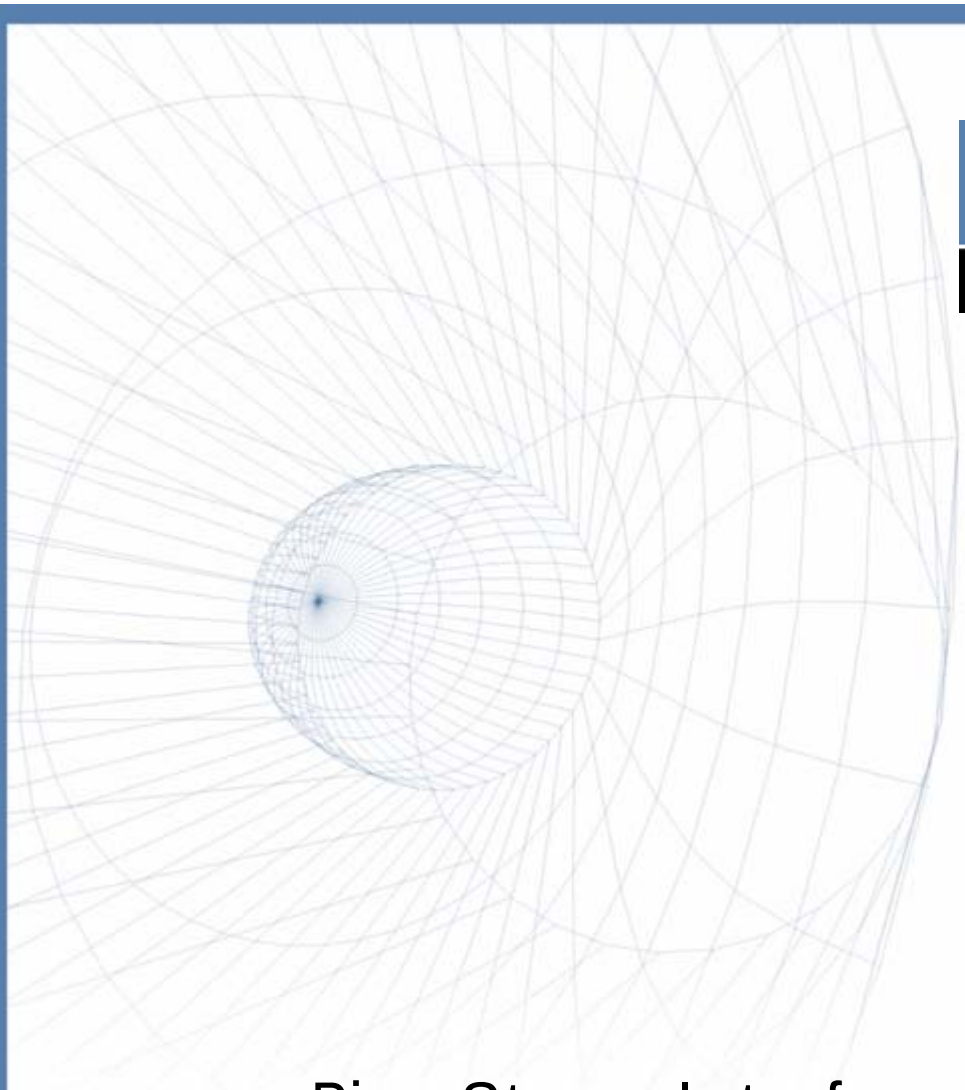


Pipe Stress Interface Administrator Guide

www.aveva.com

Disclaimer

Copyright

First published September 2007

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

Trademarks

Administrator Guide

Contents **Page**

Administrator Guide

1 Introduction

1.1 About this Guide

This manual is designed to give the administrator guidance on how to setup and configure the Pipe Stress Interface.

1.2 An Overview of the PSI Application

The PSI application allows the user to build stress groups configurations consisting of one or more Outfitting branches and transfer the geometry and relevant attributes to and from the Caesar pipe stress package. A Outfitting pipe is used to collect a list of branches to be stressed and after stressing, the same pipe holds a series of branches representing the load case deflections.

Pipe stress packages require more information than is traditionally stored in Outfitting, and a powerful configuration capability is available to enable data to be extracted from different parts of the Outfitting databases. For example, pipe wall thicknesses may be related to the specification component or to an element in the properties database. The configuration tool allows the user to set up how to get this data depending on his project.

1.3 Restrictions

PSI aims to transfer most configurations of pipe branch and components into Caesar and read the stressing results back into an acceptable form in Outfitting. Whilst every effort has been made to ensure this happens as often as possible AVEVA cannot guarantee that every configuration will transfer seamlessly between both systems. For example, the return process relies on comparing the model in PDMS with the results from Caesar but it is possible to do too many changes for the comparison to work correctly. Currently PCOM, SHU, INST elements are treated as two ended components when they are transferred to Caesar and any multiple connections will need to be added manually.

1.3.1 Zero Length Components

Zero length components in Design will be transferred to Caesar with the appropriate nodes. In some cases, this may cause a stress run to generate an error with a zero length component. In these cases it may be necessary for the stress engineer to adjust the data accordingly, by deleting the node or giving it some length value.

1.3.2 Components Without Weights

If PSI is unable to find a component weight, it will output the weight as zero for Caesar input. This will generate an error on the relevant nodes in the node edit form and the error check form, but it will not prevent the pipe being sent to Caesar.

1.4 The Administrator's Role

The administrator is responsible for the initial setup of the defaults associated with the PSI application. This document describes the defaults that can be modified by the administrator in order to configure it to specific project requirements. Each setting is discussed in detail, with extracts and examples to help explain what aspects of the PSI application can be modified and how this can be achieved.

1.5 PSI Sample Project

The PSI application is supplied with two Outfitting projects SAM and MAS. These differ from the standard Outfitting release versions of SAM and MAS as they are configured to use PSI. SAM and MAS contain a sample design model(SAM) and the AVEVA master catalogue project(MAS). For PSI, additional UDAs have been added to both projects and there are some catalogue and Specification components added to the MAS project specifically for PSI.

To configure a user project, the UDAs and Catalogue and Specification components must be in place to enable PSI to work correctly.

1.5.1 Catalogue UDAs

Because the SAM and MAS projects are separate projects and the catalogue is a foreign database in the SAM, it is necessary to have two separate groups of UDAs to apply to the relevant data.

The following UDAs are present in the MAS project (Catalogue and Property UDAs)

UDA Name	Type	Element Types	Description
:MATNUM	REAL	SOLI SMTE	Used to store the piping material number for transfer to Caesar. The configuration defaults will define which one is used.
:TWALL	REAL	TUBD	Used to store the tube wall thickness in the properties database
:CWALL	REAL ARRAY[4]	CMPD	Used to store an
:PSIWEIGHT	REAL	SOLI SMTE	Used to add an override on individual components if the catalogue weight is not available

The following UDAs are in the SAM project (Design UDAS)

UDA Name	Type	Element Types	Description
:PSISYSTEM	REF ARRAY	PIPE BRAN	Used to store the contents of a stress group.
:PSIDATE	STRING	PIPE	Used to store the data the pipe was sent for stressing
:NODETYPE	REAL	ATTA	Used to store restraint type number to match the Caesar support types
:NODEGAP	REAL	ATTA	Used to store the restraint gap
:NODESTIFF	REAL	ATTA	Used to store the restraint stiffness
:NODEFRICTION		ATTA	Used to store the restraint friction

The PSI installation provides macros for building the UDAs depending on the required installation:

For an installation with foreign catalogues like the one described above, there are two macros - one for the catalogue and properties db and one for the design dbs. These are:

`%PDMSPSI%\pdmsui\data\PSICATAUDAS.MAC`

`%PDMSPSI%\pdmsui\data\PSIDESIGNUDAS.MAC`

In a project where the catalogue and design dbs are in the same project, the UDAS can be combined into one Lexicon database. In this case, there is a single macro to build all of the UDAs. This is:

`%PDMSPSI%\pdmsui\data\PSIUDAS.MAC`

1.6 PSI Catalogue and Specifications

The PSI application is designed to work with user catalogues and specifications but there is a an additional specification in the mas project called /PSI-SPWL. This contains a variable angled bend to correct tube misalignment in branches returned from Caesar.

If the user requires this to be in their local project then there is a macro which can be run in PARAGON to create all of the required components. Note the specom names need to be the same as they are used by the application. The macro is:

`%PDMSPSI%\pdmsui\data\PSIBENDCATA.MAC`

2 Project Setup

After installing the UDAs and Variable Bends, the next phase is to install the PSI defaults file in the default directory. PSI looks for a file called “Psi-defaults.xml” in the project defaults directory. For example the SAM projects has a directory samdflts which is referenced by the project environment variable %SAMDFLTS%

The defaults file stores the variables needed to rebuild the defaults form in the state it was when the defaults were last saved. Because of this, it is recommended that the installed defaults file from %SAMDFLTS% is used when first accessing PSI.

3 Pipe Stress Interface Defaults

The PSI defaults form allows the user to configure PSI to suit the user's project data. This can be configured in many ways so it is important to provide facilities to use this data rather than make project changes to suit PSI.

The defaults file is principally used to tell PSI where to extract information and what units to expect, but it also defines how PSI works in the user environment.

The **Pipe Stress Interface Defaults** form is accessed by selecting the **Modify Stress Defaults** icon  on the PSI toolbar.

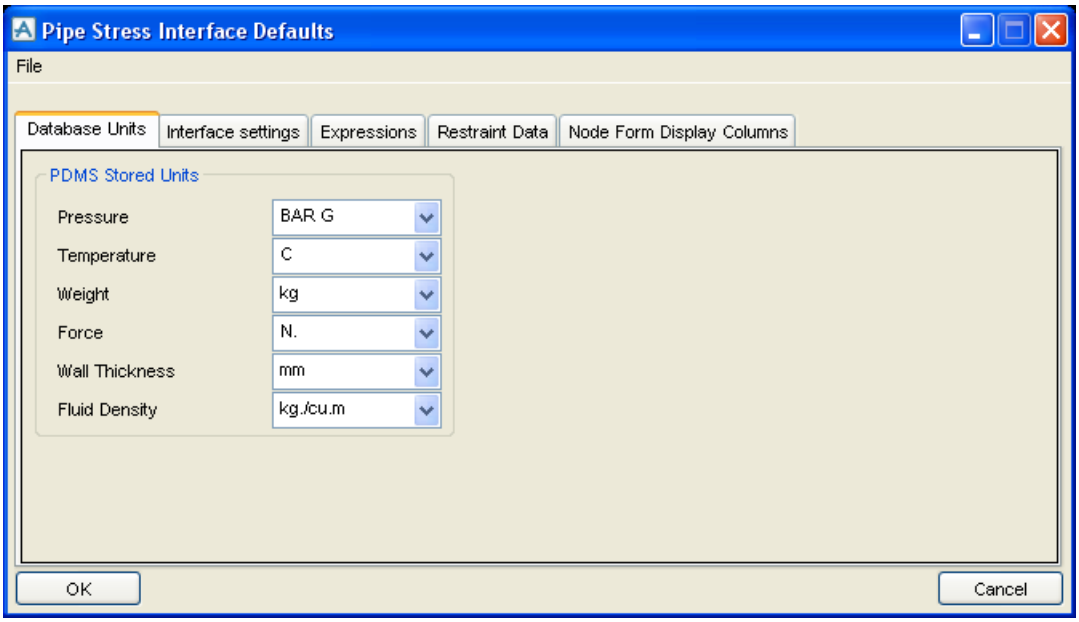


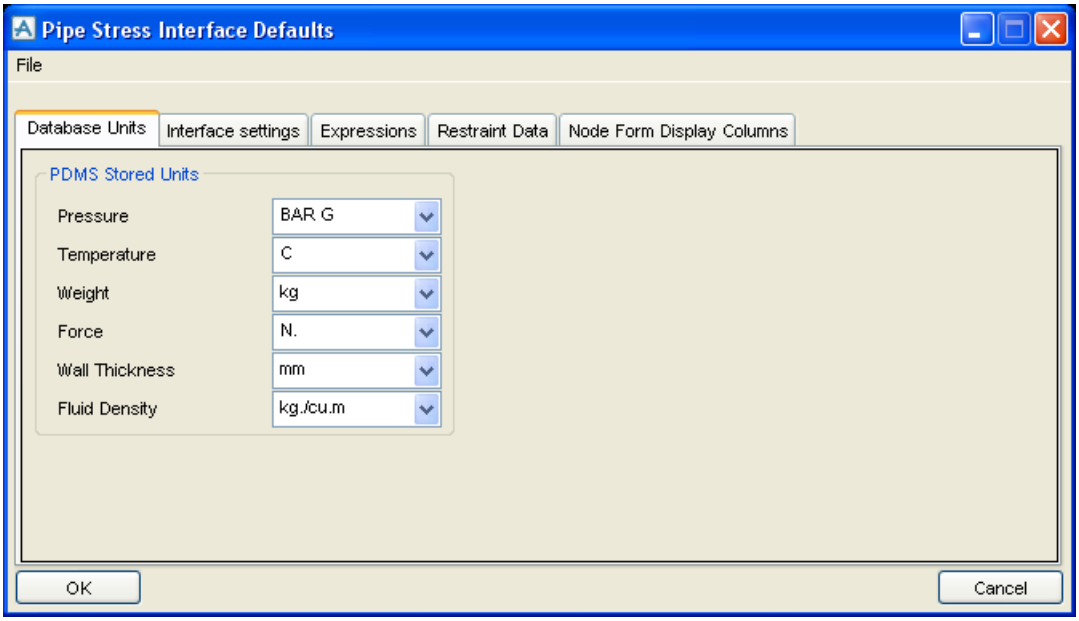
Figure 3:1. The Pipe Stress Interface Defaults Form

The **Pipe Stress Interface Defaults** form has five tabs:

- Database Units
- Interface Settings
- Expressions
- Restraint Data
- Node Form Display Columns

3.1 Database Units

A number of Outfitting units are simple real numbers with no concept of units. For example the temperature on a pipe or branch is a real number there is nothing to say if this is in Fahrenheit or Centigrade. Pressure can be in any number of units but the Outfitting attribute is just a number. For this reason, the PSI system needs to tell Caesar what units to expect. This is stored in the **Pipe Stress Interface Defaults** form.



For each of the properties above, there is a list of appropriate values matching those in Caesar

The Outfitting Stored units are set to default values e.g. Fluid Density is by default measured in kg/cu.m, but needs to be checked and changed as necessary before the pipe is sent to be stressed.

3.2 Interface Settings

The **Interface settings** tab defines various default settings to define how PSI behaves

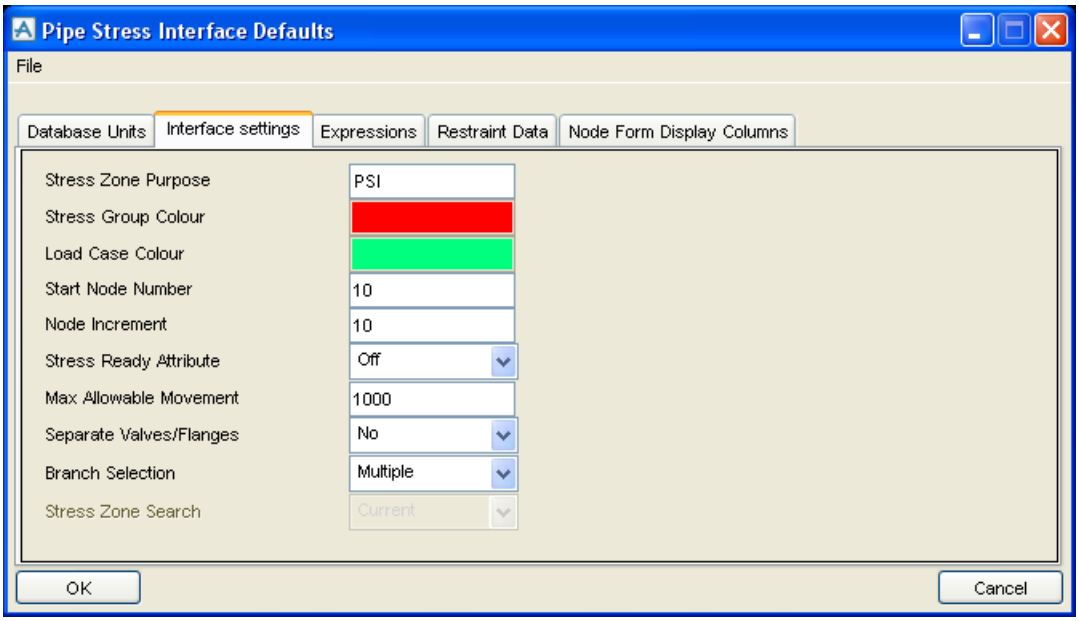


Figure 3:2. Stress Defaults Interface Settings

Stress Zone Purpose

PSI looks at the zone purpose attribute to find zones containing stress groups. In the example shown, the Stress Zone purpose should be set to PSI to be identified as a stress zone.

Stress Group Colour

This refers to the graphics colour of the current stress group.

Load Case Colour

This refers to the display colour of the returning load case branches.

Note: To change the colour of a group click on the coloured box next the appropriate group and you will be given a selection of colours to choose from.

Start Node Number

By default, for any stress pipe, this is the number to start numbering the nodes from.

Node Increment

This is where the increment number between the nodes can be set, by default it is 10 but can be changed to 100 or any other uniform numbering convention that you may have.

Note: For changes to take affect the stress group has to be refreshed, this is achieved by right clicking on the desired stress group and selecting “Refresh Current Group” and toggling the “Nodes” check box on and off. The new node numbering system should now be displayed.

Stress Ready Attribute

This turns the stress ready checking on and off. I.e. when it is turned on each branch that is added to a Stress Group will be checked that it is ready to be stressed. This is used in conjunction with an expression which allows the stress ready attribute to be configured. If the stress ready system is used, it prevents branches from being added to a stress group. It does not prevent a stress run.

Max Allowable Movement

This is the maximum distance that a stressed pipe can be moved when returning into Outfitting. If any load case exceeds this value then the load case is deemed as invalid and the load case will be deleted. If this event occurs then a warning message will be displayed.

Separate Values/Flanges

This option allows connected flanges to be treated as separate rigids. Changing this setting will require the stress group to be refreshed to get a new set of nodes.

Branch Selection

This option allows branches to exist in just one or multiple stress groups.

Stress Zone Search

This option is only active if **Single** is selected in **Branch Selection** which means that a branch can only be in one stress group at any one time. The **Stress Zone Search** allows you to set where to search to see where a branch exists. There are two options: **All** and **Current**. If the search is set to **All** then all stress zones will be checked, or if it is set to **Current** then only the current creation zone is checked to see if the branch exists in another stress group.

3.3 Expressions

The expressions tab allows you to configure where to retrieve information to pass to CAESAR from. Weight, wall thickness, material, temperature, pressure, fluid density, SIF's and insulation density can all be configured from here using PML functions or expressions. The expressions list allows the following to be configured separately.

Weight

Wall Thickness

Material

Corrosion Allowance

Fluid Density

SIFs

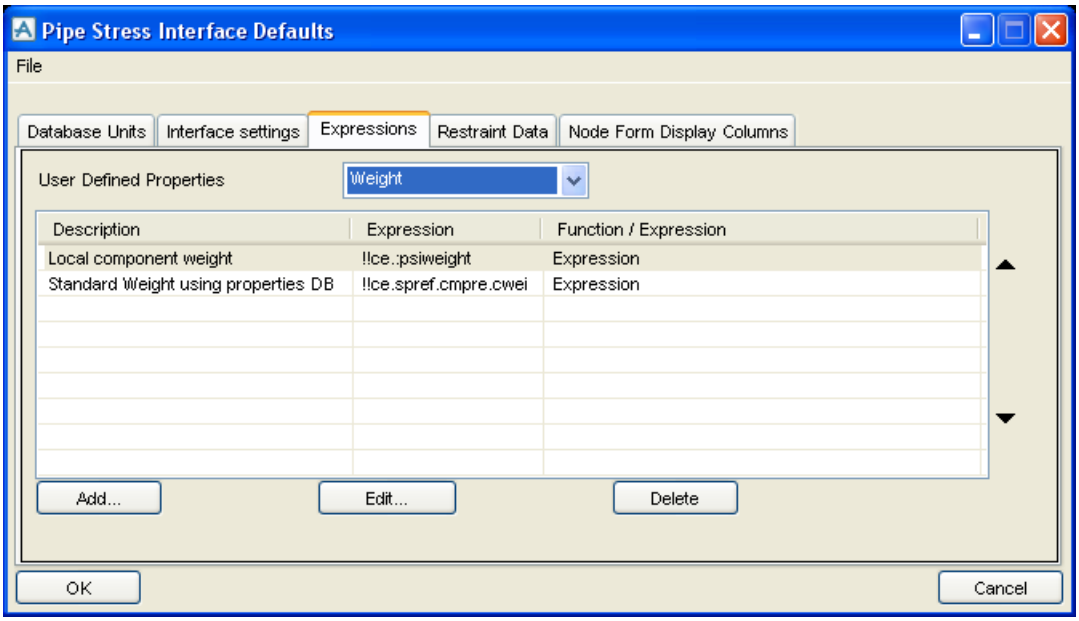
Insulation Density

Ignored Components

Stress Ready

Temperature1 to Temperature9

Pressure1 to Pressure9



Expression Settings

Expressions may be in the form of a simple pml expression or alternatively a pml function. In all cases except temperatures and pressures, there may be multiple expressions which are tested in sequence until a non zero result is obtained.

Expressions are evaluated after navigating to the component for which data is required, so the variable !!ce may be used freely in expressions.

Functions are pml2 functions defined with a fixed set of parameters. For example wall thickness is defined as a function which gets the current tube wall thickness or alternatively the wall thickness from the current component. When a function is called, it called from an object called a stressgroup which has a member !this.node which represents the ppoint for which information is being gathered. The member !this.node is a stressnode object which has members:

- .bore
- .Component
- .ComponentType
- .Connection
- .ConnectionType
- ..Position
- .wallthickness
- .point
- .OutsideDiameter
- .insulationmat
- .insulationthickn
- .material
- .weight
- .ltubelength

- .bendradius
- .AlphaAngle
- .temperature
- .pressure
- .sif
- .restlines
- .caselist
- .modified
- .restraint
- .Type
- .Stiffness
- .Gap
- .Friction
- .load
- .deflection
- .rotation
- .status
- .bendtype

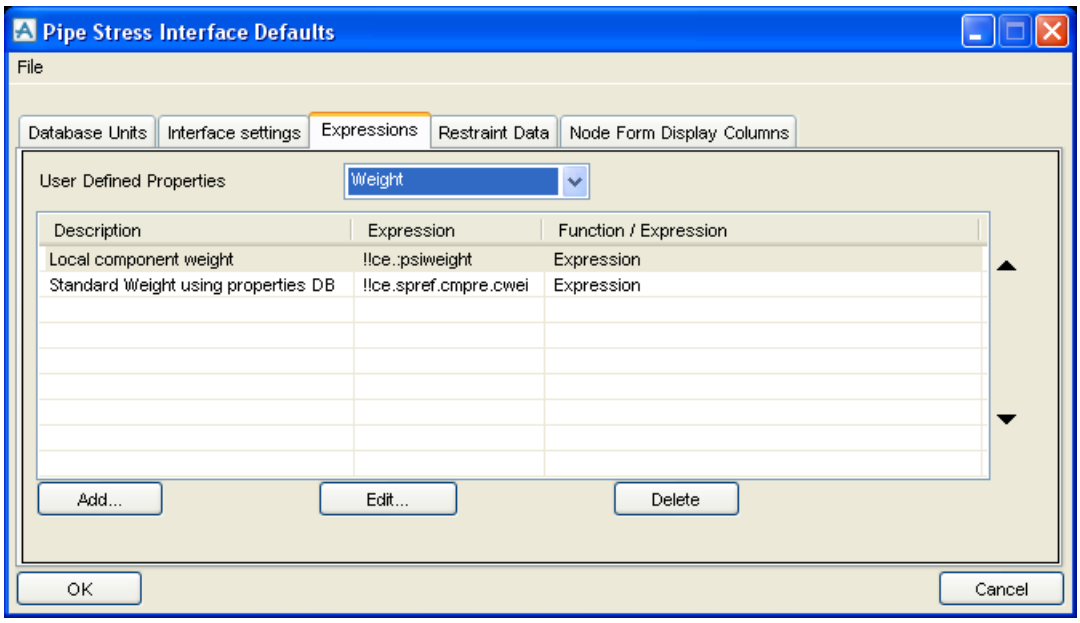
All of these may be used by the user function provided they are passed to it as shown below:

!!psifindwall(!this.node.component,!this.node.point)

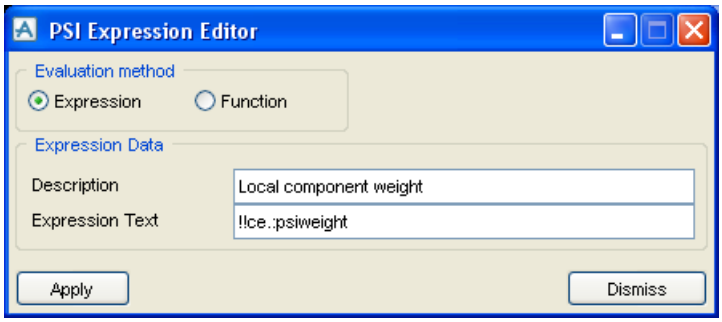
Here the parameters are a dbref and current ppoint.

Note: !this.node.component could have been replaced by !!ce as the system would be at that component at the time it was run.

3.3.1 Creating and Editing Expressions



Expressions may be created edited or deleted by the **Add**, **Edit** and **Delete** buttons on the **Pipe Stress Interface Defaults** form. Adding and editing expressions is done on the **PSI Expressions Editor** form:



The evaluation method is either an expression or a function. For expressions the expression text can be tested by entering it on the command line in Design whilst at a suitable component. The description is simply a text string to describe the expression.

Note: The order in which the expressions are evaluated may be changed using the Up/ Down arrows on the side of the **Pipe Stress Interface Defaults** form.

3.3.2 Expression Details

Weight

The defaults form comes with preset expressions and functions, as can be seen for weight. The PML expression **!!ce.:psiweight** for example assumes you are at a piping component as the expression is based on the current element (!!ce), so this expression get the weight from the **psiweight** attribute of the current element.

Wall thickness

The function to determine the wall thickness is worth noting:

!!psifindwall(!this.node.component,!this.node.point)

The function called *psifindwall* uses an object called a stress node **(!this.node)** which is the current node and is passed to the function.

The functions are stored in …\pmllib\design\functions and are called *.pmlfnc. To find where your PSI pmllib folder is, type *q evar pmllib* on the command line. Any new PML functions need to be saved in here.

Material

Materials are determined by the following expression **:matnum of matxt of spref of hstu of ce**

- *:Matnum* is a UDA (user-defined attribute) which is added to SPCO's
- *Matxt* is the text describing the material from which the component is constructed
- *Spref* is the component specification reference
- *Hstu* is the head tube specification
- *Ce* is the current element

So it obtains the material from the material text of the specification of the head tube of the current element.

Temperature and Pressure

These are determined by the expressions **Press of bran** and **Temp of bran**, which simply get the temperature and pressure values at branch level.

Fluid Density

The Fluid Density is determined by the following expression **Density of sden1 of tden1 of flure of pspec**

- *Sden1* is an element of the fluid reference (Spot density)
- *Tden1* is an element of the fluid reference (Table of density)
- *Flure* is the fluid reference
- *Pspec* is the piping specification

So it obtains the density by looking at the piping specifications fluid reference, and then looks at the spot density from its table of density.

Insulation density is obtained in a similar way to Fluid Density and both relate to values at branch level.

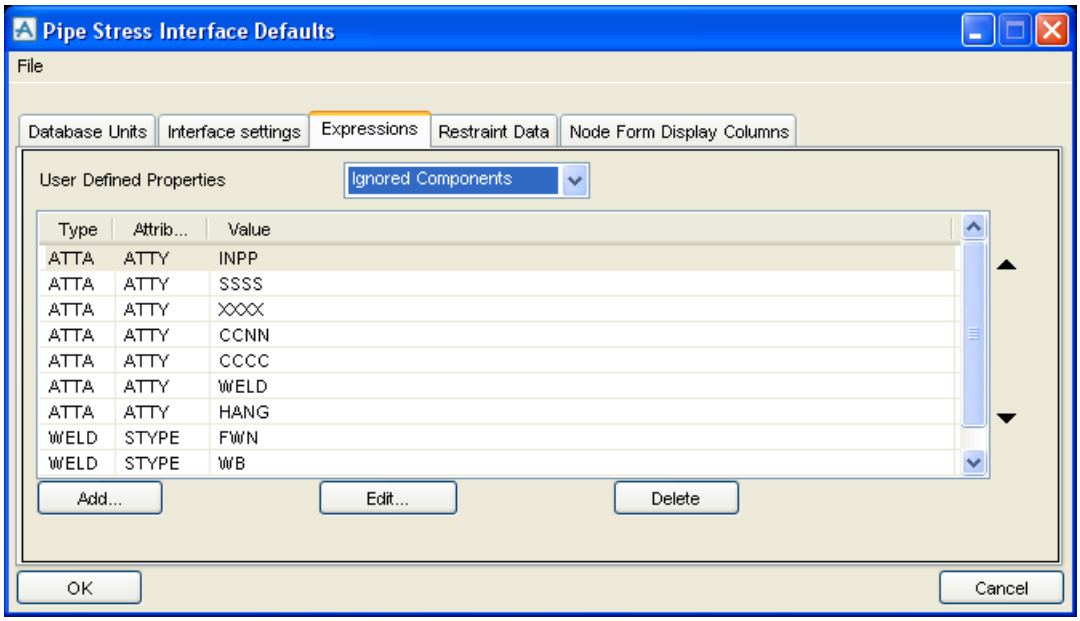
SIF

SIFs are determined by the following expression **rating of cate of catref of ce** which means it looks for SIF values at the rating of the category of the catalogue reference of the current element. The rating refers to a number which corresponds to a CAESAR intersection type code (1-17, where 0 is unset). These will need to be setup in the catalogue.

Ignored Components

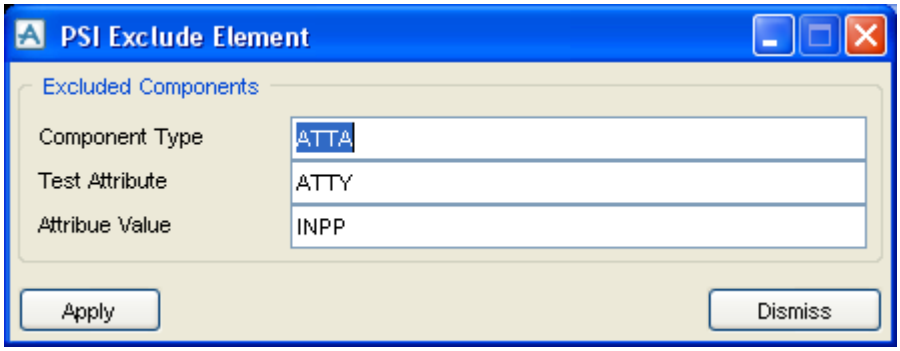
These settings control which components **are not** sent to CAESAR for analysis. For example support ATTA's are defined by the setting of their ATTY attribute. Insulation symbol

ATTAs have ATTY set to INPP (insulation personnel protection) and Split ATTAs for system isometrics have ATTY set to SSSS. As seen below, ATTA's with their ATTY set to INPP and SSSS are not sent to CAESAR as they are in the Ignored components list. Item can be added by pressing the **Edit** button and removed by pressing the **delete** button.



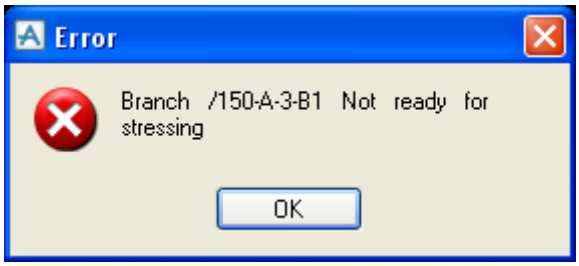
Ignored components are a selection rather than an expression and they are evaluated by checking the components attributes against an attribute value. For example the first ignored item in the list is an atta with the atty attribute set to INPP. Each ignored component type needs to have a separate entry on the form

The add and edit form is as shown below:



Stress Ready

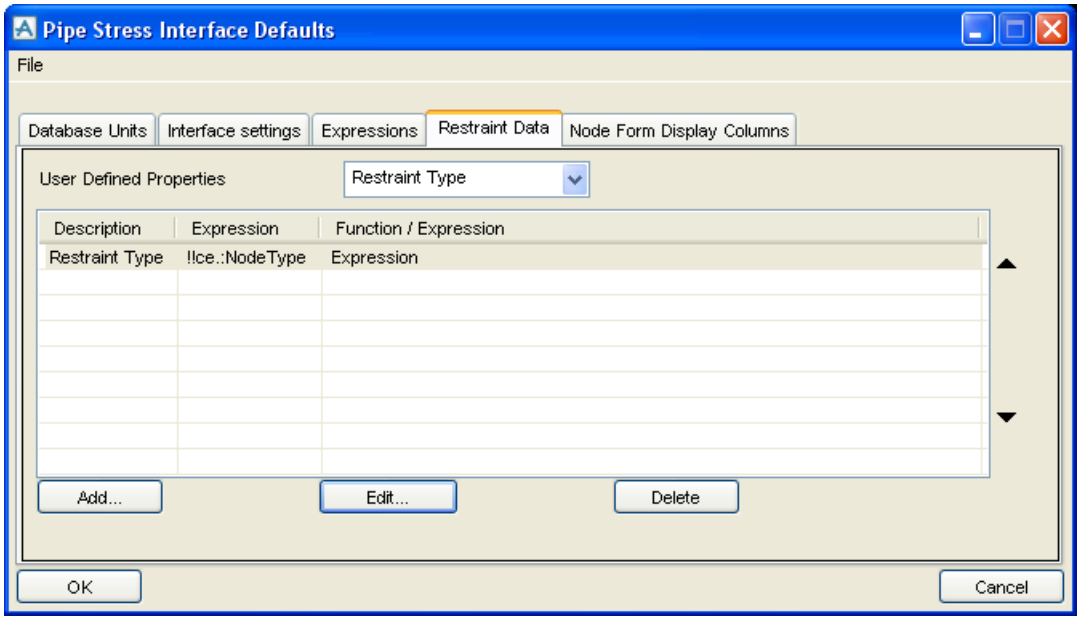
The Stress Ready setting controls when a stress group is ready to be added to a stress group. It can be turned on/off in the Interface Settings. The default expression for this is **!!ce.lissue.** This means that when a branch is added to a stress group it is checked that **!!ce.lissue = true** If the branch is not ready for stressing then a message will be displayed to the user and it will not be added to the stress group.



An existing expression can be changed by pressing the **Edit** button or a new one can be created by pressing the **Add** button. Multiple expressions/functions are allowed for the same property. If multiple expressions/functions exist then the first expression/ function is checked to see if returns a value if it does then it uses this value, if it doesn't it tries the next one and carries on until it finds one. The order of precedence can be changed by using the arrows(st) which move expressions up or down. If no value is found then it is set to zero. Expressions can also be removed by pressing the **Delete** button.

Note: For more information, refer to the Outfitting PROPCON and PARAGON Reference manuals.

3.4 Restraint Data



A restraints type, stiffness, gap and friction coefficient are set by four UDA's (User Defined Attributes):

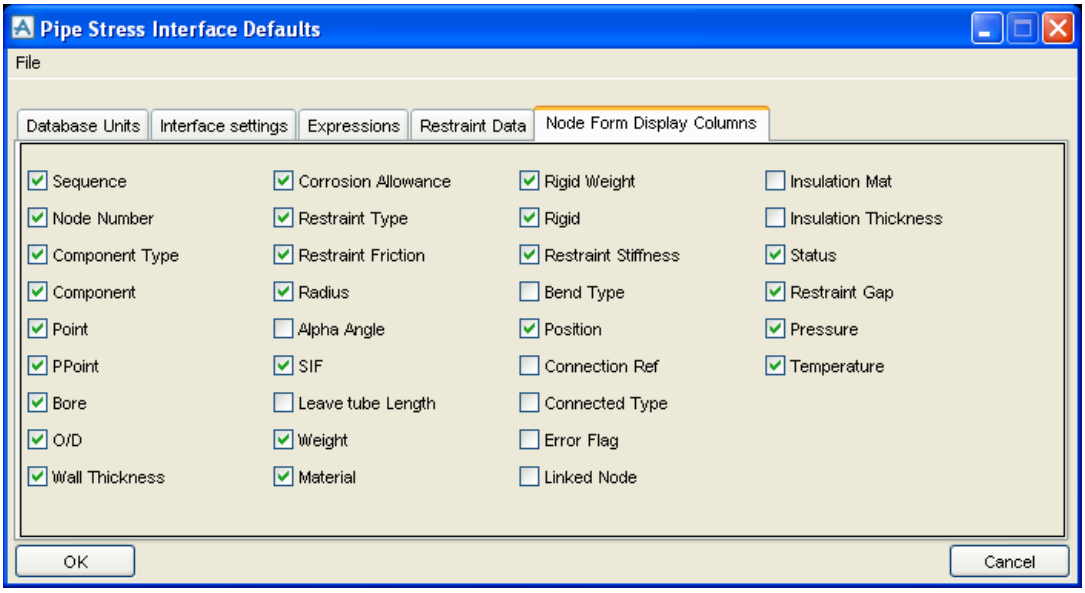
- **NodeType**
- **Nodestiff**
- **Nodegap**
- **NodeFriction**

!!.ce.:NodeType should return an integer value in the range 1-62, which represents one of the 62 restraint types in CAESAR. E.g. Type 1 is an Anchor (ANC).

Stiffness, gap and friction should also be integer values.

Note: For more information and a full table of possible restraint types please see the CAESAR II technical reference manual.

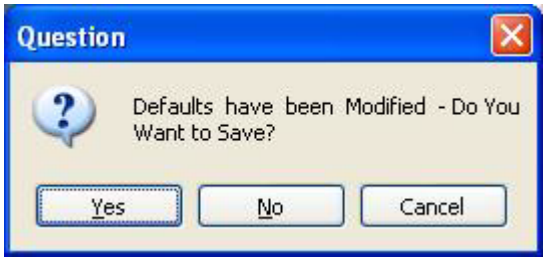
3.5 Node Form Display Columns



The **Node Form Display Columns** tab contains a number of check boxes which control what columns are shown in the grid on the **Node Details** form. Ticking the check box enables the relevant column and un- ticking it disables it from view.

3.6 Saving Changes to the PSI Defaults

If the defaults from has been changed and the **OK** button is pressed on the main PSI Defaults form then you are prompted to save the changes made.



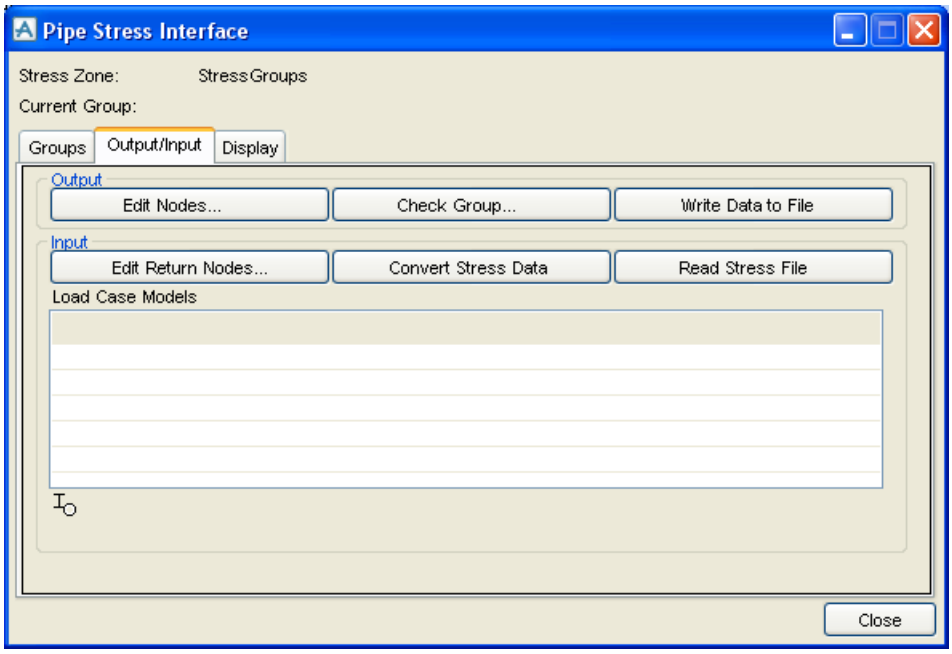
Alternately, you can save the defaults at any time by selecting **File >Save Defaults** from the main menu on the **PSI Defaults** form.

4 External Programs and Log Files

4.1 Overview

The PSI application writes data to an xml file which is converted to and from Caesar format by some external programs. These are called automatically by PSI and generate their own diagnostic data and log files. These log files are located in C:\ProjectPath\Project\Sampsi\logs where the name of the project is **Sam**.

4.2 Logs Created from and to PSI



The **logs** folder contains HTML reports viewable via a web browser. A report is generated when a CAESAR II Neural file is created i.e. when the **Write data to cii file button** is pressed and when data is being brought back into PSI from CAESAR II i.e. when the **Convert Stress Data button** is pressed.

The HTML report which is produced during the creation of the Neutral file is called *_log.html. This contains information about the conversion from the *.xml file to the Caesar II Neural File. The _log.html contains version information for the .cii converter, and also any messages to do with the running of the conversion process .It also contains information about the Outfitting elements translated, and any assumptions and default values used (for

example missing weights or wall thicknesses). The stress engineer may find it useful to have a look at this file.

The HTML report which is produced during the creation on the return XML file is called *_RetLog.html. This contains information of the conversion process from a Caesar II Neutral file, Stress Report and original XML file to a new XML file. The log gives information on what the conversion process has found and where, if any problems have occurred. Any Errors will be highlighted in Red, Warnings in Blue and Restraints in green.

These logs are a good place to look if anything goes wrong.

4.3 CAESAR Axis System

PSI can be configured to output to CAESAR using two axis systems. the default is to match the PDMS axis system of Z is U and Y is N using PDMS world coordinates.

Alternatively PSI can be configured to suit the conventional CAESAR axis system where Y is U. To change from the default of Z is U. This configuration is defined by setting an appropriate template file as described below:

4.3.1 Template Files

The process to convert a PDMS model to a CAESAR CII file is done by an external program psi2cii.exe in the PSI core directory. This uses template files to determine the configuration of units and axes to be transferred to CAESAR. The configuration is defined by editing the file convert_to_cii.bat in the PSI core directory to change the uncommented line containing **set Template** to the required template file.

This file has the following lines:

rem choose one of the following templates:

rem set Template=%CoreLoc%\mmtemplate.cii

rem set Template=%CoreLoc%\mmtemplate_yup.cii

rem set Template=%CoreLoc%\inchtemplate.cii

rem set Template=%CoreLoc%\inchtemplate_yup.cii

set Template=%CoreLoc%\mmtemplate.cii

For mm projects use a mm template and for imperial projects use an inch template.

To configure the axis system to Y is U use yup template

Index