot,n,gal,3	(Output) Totalized flow at discharge
O7=Re,n,,0	(Output) Reynolds's number at the source end
O8=f,n,,5	(Output) Friction factor at the source end

The valve operator component

The valve operator component dialog box is show below:

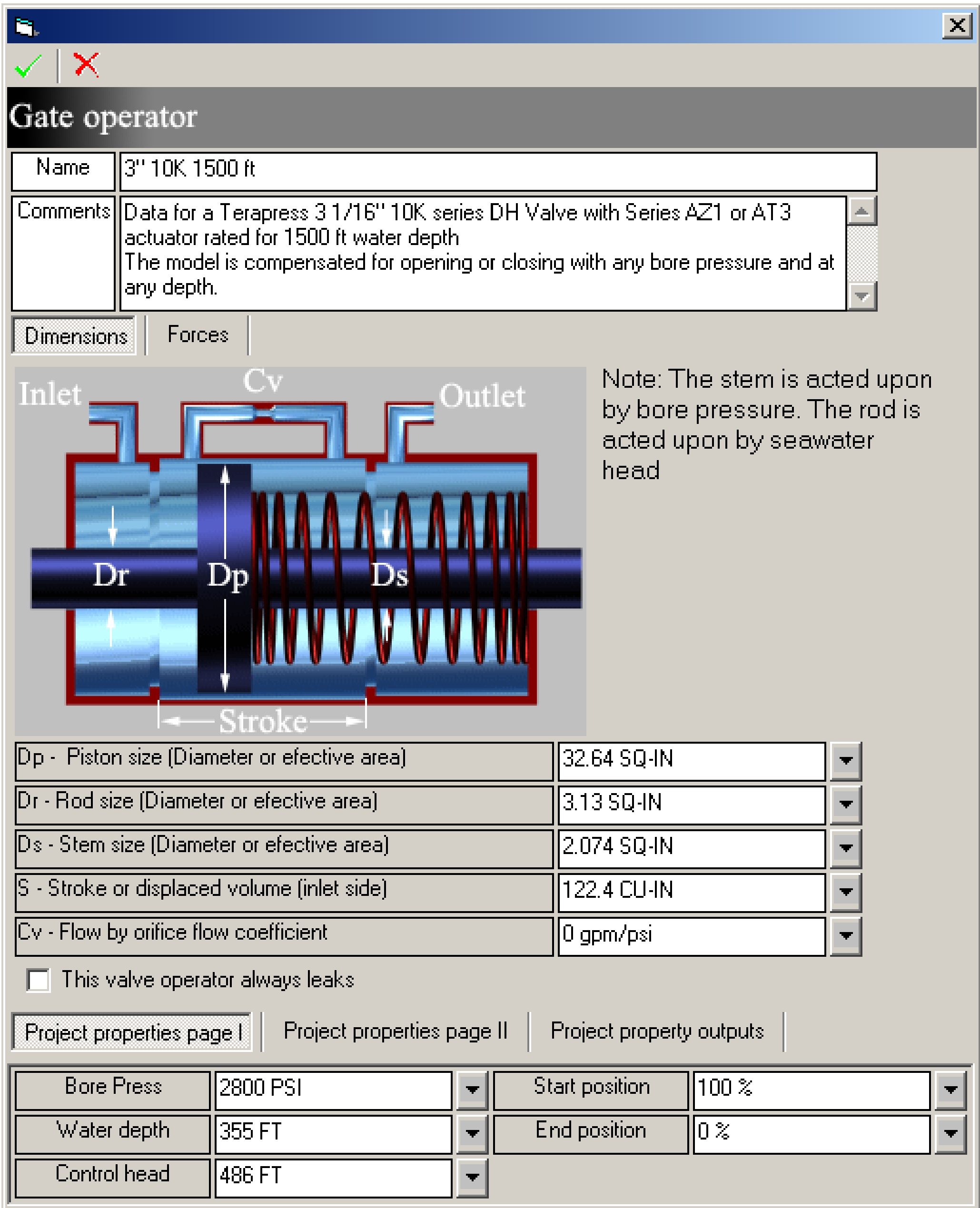


Figure 46

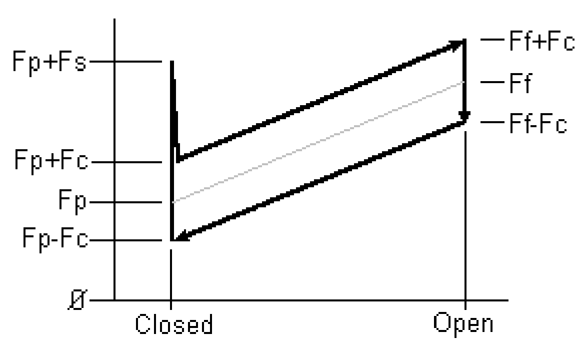
Property	Description
[ValvOpr_1]	Unique identifier for this component
Cn = ""	Comment line where *n* is an integer >=1
QtyComments = n	
Piston_Size=3	This is the outer diameter of the piston (Dp) or the wetted area [0.25*Pi*(Dp^2 – Dr^2)]. The selection is made on the basis of the type of unit (length or area) selected. This area is exposed to the control fluid as specified in Control_Head.
Rod_Size=2 in	Rod diameter (Dr) or area (0.25*Pi*Dr^2) depending on the unit selected. This area is exposed to seawater pressure as specified in Water_Depth.
Stem_Size=1.75 in	Stem diameter (Ds) or area (0.25*Pi*Ds^2) depending on the unit selected. This area is exposed to well bore pressure
Stroke=5	Stroke (S) or volume [S*0.25*Pi*(Dp^2 – Dr^2)] required to move the operator from full close to full open.
Preload=250 psi	Spring preload (Fp - see diagram below) spring force when the valve is in the full closed position. If specified as a pressure, it is assumed to be based on the pressure difference between the front and backside of the piston.
Full_Load=500 psi	Spring full load (Ff - see diagram below) spring force when the valve is in the full open position. If specified as a pressure, it is assumed to be based on the pressure difference between the front and back sides of the piston
Low_Running_Friction=0 lbf	Constant friction opposing valve movement while moving (Fc - see diagram below), evaluated with **no** bore pressure. See High_Running_Friction for more comments.
High_Running_Friction=200 lbf	Constant friction opposing valve movement while moving (Fc - see diagram below), evaluated with **full** bore pressure. If specified as a pressure, it is assumed to be based on the pressure difference between the front and back sides of the piston. Fc is computer according to: Fc = Bore_Pressure*(High_Running_Friction - Low_Running_Friction)/High_Bore_Pressure + Low_Running_Friction.
High_Sticking_Friction=1500 psi	Sticking friction (Fs - see diagram below) Force needed to crack the valve open with full bore pressure across the gate. If specified as a pressure, it is assumed to be based on the pressure difference between the front and back sides of the piston
High_Bore_Pressure=4000 psi	Bore pressure upon which the high running and sticking frictions are based. This pressure is for calculating actual friction at the given bore pressure and may not actually exist anywhere in the system being simulated.
CvFlowBy=0 gpm/psi	Flow restriction of a leak across the piston. This is usually 0.
AlwaysLeaks=y	If yes, leak occurs around the piston even when the operator is in the full open or closed state. Otherwise, it only leaks in the mid 90% of travel.
Bore_Pressure=3000 psi	The actual bore pressure applied to the stem and seals. ***NOTE: this is an absolute pressure (relative to vacuum) not a gauge pressure***. In other words, 3000 psi bore in 1000 ft of water means something different than 3000 psi bore in 5000 ft of water.
Water_Depth=2000 ft	This determines the absolute head pressure applied to the rod and the backside of the piston in the absence of a return line.

Control_Head=2050 ft	This determines the absolute head pressure added to the control side of the piston in addition to the gauge pressure in the line. Control Head may be different from Water Depth due to the HPU being located high above the water surface.
Start_Pos=0 %	% of full open at the beginning of the simulation.
End_Pos=100 %	% of full open at which to stop the simulation. Note that the actual position of the valve model will not go outside of the range, 0% to 100%. Therefore, if you do not want the simulation to stop on the basis of valve position, use –1% when closing or 101% when opening. The simulation will then stop on the basis of time only.
Group_Spacing=20 min	Groups give you a way of simulating the sequential shut-in of several trees with one simulation. In order to simulate groups of identical valves, the End_Pos must be set to end only on the basis of time
Opr_Per_Group=5	Valve operations per group (must be >= 1)
Opr_Interval=2 min	Interval between operations in a group. At the beginning of each operation, the valve position is set to Start_Pos.
Outlet_Head=Sea water	Determines whether the head on the outlet is determined by sea water and water head, or control fluid and control head.
QtyOutputs=6	
O1=Src_P,y,psi,0	(Output) pressure at operator inlet
O2=Src_Q,n,gpm,3	(Output) flow rate into operator
O3=Pos,y,%,2	(Output) position of operator, % of open
O4=Flow_By,n,gpm,3	(Output) leakage rate past the piston
O5=Dis_Q,n,gpm,3	(Output) flow rate at operator outlet
O6=Dis_P,n,gal,0	(Output) pressure at operator outlet

Notes:

- Loads and frictions are expressed in terms either of force or area.
  - Preload > 0
  - Full_Load > Preload
  - Frictions must be >= 0. If given in psi, then the number refers to the pressure difference between the inlet and outlet ports.
- The high friction terms are for "High_Bore_Pressure" in the bore.
- The low running friction is for no pressure in the bore.
- Friction is assumed to be proportional to bore pressure.

Forces Required Diagram



Forces are those that would be measured in a bench test with a pressure gauge on the inlet

Figure 47

Troubleshooting

Selected Link does not conform…

Consult the section on Allowable Assembly Schematics if you try to run a simulation and get an error message saying something like "The selected link does not conform with any of the currently supported mathematical models…".

Instability

An unstable solution is one that results in huge, crazy fluctuations in pressure and flow that usually grow as the simulation continues. These fluctuations can result in the program terminating early or even crashing. The most common causes of instability are input data errors, not dividing the lines into enough parts (reaches), an accumulator that is way too small, or combining very long lines with very short ones.

NOTE: It is normal for the pressure to show alternations of a percent or so from one step to the next, superimposed on top of the normal pressure rise or decay curve. You will only see these alternations if you set the print interval so short that each calculated time point is printed. They are artifacts of the calculation process, not instability.

If you encounter instability, look for any input data where you slipped a decimal place or where some value is very wrong. If that does not fix it, increase the number of reaches ("Min # reaches" in the timing component) so that everything is done in smaller steps. Make sure that the time steps are small enough that it takes many steps to fill an accumulator or move a valve.

A common cause of instability problems is from trying to mix really long lines with really short ones. See the next section for more information.

Combining Long and Short Lines

Instabilities can result when using multiple lines if one line is much shorter than the longest line. This happens when the software cannot properly set the lengths of the segments of the various lines. When using multiple lines, look at the timing section of the header or summary output from the program. *Always select the number of reaches (in the timing component) so that timing section does not include messages saying something like "Number of segments revised to...". It may, however, include revisions of the wave speed, as long as they are not large changes.*

The Fix

The fix usually involves increasing the number of requested segments ("Min # reaches" in the timing component). However, if you cannot get rid of the warning by revising "Min # reaches", then one line is probably too short to simulate accurately (like a 50 km supply line combined with a 50 meter jumper). In this case, you will have to use an equivalent restriction in place of the short line.

Why This Happens

The mathematical model used in The Control Simulator requires a well-defined relationship to exist between the number of segments (reaches) a given line is divided into, the size of the time steps, and the speed of sound on the line. More reaches or shorter lines result in shorter time steps.

The number of reaches must be greater than 15 because simulation accuracy suffers with fewer reaches. But requiring 15 reaches or more in the shortest line sets the required time step to a small value. When you apply that small time step to a long line, it results in a large number of reaches for the long line. However, the maximum number of reaches is limited to 4093 to limit memory usage and run time. If the compute engine cannot satisfy the relationship between reaches and time step on all of the lines, then the timing relationship for one or more lines will be wrong and an accurate simulation is impossible.

Glossary of terms

Database

The Control Simulator stores all information in databases. One useful way to think of and use a database is that it contains in one file all of the simulation cases related to a single project. Each simulation case (charging the supply system or closing a valve, for instance) is contained in an assembly, with one (or more) assembly per case.

The structure of a database allows the Control Simulator to store as many components and assemblies as the user decides and is only limited by the size of the drive containing the database file. Because of this flexibility, the user may choose to use one database to store one assembly, an entire project, or all projects. When requested by the user, the Control Simulator creates a new database by creating a copy of the prototype file Template.mdb. For this reason, any component definitions and assemblies included in the Template.mdb file will also be present in the newly created database.

Control Simulator database files are named *Filename.mdb*

Component

A component in the Control Simulator represents a physical hydraulic component or group of components that can be analyzed as a unit. A collection of properties called intrinsic properties defines a component in the Control Simulator. These properties do not change regardless of where the logical component (the representation of the physical component or components) is used. If you change the value of one of these intrinsic properties (such as the volume of an accumulator), that change is applied to every instance of the component any assembly where it is used.

In order to use components having different intrinsic properties you must define different components. For example, to use a 20 gallon accumulator in one assembly and a 5 gallon accumulator in another, you must define two accumulator components.

Instance

An instance is defined as a reference to a component definition from within an assembly. It consists of both the component's intrinsic properties as well as of any applicable project properties.

Assembly

In the Control Simulator, an assembly is defined as a collection of components that are interconnected via logical links (see Logical link) and represent a hydraulic circuit (or a portion of one).

Intrinsic property

Intrinsic properties are part of the basic definition of a component. These are properties that do not change regardless of where the component is used. Example: the internal empty volume of an accumulator, the stroke of a valve, or the outside diameter of a stainless tube. (see Project Property)

Logical connector

A logical link is the means by which component sequence and connection configuration is established in the Control Simulator GUI. These are depicted as lines running from an outlet port of a component to an inlet port of the next downstream component. The upstream side of the logical link is always shown in a different color (red when unselected, blue when selected).

Project property

Project properties are associated with an instance of a component. These properties may change depending on where a component is used. Example: the precharge pressure in an accumulator, the starting position of a valve, or the length of a control line. (see Intrinsic Property)

Although associated with an individual instance of a component, project properties are actually part of the assembly and are saved as part of the assembly. For instance, suppose you have an assembly using ½" tubing and you want to see the effect of changing to 3/8" tubing. By dragging the 3/8" tubing from the database and dropping it on top of the ½", all of the intrinsic properties (such as diameter and wall thickness) are replaced by those of the 3/8" tubing. But project properties (like the length of the tubing) are not changed.

Top-level assembly

Assemblies can be used as building blocks for defining more complex assemblies. The top-level assembly is the assembly that is active when the simulation is being run. Any sub-assemblies used as building blocks for the assembly being analyzed are said to be *contained* by the top-level assembly.

Source

A source component is a component that can be placed at the control location (the platform, for instance). Source is a functional title, not an indication of the direction of fluid flow. The source may be a supply or vent for fluid depending on the pressures in the line and the source. The Control Simulator currently supports two types of sources: A pump component and a pressure source component.

Also, the end of a line, component, or assembly that is nearest to the control location is called the source end. As with the source component, "source" does not imply direction of fluid flow. Fluid can flow into or out of the source end. Flow is always positive when the fluid is moving from a source toward a discharge. (see Discharge)

Line

A line is the means for conveying hydraulic fluid from one component to another. The two types of lines currently supported by the Control Simulator are the hard tubing and the thermoplastic hose.

The hard tubing model applies to any conduit that exhibits elastic, linear stretching of the walls with changes in pressure. Stainless steel, titanium, and carbon steel, are obvious examples of hard tubing.

The thermoplastic model applies to any conduit that exhibits non-linear, viscoelastic, time dependent stretching of the walls with changes in pressure. Hoses reinforced with Kevlar or polyester, and PVC pipe[2] are examples of conduits that require the thermoplastic hose model.

Discharge

A discharge component is a component that can be placed at the "far" end of a line, away from the control location (the tree, for instance). Discharge is a functional title, not an indication of the direction of fluid flow. The discharge may be a supply or vent for fluid depending on the pressures in the line and the discharge.

Also, the end of a line, component, or assembly that is farthest from the control location is called the discharge end. As with the discharge component, "discharge" does not imply direction of fluid flow. Fluid can flow into or out of the discharge end. Flow is always positive when the fluid is moving from a source toward a discharge. (see Source)

[2] I am not sure why you would ever use The Control Simulator with PVC pipe, but you could… if you could ever find the needed parameters.