

Argus ONE

New Features Supplement
For version 4

Information in this document is subject to change without notice and does not represent a commitment on part of the Argus Holdings. The software described in this document is furnished under a license agreement or nondisclosure agreement.

Information in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Argus Holdings.

© Copyright 1994-1997 Argus Holdings, Limited. All rights reserved.

Argus ONE and MeshMaker are trademarks of Argus Holdings Ltd.
All other products or service names mentioned herein are trademarks of their respective owners.

This product was developed using Neuron Data Open Interface Elements.

Sales and Services offices at:

Argus Interware, Inc.

11 Merit Lane

Jericho, NY 11753

USA

Argus Open Numerical Environments Version 4

The many new features and capabilities introduced in Argus Open Numerical Environments (Argus ONE) version 4 are described in detail in this supplement. We suggest that you go through the table of contents and locate items that may relate to your work.

Contents

Argus Open Numerical Environments Version 4.....	3 s4
Contents	3 s4
Compatibility Issues	4 s4
Revised Modules Structure.....	4 s4
Improved Installer for MS Windows.....	4 s4
Support for MS Windows 3.x Discontinued	5 s4
Registration of Argus ONE Version 4.....	5 s4
User Interface Enhancements	6 s4
New Data Layers Capabilities	14 s4
Grid Rotation	16 s4
Miscellaneous New Features	21 s4
New Functions.....	30 s4
New Functionality in Information Type Layers.....	35 s4
New Maps Layer Capabilities	36 s4
Values Floaters	41 s4
Values Floaters and Charts.....	44 s4
New Export Script Commands.....	48 s4
New and Enhanced PIEs.....	53 s4

Compatibility Issues

Full backwards compatibility exists between version 4 and version 3. This means that you can open Argus ONE files created with older versions in Argus ONE version 4. However, you can not open files which will be saved under version 4 with earlier versions.

We suggest that until you are certain that all functions operate as you were accustomed to in earlier versions you save backup copies of your files created with the version you have been using prior to upgrading.

Also, if you have enough disk space, do not erase your previous copy until you are certain that all is fine with results you get from version 4.

Revised Modules Structure

Argus ONE module structure has been revised in version 4. The two new modules which have been introduced are the GIS and the Programmable Export modules. The GIS module includes all non mesh and grid layers, such as Information, Data, Maps, and Domain type layers and all related functions and capabilities. The GIS module is required for the operation of all other modules.

Export template creation, editing and export by template have been grouped under the Programmable Export module to support future enhancements of programmable export capabilities for all GIS layers. Programmable export template capabilities for all mesh and grid modules are still distributed with those modules as before.

Currently the following modules are available:

- GIS
- Grid - Finite difference Grids
- Tri Mesh - Triangular Finite Element Meshes
- Quad Mesh - Quadrilateral Finite Element Meshes
- Programmable Export

Improved Installer for MS Windows

A new installer for MS Windows 95 and NT has been adopted. The installer by InstallShield allows you to select the components you wish to install, to easily select the directory in which you want to install Argus ONE and also supports automatic uninstalling of the application.

The new installer does not erase files or folders you added yourself under the Argus ONE folder but only replaces identical files and folders. Thus, if you have installed PIE files or Export templates, or added folders under the Argus ONE folder they will remain intact when you upgrade your most current copy of version 4 to later revisions or versions.

Support for MS Windows 3.x Discontinued

As of version 4, Argus ONE for PC will be supported only on MS Windows 95 and NT. Since all of the Microsoft development environments used by Argus are not supporting non 32 bit clean code development we can no longer support 3.x versions of MS Windows. Files created with Argus ONE on MS Windows 3.x versions are still supported and can be opened with Argus ONE version 4 running on MS Windows 95 and NT.

Registration of Argus ONE Version 4

If you are upgrading from an earlier version to version 4 you need to renew your registration with Argus. If you purchased the upgrade, you will be sent new registration and activation codes automatically. You will then have to type them in the Registration Information Dialog which can be accessed from the About menu under the Help menu.

User Interface Enhancements

Many components of the user interface have been enhanced and new ones have been added. The most notable ones are the toolbar, the tool tips, the information ruler which also shows help about tools, and the help system.

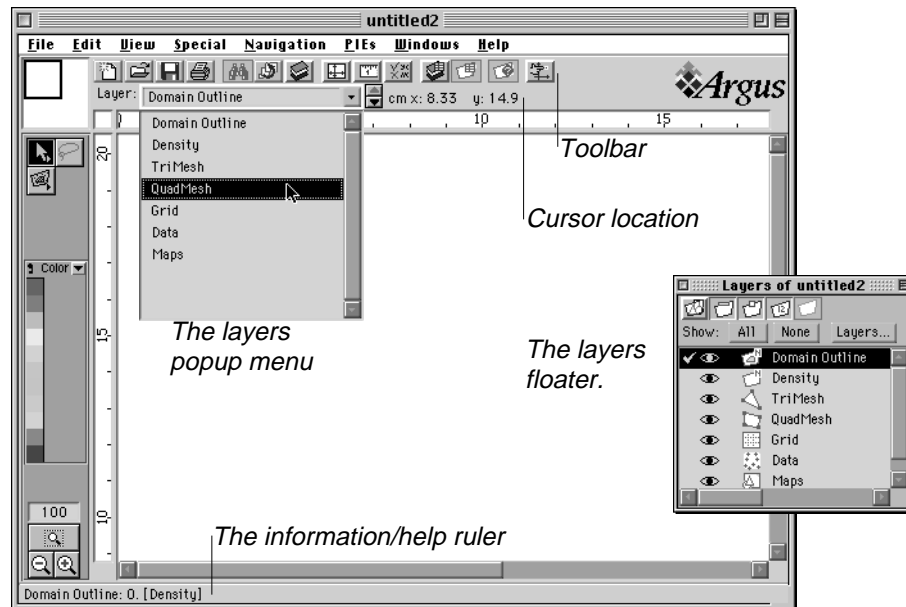
Main Window's New Components and Changed Locations

The following new components have been added to the main window:

- **PIE menu** - Located to the right of the Windows menu
- **Help menu** - Located as the right most menu on the menu bar.
- **Toolbar** - Located below the menu bar. (On the Mac below the drag bar)
- **Information/Help ruler** - Located at the bottom of all windows and below the horizontal scroll bar in the main window.
- The layers floater automatically opens when a new project is created. It also features a View toolbar and layer-selection by typing the layer's name.

The following components have been relocated and changed:

- The cursor position info line is located below the toolbar.
- The layers popup menu is also located below the toolbar and shows only visible layers.
- The information ruler is located at the bottom of the window below the horizontal scroll bar.
- Some menu items have been moved to different menus.



The Toolbar

In some layer types you might need to resize the window to see all tools.

The toolbar, located at the top of the window, contains shortcuts to the more often used menu commands. The number of toolbar icons is layer-dependent, and some icons might be dimmed (inactive) if they are not applicable.

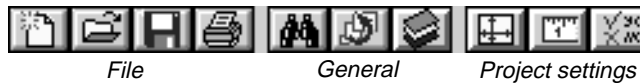
Project related tools

1. File menu commands - New, Open, Save and Print
2. General commands - Search, Rotate and Scale, and Layers dialogs
3. Project settings - Drawing size, Scale and Units, and Preferences dialogs

Layer related tools

1. Floating windows - Layers and Charting floaters
2. Info dialogs
3. Layer specific commands
4. Goto commands

Project toolbar



Information layer toolbar



Mesh layers toolbar



Grid layer toolbar



Data layer toolbar



Map layer toolbar

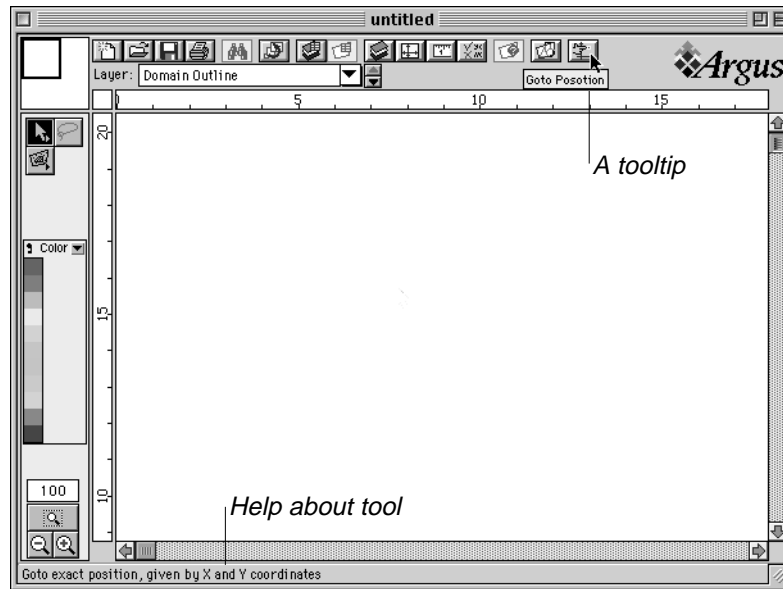
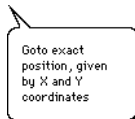


Tooltips and tools help

Tooltips and help in the information ruler are also available for tools in the tool palette and buttons in some dialogs.

When you place the cursor above a tool's icon, a tooltip appears describing the action of that tool. The information ruler at the bottom of the screen shows a more detailed explanation of the tool.

Macintosh: If Balloon Help is on a balloon will appear showing the text which appears in the information ruler.

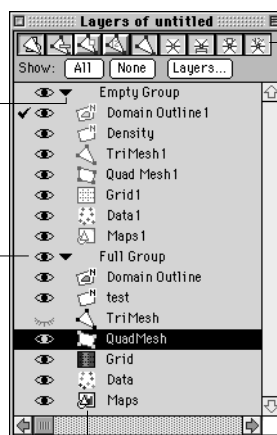


Layers' Floater New Functionality

Two new columns of icons have been added to the layers' floater. The Collapse/Expand Group icons column and the Layer Indicator icons column. A View toolbar, at the floater's top enables quick access to view commands.

Click to collapse all layers in the group.

Click the eye icon of a group layer to show/hide all of its layers.



View toolbar - Shortcuts to View menu commands.

Each layer type has a different set of viewing commands.

You can change the viewing options of a layer even if it is not the active layer.

Layer indicator - Shows: layer type, empty/full layer and interpretation type.











Layer Indicators

The layer indicators are a set of dynamically changing icons indicating the layer type and some other characteristics of the layer.

A quick glance at the layers list will reveal to you the:

- Layer type
- If the layer contains objects
- The interpretation method (Information type layers)

The following table lists the icons and their meaning.

Layer Type	Icon	Indicates ^a
Information		Interpretation method: Nearest
Information		Interpretation method: Exact
Information		Interpretation method: Interpolation
Information		Allow intersection is on ^b
Domain		Interpretation method: Nearest
Maps		
Data		
Grid		
Quadrilateral mesh		
Triangular mesh		

a. When a layer's icon is dimmed the layer has no objects. When there are objects in the layer it's icon will be colored.

b. Allow intersection is independent of the layer interpretation method and thus may appear with any of the icon variations of information type layers.

View Toolbar

The View toolbar at the top of the layers floater allows you to show and hide various attributes of objects in a layer. You can do so without first making the layer active.

To change the viewing option of a layer

1. Click the line of the layer. You do not have to activate the layer.
The toolbar will change to show the available view commands.
2. Click the appropriate tool.

Collapsing and Expanding Groups of Layers

By entering Group type layers you can group your layers into subjects. Group layers allow you to:

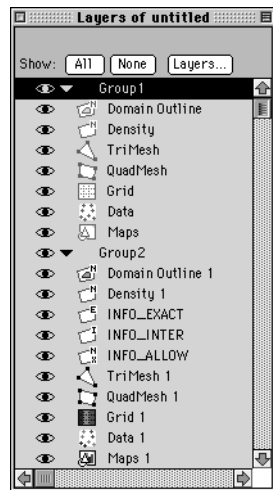
- Show and hide all layers belonging to a group with one mouse click.
- Collapse a group in the layers floater.

To show and hide all layers belonging to a group

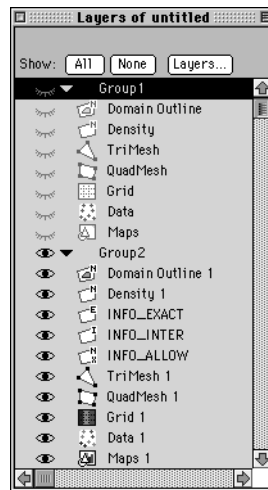
- Click the eye icon of the group layer they belong to.

To Collapse a group layer

- Click the arrow to the left of the group layer's name.

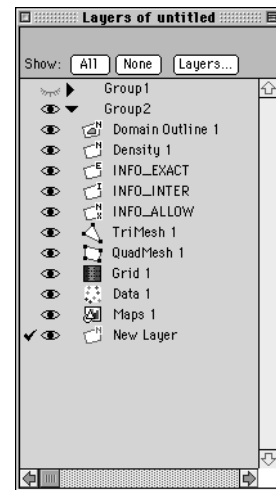


To hide all layers belonging to a group, click the eye icon of that group...



...All layers of the group are hidden.

To collapse a group, click the arrow next to that group...



...All layers belonging to that group are collapsed.

Selecting a Layer by Typing its Name

You can type the first letters of a layer's name and it will be selected. This is especially convenient when you have many layers.

To select a layer by typing its name

1. Type the first (or as many as you can) letters of the layer's name.
2. The selection will move to the first layer whose first characters match the ones you typed.
3. To select the next layer, in alphabetical order, press the tab key.
4. To select the previous layer, in alphabetical order, press shift+tab keys.

If the list is longer than the floater, it will automatically scroll the selection into view.

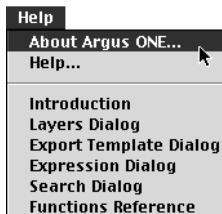
Greeting Window

The greeting window which opens whenever you launch Argus ONE disappears after a number of seconds. It can be accessed from within the Argus ONE workplace using the About Argus ONE... menu.

About Window

The About window allows you to change your registration information and also lists the following information:

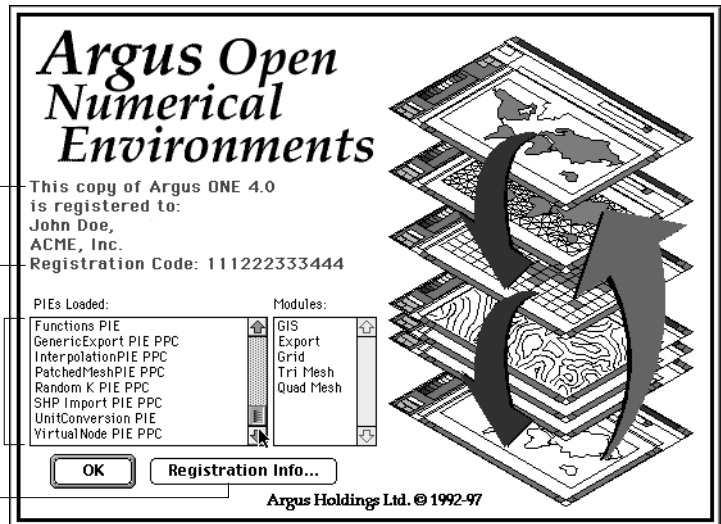
- Version number
- Name and company of the registered user
- Registration code
- Lists of the available Argus ONE PIEs and modules



Version and Registration information

Available PIEs and Argus ONE Modules

Click to change registration information

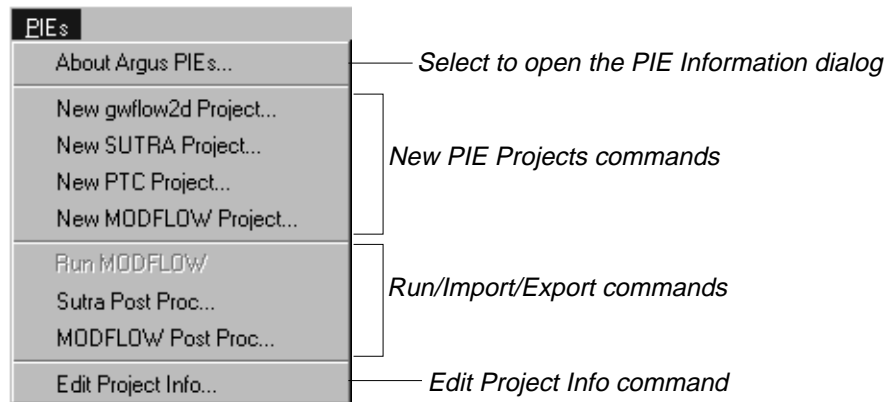


The PIEs Menu

The PIEs menu organizes menus which are created and installed by PIEs. Not all PIEs will have their menus installed under the PIEs menu. The PIEs menu also hosts the About Argus PIEs command which open a dialog listing the available PIEs and their description.

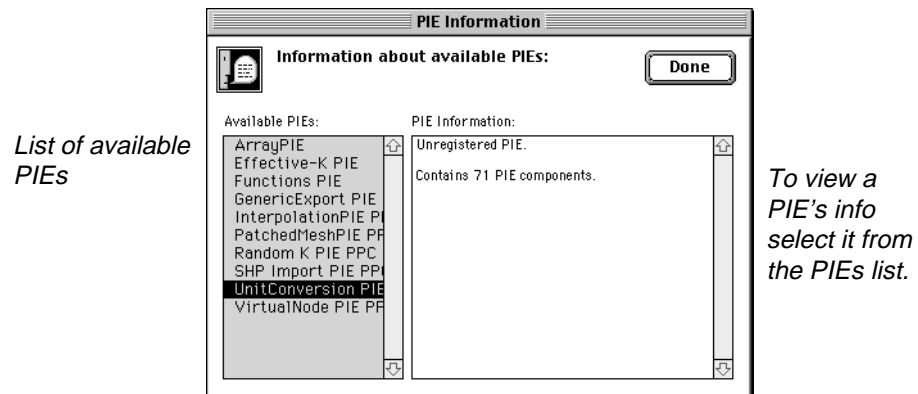
There are 3 groups of menu items:

1. The New Project group - storing all commands to create a new PIE project.
2. The Run/Import/Export group - storing menus which invoke a model, export data or import data.
3. The Project Information group - storing the current project's PIE information menu.



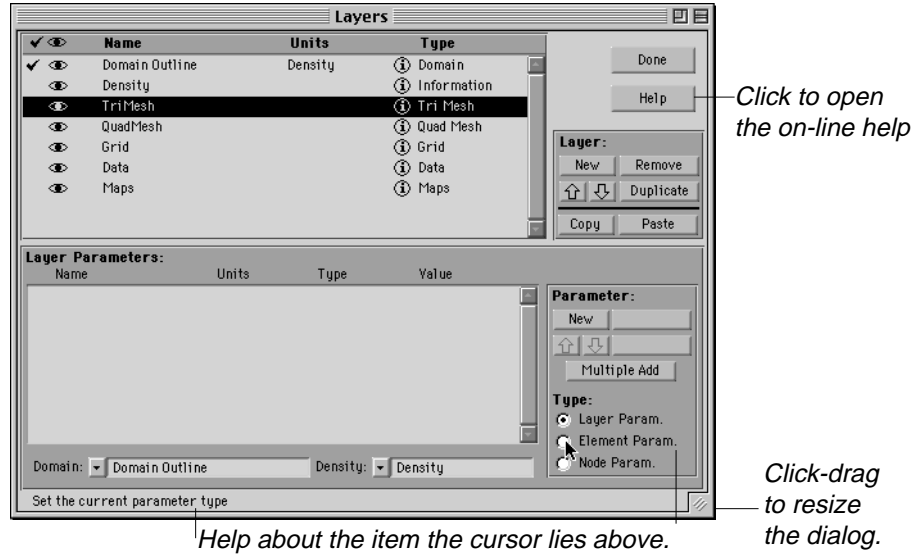
The PIE Information Dialog

The PIE Information dialog provides you with valuable information regarding the PIEs that are installed in your Argus ONE workplace. Such information might include the PIE developer, PIE name and the PIE version. The information is provided by the PIE developer and Argus can't control it.



Dialog Enhancements

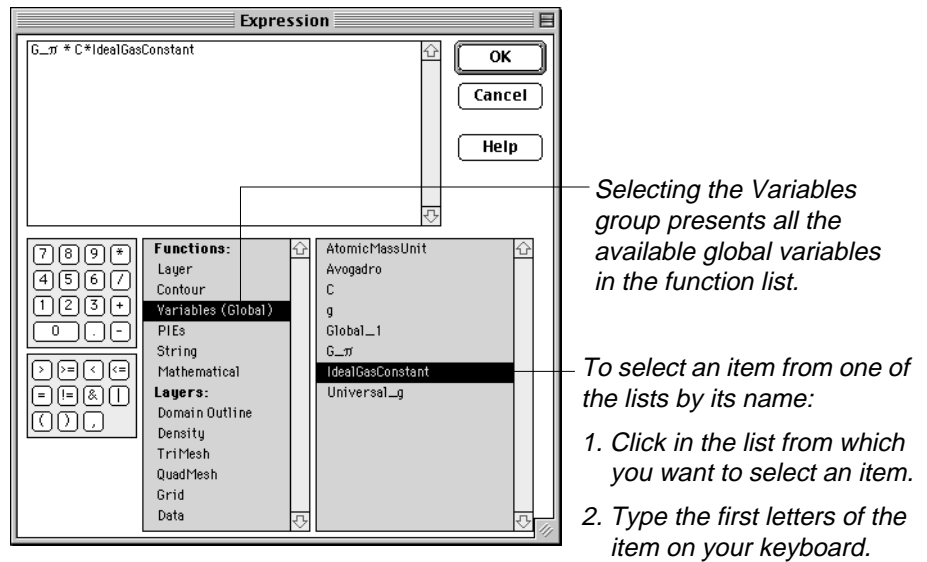
Most of Argus ONE dialogs are resizable. Some of the more complex dialogs feature tooltips for buttons, an information ruler and a help button.



Expression Dialog New Capabilities

The following two new features have been added to the expression dialog:

- You can select a function or a parameter by typing their first letters.
- Global and export template variables are accessible from the function list.



New Data Layers Capabilities

Three new features have been added to Data type layers:

- Evaluation of data layer parameters using colors.
- Reading data from grids at block centers.
- A progress bar indicating the time remaining for reading data from layers.

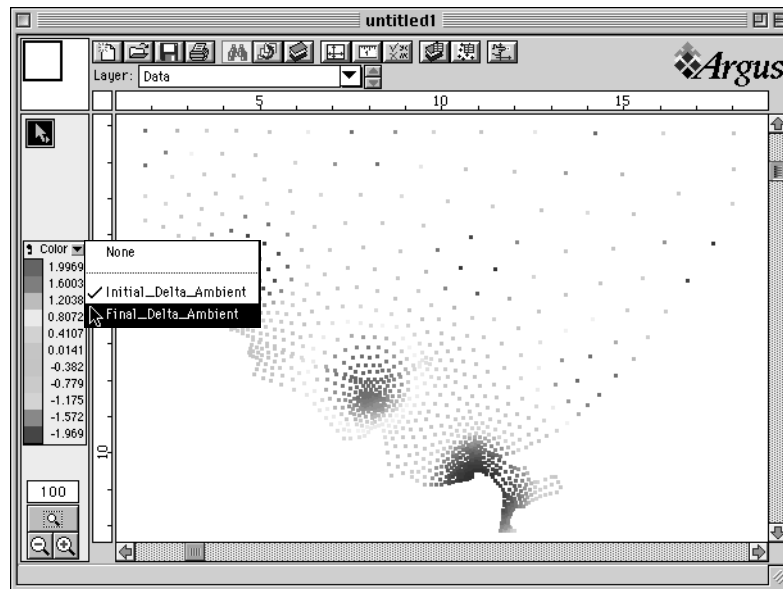
Evaluation of Data Layer Parameters using Colors

As in Information, Grid, and Mesh type layers you can now visualize the values of data layer parameters using the Evaluate by Color method. This allows you to quickly view the distribution of data.

To evaluate data by color

1. Make sure the active layer is a data type layer and that it contains data.
2. In the tool palette click and hold the Color popup menu to open the parameters list.
3. Select the parameter by which you wish to color the data points.

For a detailed discussion of how to control the color range see Chapter 1, "The color tools and legend".

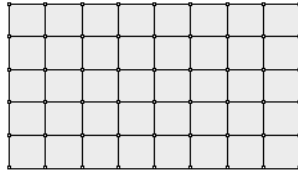


Reading Data From Grids at Block Centers

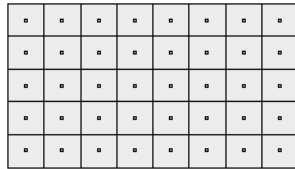
When using block centered grids the evaluation of block and grid parameters is performed at block centers. To conform with this evaluation locations, you can also read data from grids at block centers. This will also allow you to compare simulation results with inputted grids using the same locations for both. By default the data is read at block corners.

To read data from grids at block centers

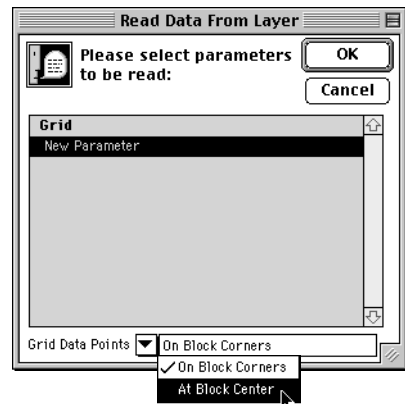
1. From the Edit menu select the Read Data from Layer... menu.
2. In the dialog select the layer and parameters their data you want to read.
3. Click-hold the Grid Data Points popup menu at the bottom of the dialog to open the menu.
4. Select "At Block Centers".
5. Click the OK button.



Data read at block corners.

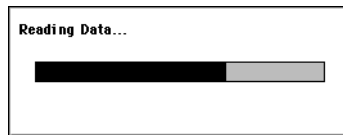


Data read at block centers.



Read Data from Layer Progress Bar

The progress bar appears whenever the product of number of grid blocks and number of parameters being read exceeds one thousand (1,000).



Grid Rotation

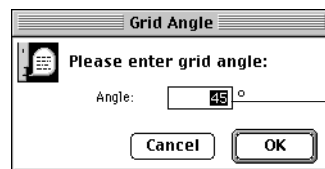
Under certain conditions it is preferable to discretize a domain using a rotated grid to minimize the number of grid blocks and thus to allow for a faster modeling cycle. Grids can be rotated using one of three commands.

Manual Grid Rotation

A grid can be manually rotated using the Rotate and Scale Dialog and by setting the grid angle.

To rotate a grid using the Grid Angle command

1. From the Edit menu select Grid Angle..., The Grid Angle dialog opens.



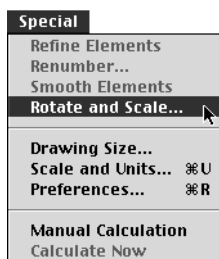
Type the grid rotation angle in degrees

2. Type in the grid rotation angle (positive and negative values are allowed).
3. Click the OK button.

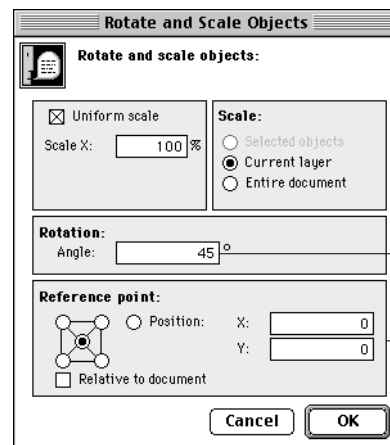
To rotate a grid using the Rotate and Scale command

Rotating a grid using the Rotate and Scale command offers greater control by allowing you to set the point of reference and to combine rotation and scaling.

1. From the Special menu (or from the toolbar) select Rotate and Scale...
2. Enter the grid's rotation angle (both positive and negative values are allowed).
3. Select the reference point for the rotation.

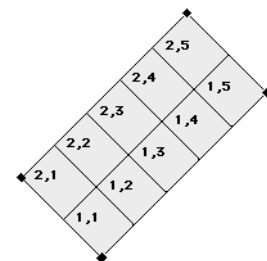


If when rotated the grid extends beyond the document's borders the drawing size and origin are changed.



Type the grid rotation angle in degrees

Select a reference point



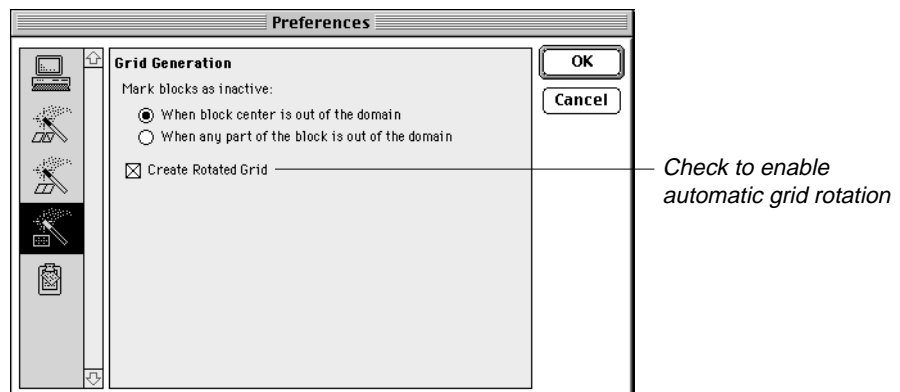
Automatic Grid Rotation during Auto Grid Generation

When auto grid generation takes place Argus ONE can automatically calculate the grid angle that will minimize the number of grid blocks and thus also the number of inactive blocks.

In projects created by version 4 or later, automatic grid rotation is the default behavior.

To enable or disable automatic grid rotation

1. From the Special menu select Preferences...
2. Click the Grid Magic Wand to select the auto grid generation preferences pane.
3. To enable automatic grid rotation check the Create Rotated Grid check box.
4. To disable automatic grid rotation uncheck the Create Rotated Grid check box.
5. Click the OK button.

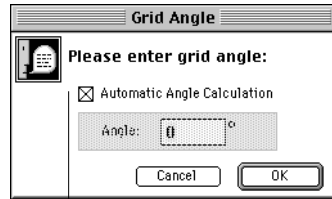
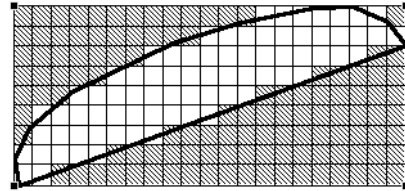


Using automatic grid rotation

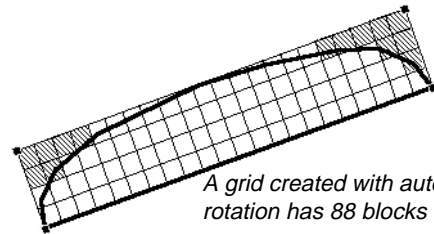
When this option is selected in the grid preferences the following dialog will appear when you click the magic wand above a domain contour to be grided.

The dialog allows you to decide whether you wish to use auto-rotation, and if not what, at which angle to rotate the grid.

A grid created without auto-rotation has 189 blocks



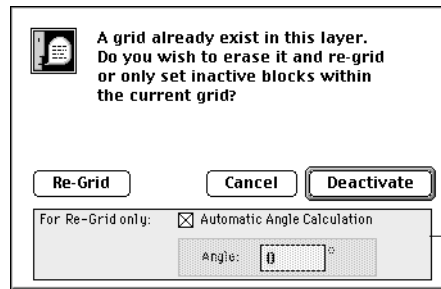
Check to use auto rotation or uncheck and specify the angle of rotation.



A grid created with auto rotation has 88 blocks

Regridding with automatic grid rotation

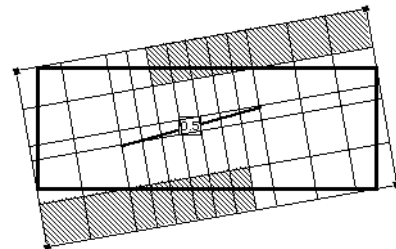
When a grid already exists and you choose to regrid the domain, the re-grid dialog allows you to select auto rotation, no rotation or manually specify the rotation angle.



Check to use auto rotation or uncheck and specify the angle of rotation.

The effect of density on auto grid rotation

Since, the auto rotation attempts to minimize the number of grid blocks, density defined in the density layer associated with the grid is also taken into consideration. Under certain geometries and grid densities it may happen that the minimum number of grid blocks will occur in a rotated grid.



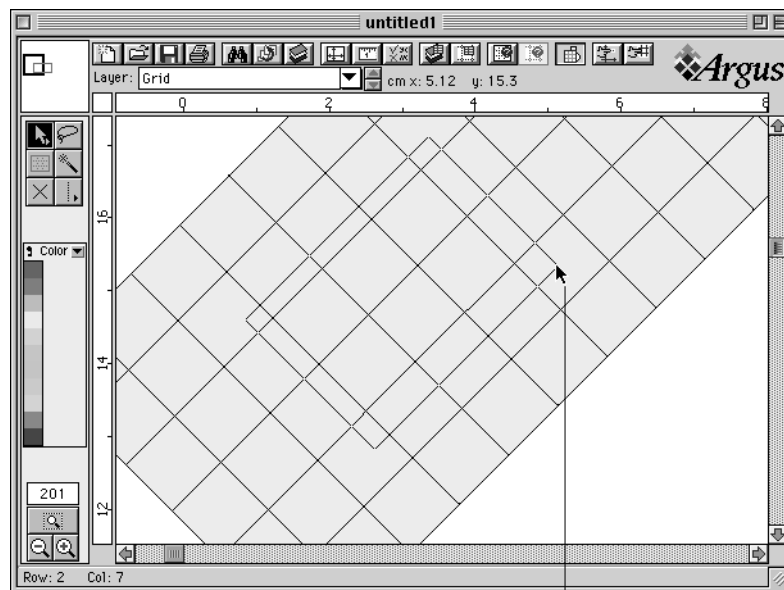
The automatically rotated grid has fewer blocks than the unrotated one.

Operation of Tools in Rotated Grids

The operation of grid tools which involve stretching a rectangle reflect the rotation of the grid by creating the stretch rectangle along the grid axis.

The following grid tools create a rotated stretch rectangle

- Magnifying glass, zoom in and out
- Arrow, select with
- Arrow, move grid line
- Delete rows and columns
- Add rows and columns



Stretch rectangles are rotated to allow better selection

Changes in Export and Import File Formats

The first line of a grid file also lists the grid rotation angle. If it is omitted Argus ONE assumes it is zero.

The first line of a grid export/import file

Number of Row	Number of Columns	Number of grid & block parameters +1	Grid Rotation Angle in radians (optional)
---------------	-------------------	--------------------------------------	--------------------------------------------------

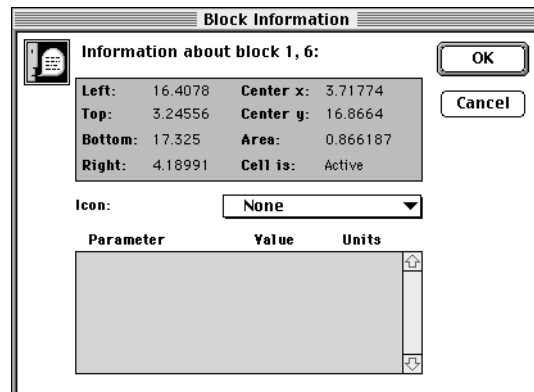
GridAngle() Function

The `GridAngle()` function returns the grid angle of rotation in radians. See description under new functions section in this supplement.

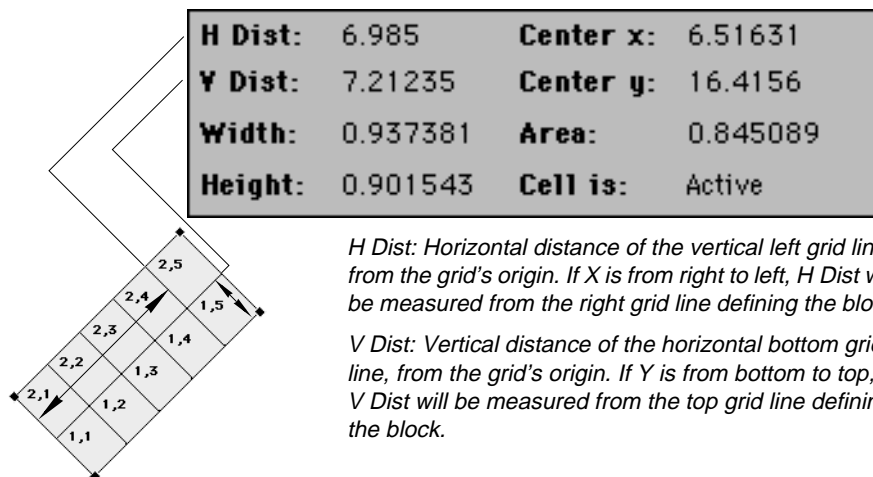
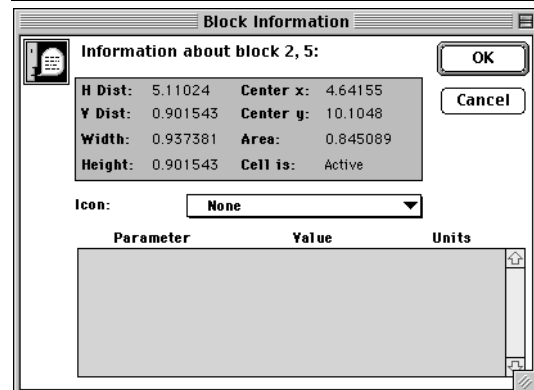
Block Information Dialog

Some of the static information listed in the block information dialog has been changed to support rotated grids. Instead of listing the locations of the grid lines defining the block, the distances of the grid lines from the grid's origin, and the block height and width are specified.

Block information dialog before version 4. Since the grid was always aligned with the coordinate system, all block information was related to the document's origin.



Block information dialog from version 4. Since the grid may be rotated with respect to the coordinate system, all block information is related to the grid's origin.



H Dist: Horizontal distance of the vertical left grid line from the grid's origin. If X is from right to left, H Dist will be measured from the right grid line defining the block.

V Dist: Vertical distance of the horizontal bottom grid line, from the grid's origin. If Y is from bottom to top, V Dist will be measured from the top grid line defining the block.

Miscellaneous New Features

Global Variables

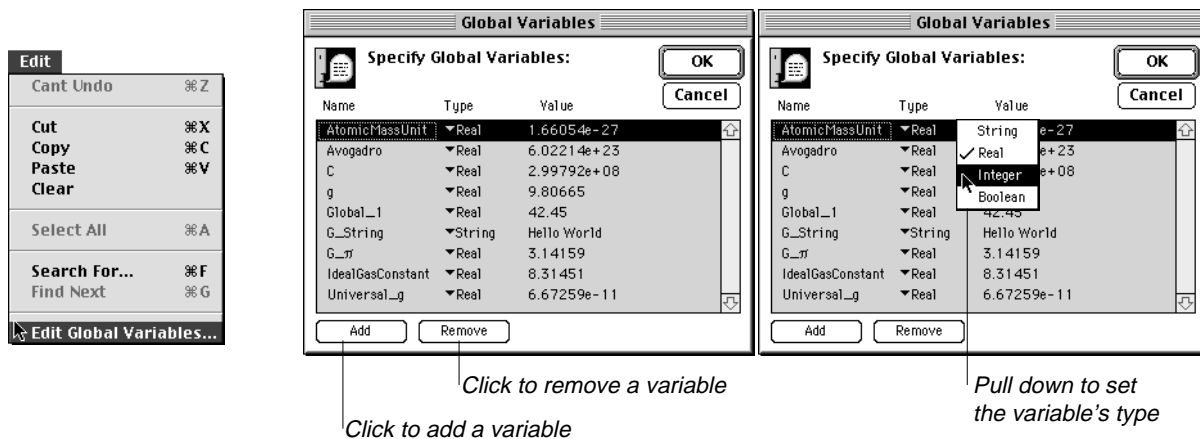
Global variables allow you to define constants which are independent of the Argus ONE layers in your project. Global variables can be of type real, integer, boolean and string. They can be accessed from the expression dialog and thus can be introduced into any Argus ONE expression, either into parameter expression or into export expression or variable.

Defining and Editing Global Variables

You define a global variable by assigning it a name, a type and a value.

To define a global variable

1. From the Edit menu select Edit Global Variables...
The Global Variable dialog opens
2. Click the Add button at the bottom of the dialog to add a new variable.
3. Click in the name text edit box and type its name.
4. Click the Type popup menu to set the variables type.
5. Click in the Value text edit box and enter the value.



To remove a global variable

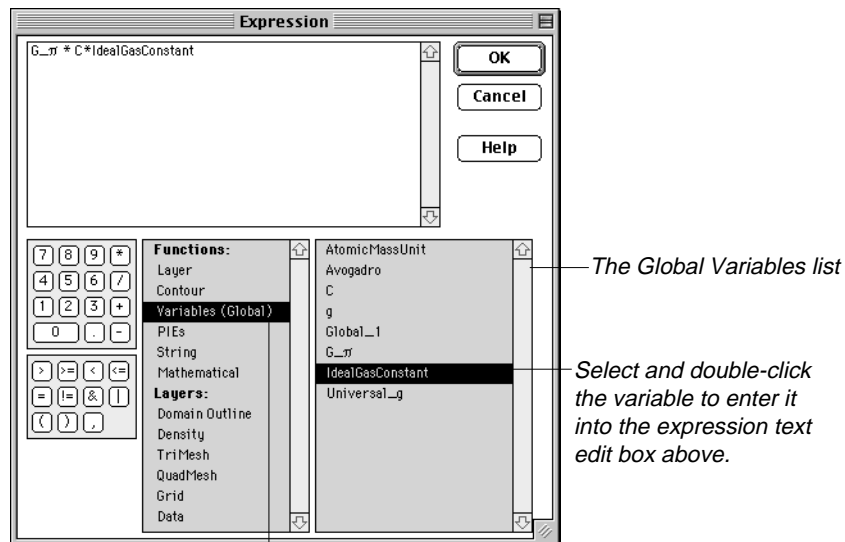
1. From the Edit menu select Edit Global Variables...
The Global Variable dialog opens
2. Select the line of the variable you want to remove.
3. Click the Remove button at the bottom of the dialog.

Using Global Variables

Global variables can be accessed from the expression dialog, both when it is invoked from the layers dialog and when it is invoked from the Edit template dialog. The expression dialog groups all global variables under a group named Variables (Global) to differentiate them from export template variables.

To use a global variable

1. Open the expression dialog.
2. In the Functions and Layers list select the Variables (Global) group.
3. In the right list double click the variable you wish to enter to the expression or type its name.



Select the Global Variables group to view all available variables in the list to the right.

Using Global Variables in Export Templates

If you are creating your own interface to a model, using global variables is a convenient way to set your model non-distributed parameters such as model control parameters.

Group Type Layers

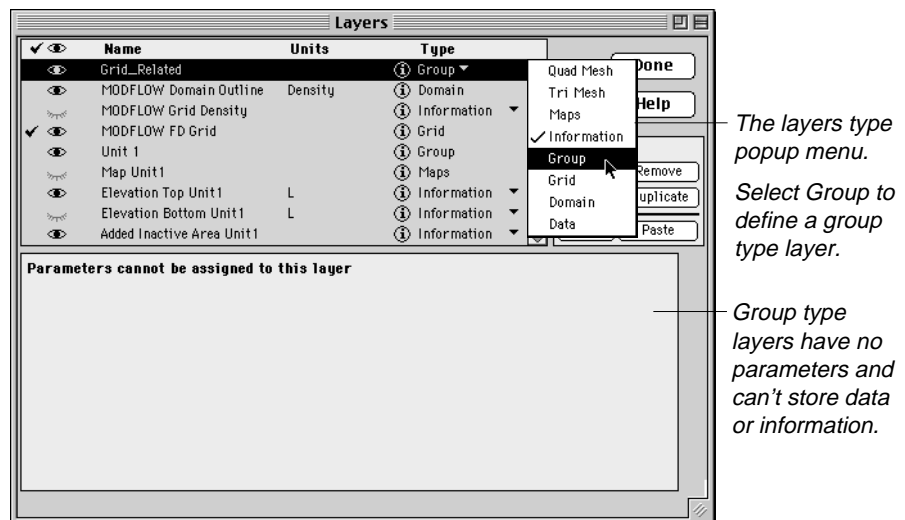
Group type layers enable you to organize any number and type of layers into groups. This allows you to hide, show and collapse all layers belonging to a group and thus better control your views. Group type layers have no parameters and can not store data or information of any type.

Creating Group Type Layers

Group type layers, as layers of all other types, are created in the layers dialog. All layers below a group type layer and up to the next group type layer belong to that group.

To create a group type layer

1. Open the layers dialog.
2. From the layers list, select the line of the layer below which you want to create a group type layer.
3. Click the New button in the layers control panel.
4. Click and hold the mouse button on the Type field to open the layers type popup menu and select Group.



For a detailed explanation on how to show and hide, and how to collapse and expand group layers in the layers floater refer to “Collapsing and Expanding Groups of Layers” on page 10 s4.

The Hand Tool

The hand tool enables you to scroll your view in the vertical and horizontal directions simultaneously.

Using the Hand Tool

If you magnify your view to a size larger than the window, you can use the hand tool to move new areas into view.

The hand tool is interchangeable with the arrow tool and although it does allow you to select objects you can not move them with it.

To select the hand tool

1. In the tool palette click and hold the arrow tool.
A popup menu opens.
2. From the popup menu select the hand tool.

To select the arrow tool

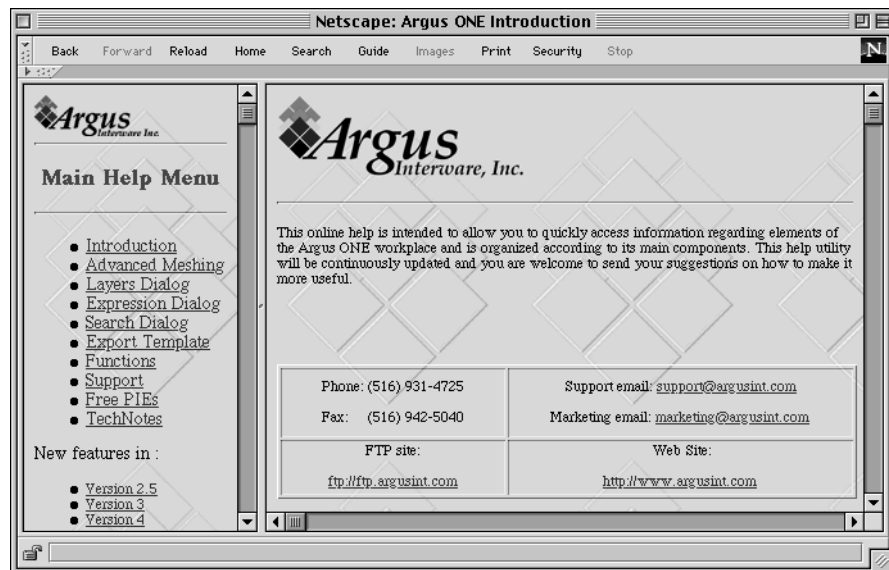
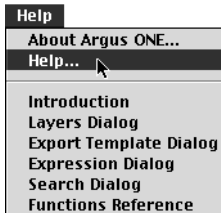
1. In the tool palette click and hold the hand tool.
A popup menu opens.
2. From the popup menu select the arrow tool.

Moving a document around with the hand tool is like moving a piece of paper around on a desk with your hand.



On-line Help

On-line help is available through the browser of your choice. Whether you use the Help menu, or click the help button in a dialog, your browser will open to present the appropriate section in the on-line help.



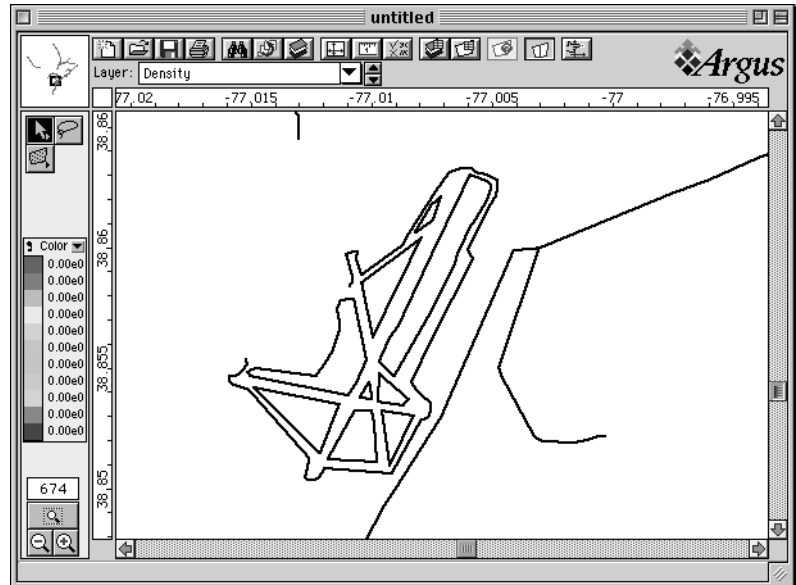
Zoom to Fit

The zoom to fit command enables you to set your view so that all objects in your document fit in the current window size.

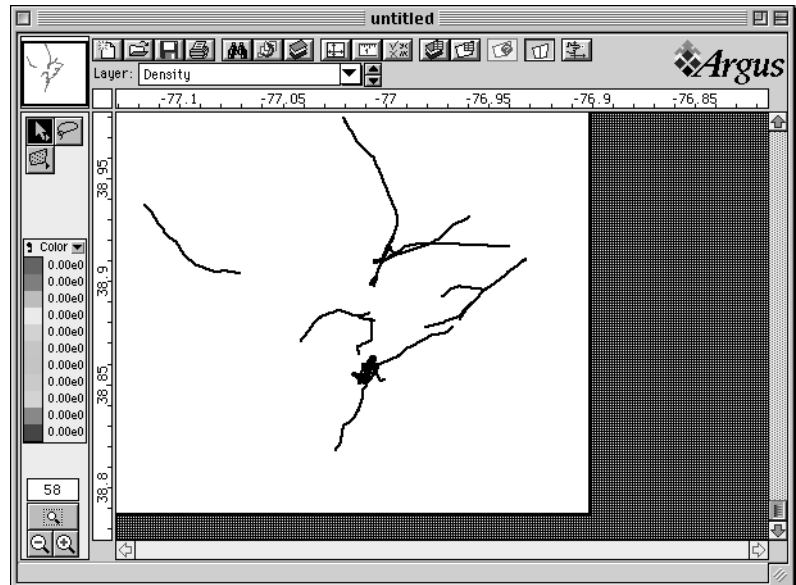
To Zoom to Fit

- From the Navigation menu select Zoom to Fit.

Not all objects fit in the current window size.



All objects fit in the current window size.



The Zoom to Fit command fits all objects in your document to the current window size, including those in hidden layers.

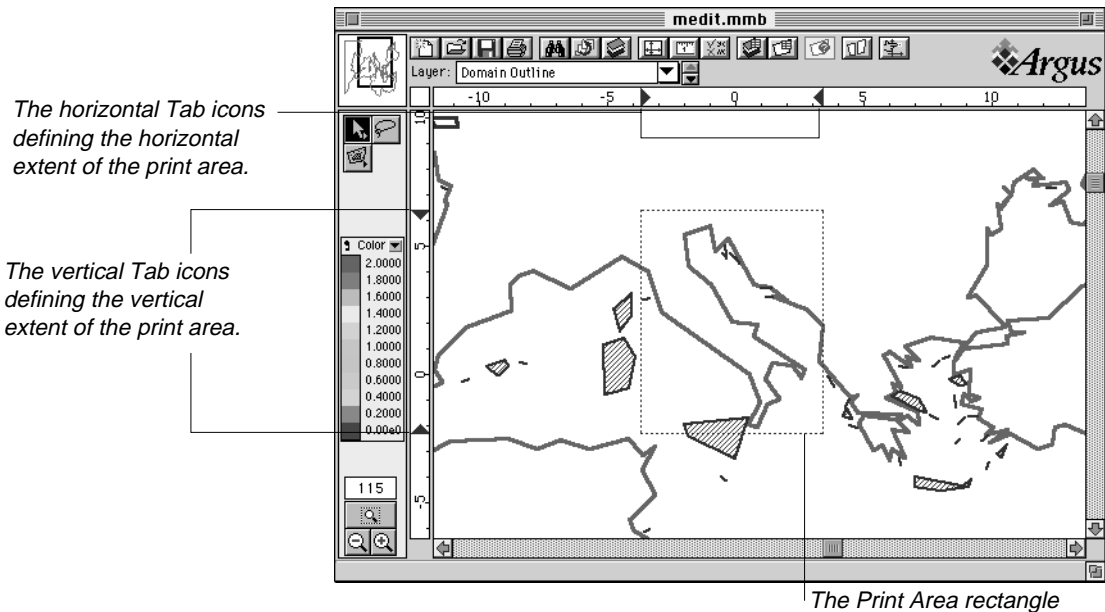
Print Area

The print area option allows you to set an area within your project's drawing size that will print when you issue the print command. The default print area is set to be identical to the drawing size.

Setting the Print Area

You can set the print area in one of two ways:

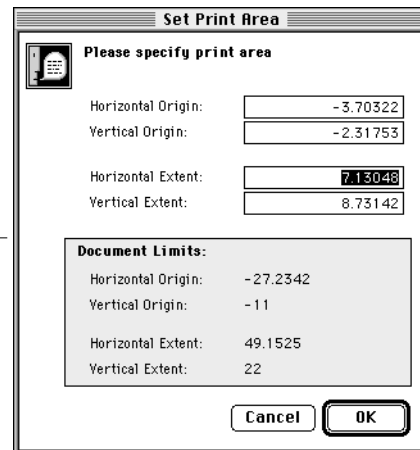
- Moving the Tab icons in the horizontal and vertical rulers.
- Using the Set Print Area dialog to define it numerically.



Pull to Show/Hide the print area rectangle.

Pull to open the dialog.

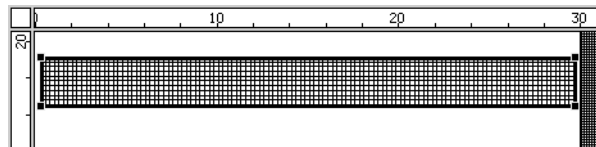
Pull to reset the print area to the drawing size.



Non Uniform Scaling

When you solve a cross-sectional problem, or a problem domain which lies along an elongated strip such as a river, one dimension of the domain would be much larger than the other. Such elongated problems present a visualization problem if the coordinate system is uniformly scaled, that is, when the scale of real world units to screen units is identical in both the X and Y axis.

Non uniform scaling is a very powerful option which enhances the Argus ONE workplace to allow one to design, work and visualize very elongated problem domains. The three Argus ONE meshing modules support this option and automatically create grids and meshes which take into account the non-uniformity of the problem domain while performing auto mesh and grid generation, thus creating elongated blocks and elements.



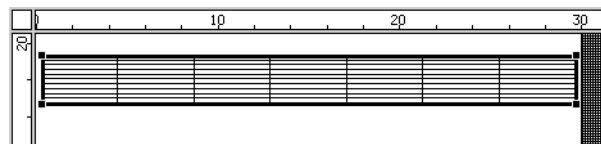
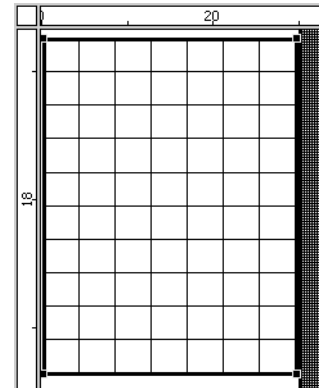
Scale is set to, 1 to 1 and the domain density is set to 0.25.

A domain of 30 units long by 2 units high is automatically grided using the above specified density and scale. The grid might be over discretized in the X direction. Also viewing the grid using zoom-in is relatively limited because of the exaggerated length.

Scale is set to, 15 (X) to 1 (Y) and the domain density is left at 0.25. When the domain is automatically grided at this scale, the X density (block side length) is reduced by the X scale ($\sim 0.25 \times 15$), while the Y density remains the same.

Two objectives are achieved:

1. The grid is less discretized in the X direction,
- and
2. Viewing, zooming and editing is as convenient as working with non-elongated domains.




You may at any time change the scale back to uniform ($X=Y$) and view your domain in the real world projection.

In the real world projection ($X=Y$) the blocks are elongated in the long axis of your problem domain.


The Scale and Units Dialog

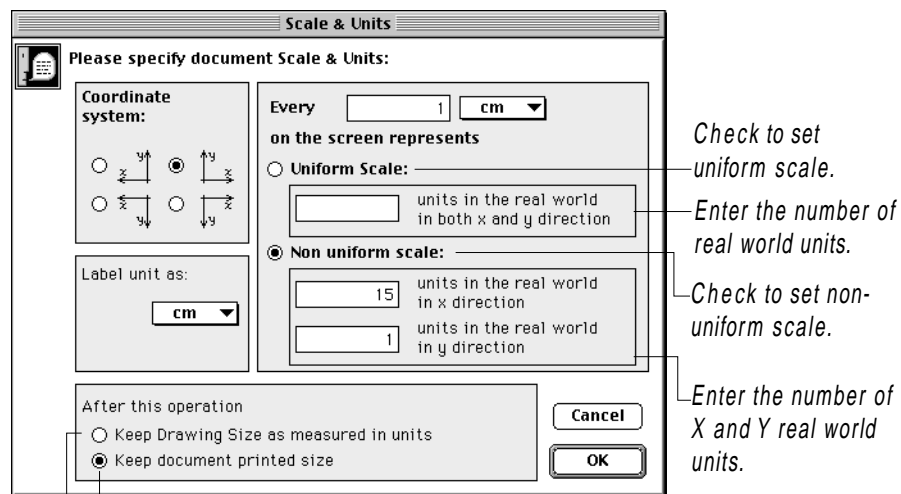
You switch between uniform and non-uniform scaling in the Scale and Units dialog.

To set non-uniform scaling

1. In the toolbar click the Scale and Units tool .
2. Check the Non Uniform Scale radio button to set scale to non-uniform.
3. Enter the number of real world units in the X direction.
4. Enter the number of real world units in the Y direction.
5. Check the appropriate radio button to keep the drawing size or the printed size.

To set uniform scaling

1. In the toolbar click the Scale and Units tool .
2. Check the Uniform Scale radio button to set scale to uniform.
3. Enter the number of real world units.
4. Check the appropriate radio button to keep the drawing size or the printed size.



Check to set uniform scale.

Enter the number of real world units.

Check to set non-uniform scale.

Enter the number of X and Y real world units.

The drawing size is scaled in both directions according to the X and Y scale specified.

The drawing size is kept as is.

Effect of Non-Uniform Scale on Auto Mesh and Auto Grid Generation

Once you scale your project using a non-uniform scale, Argus ONE scales the domain and densities specified in the domain outline and density layers and sends this scaled information (as you see it on the screen) to the auto mesh and grid generation engines. After the auto mesh and grid generation finish their job, the mesh is rescaled and sent back to Argus ONE, where it is presented in the current scale. This process automatically creates elongated elements or blocks which might be required to decrease the number of equations solved.

To use the non-uniform effect on meshes and grids

- Automatically mesh or grid a domain when non-uniform scale is on.

To view your mesh or grid as they actually appear in the real world

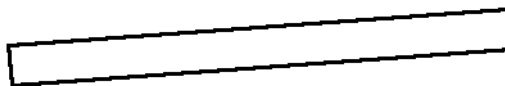
- Switch to uniform scale.

Effect of rotation on objects presented in non-uniform scale

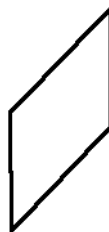
While rotated objects, when presented in uniform scale, retain angles, they do not conserve them when viewed in non-uniform scale. This is a mere visualization effect. If the scale is changed back to a uniform one, the real world is presented and it does conserve the angles.



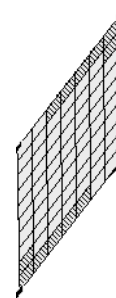
1. A domain outline contour viewed in uniform scale. (length=30, height =2)



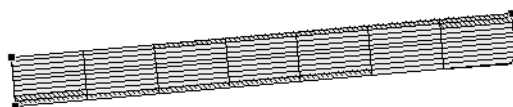
2. The same domain, viewed in the same uniform scale, but rotated at 4 degrees from X.



3. The rotated domain viewed in non-uniform scale (15:1). Angles are not preserved.



4. The rotated domain overlaid with a grid automatically generated, viewed in non-uniform scale (15:1). Angles are not preserved.



5. The rotated grid viewed in uniform scale. Angles are preserved.

New Functions

The new functions added in version 4.0 are listed here by category.

New Math Functions

<i>Name & Syntax</i>	<i>Arguments</i>	<i>Description & Return Value</i>
Ceil(<i>number</i>)	<i>number</i>	Rounds <i>number</i> (real) up to the nearest whole number.

Example:

Ceil (45.1)— Returns 46.0

Ceil (45.9)— Returns 46.0

<i>Name & Syntax</i>	<i>Arguments</i>	<i>Description & Return Value</i>
Floor(<i>number</i>)	<i>number</i>	Rounds <i>number</i> (real) down to the nearest whole number.

Example:

Floor (45.1)— Returns 45.0

Floor (45.9)— Returns 45.0

<i>Name & Syntax</i>	<i>Arguments</i>	<i>Description & Return Value</i>
Rand()	<i>no arguments</i>	Returns a random real number between 0 and 1.

New Contour Functions

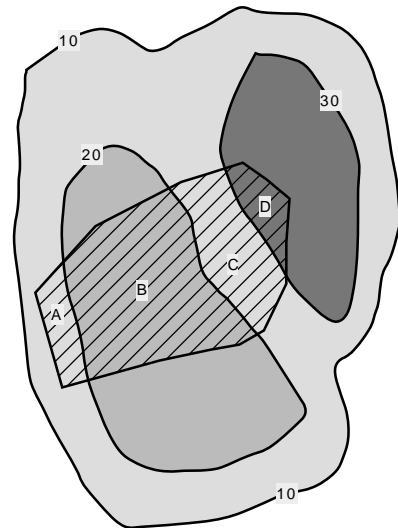
<i>Name & Syntax</i>	<i>Arguments</i>	<i>Description & Return Value</i>
EffectiveValue(<i>parameter_name</i>)	<i>parameter_name</i>	<p>The name of the parameter or layer with which contours in the current layer will be intersected.</p> <p>Returns the sum of areas created by the intersection of a contour and the contours in the layer <i>parameter_name</i> belongs to, multiplied by the respective values of these contours.</p>

Example: In the following illustration the hatched contour belongs to a layer in which one of the parameters is assigned the expression `EffectiveValue(layer_name.parameter_name)`, where `layer_name` is the layer in which the other contours reside. The hatched contour is intersected with the contours in the referenced layer and returns the value: $A*10+C*10+B*20+D*30$, where A, B, C, and D are the areas of intersection between the hatched contour and the contours in the referenced layer, and 10, 20 and 30 are the values of the contours in the referenced layer.

If in the referenced layer, Allow Intersection is turned on, then the function will return: $A*10+C*10+B*20+D*30+B*10+D*10$, since the hatched contour is also fully encompassed by the most outer contour. This would also be the case if there were additional contours fully encompassing the hatched contour.

See also `EffectiveValue()` for mesh nodes and elements, and grid blocks.

The hatched contour is tested vis-a-vis all contours in the referenced layer. Polygons A, B, C and D are created by the intersection of the hatched contour and the contours in the referenced layer. The area of each intersected polygon is multiplied by the value of the contour which intersected the hatched polygon, The sum of these products is returned.



New Grid Layer Functions

Name & Syntax	Arguments	Description & Return Value
GridAngle()		Returns the angle of rotation of the grid in radians.

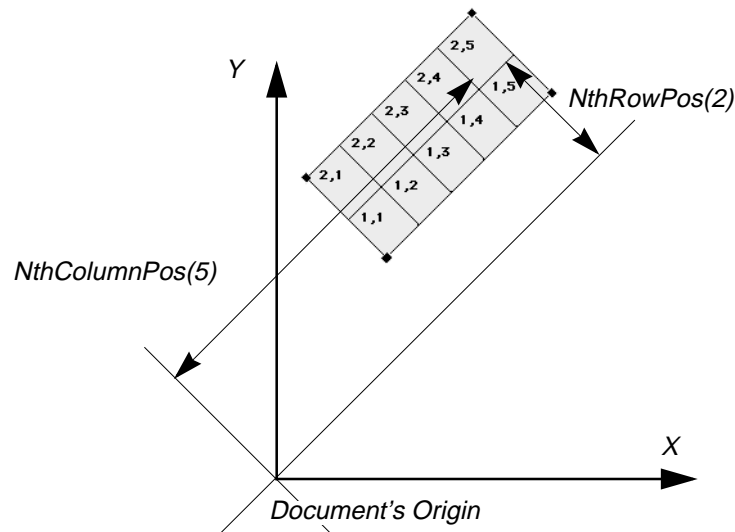
New Block Functions

Name & Syntax	Arguments	Description & Return Value
BlockGradX (parameter_name)	parameter_name	Returns the X gradient calculated as central difference (real) between the value of parameter_name at the blocks to the left and right of the current block.

Name & Syntax	Arguments	Description & Return Value
BlockGradY (parameter_name)	parameter_name	Returns the Y gradient calculated as central difference (real) between the value of parameter_name at the blocks to above and below the current block.

NthColumnPos (*col_num*), NthRowPos(*row_num*)

The two functions, described in Chapter 3, have been enhanced to return the appropriate values for rotated grids. If the grid is rotated these are measured as the distance from the document's origin along the grid rotated axes.



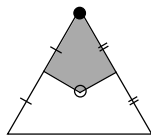
New Node Functions

Name & Syntax	Arguments	Description & Return Value
EffectiveValue (<i>parameter_name</i>)	<i>parameter_name</i>	Intersects the nodes effective area with all contours in <i>parameter_name</i> to create a set of polygons, and returns the sum of polygons areas multiplied by their respective values.

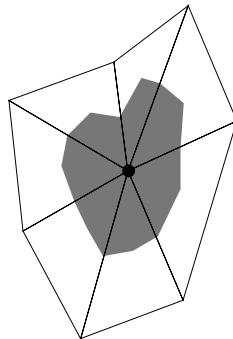
Example: In the following illustration the hatched polygon represents the node effective area. The hatched polygon is intersected with the contours in the referenced information layer and returns the value: $A*20+B*10+C*30$, where A, B and C are the areas of intersection of the node effective area (hatched) and the contours in the information layer, and 10, 20 and 30 are the values of the contours in the referenced information layer.

If in the referenced layer, Allow Intersection is turned on, then the function will return: $A*20+B*10+C*30+A*10+C*10$, this is because the hatched polygon is also fully encompassed by the most outer contour. This would also be the case if there were additional contours fully encompassing the node's effective area.

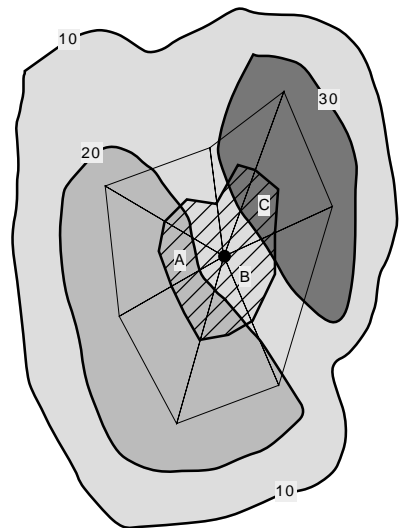
Node EffectiveValue in Triangular Elements



Node effective area of a node in a single element.

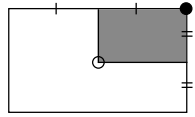


Node effective area of a node common to several elements.

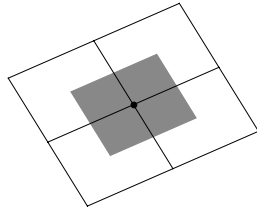


The node's effective area is intersected with all contours in the referenced layer and returns the sum of polygon areas multiplied by their respective values.

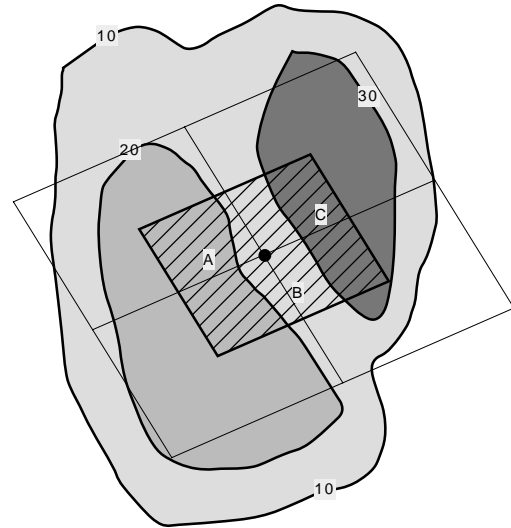
Node Effective Value in Quadrilateral Elements



Node effective area of a node in a single element.



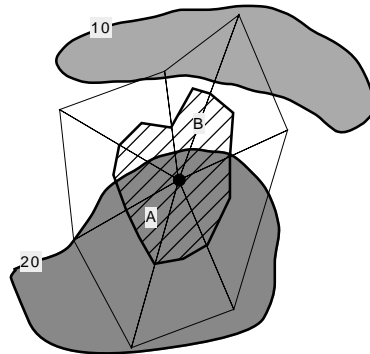
Node effective area of a node common to several elements.



The node's effective area is intersected with all contours in the referenced layer and returns the sum of polygon areas multiplied by their respective values.

Special cases: If part of the node's effective area lies outside all contours in the referenced layer, the function's return value varies with the referenced layer interpretation type. In the example below the area marked B is multiplied by:

- Nearest Contour method - The average of all adjacent contours. In this example $B \cdot (10+20)/2$
- Exact Contour method - The default value of the referenced layer.
- Interpolation method - The interpolated value at the node's location.



The area marked B, which is not intersected by a contour, is multiplied by a different value according to the referenced layer interpretation method.

See also `EffectiveValue()` for contours, mesh elements and grid blocks.

New Functionality in Information Type Layers

Information type layers now support Plug-In Extensions (PIEs) interpolators. Also, you may change an information type layer to a data type layer and vice-versa at any time.

Using PIE Interpolators in Information Type Layers

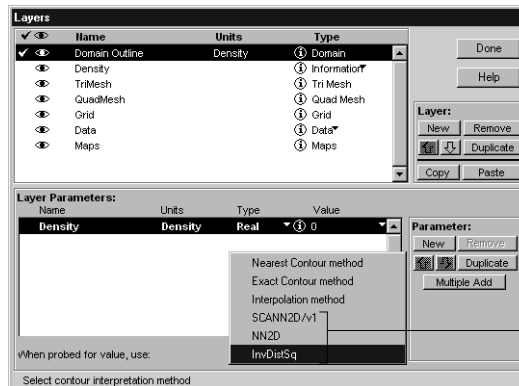
Information type layers allow for the use of an interpolator. With the introduction of Argus PIEs, you can use interpolators developed by others or yourself as the interpretation method for contours in information layers.

Using an interpolator as the interpretation method for contours in information type layers is recommended only when the layer contains point objects.

To select a PIE interpolator

1. Open the layers dialog.
2. Select the layer its interpretation method you want to change.
3. From the interpretation method popup menu at the bottom of the dialog select the required interpolator.

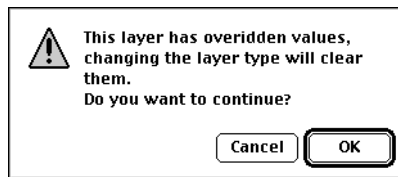
PIE interpolators developed by third-parties are supported by their respective developers.



PIE Interpolators appear below the Argus generic interpolator (3rd on the menu).

Changing Information Type Layers to Data Type Layers

You may change information type layers to data type layers and vice a versa. If the layer contain objects an alert will come up warning you that you are about to delete all objects in that layer.



New Maps Layer Capabilities

A number of new maps layer capabilities have been added. These include:

- Place Images
- Text annotation tool using system font
- Directly visualize data from mesh and grid layers
- Smooth contours in contour map post-processing objects
- New look of post-processing objects' dialogs
- Resize objects by click-drag

Images

Placing images allow you to better orientate in your domain, and digitize information from images. Images can be placed in Maps type layers. Since Images are opaque it is recommended that you place images that cover the same area on the screen in different Maps type layers. This will allow you better control of image visibility by using the show/hide layer command and by using the promote demote layer option.

Supported formats

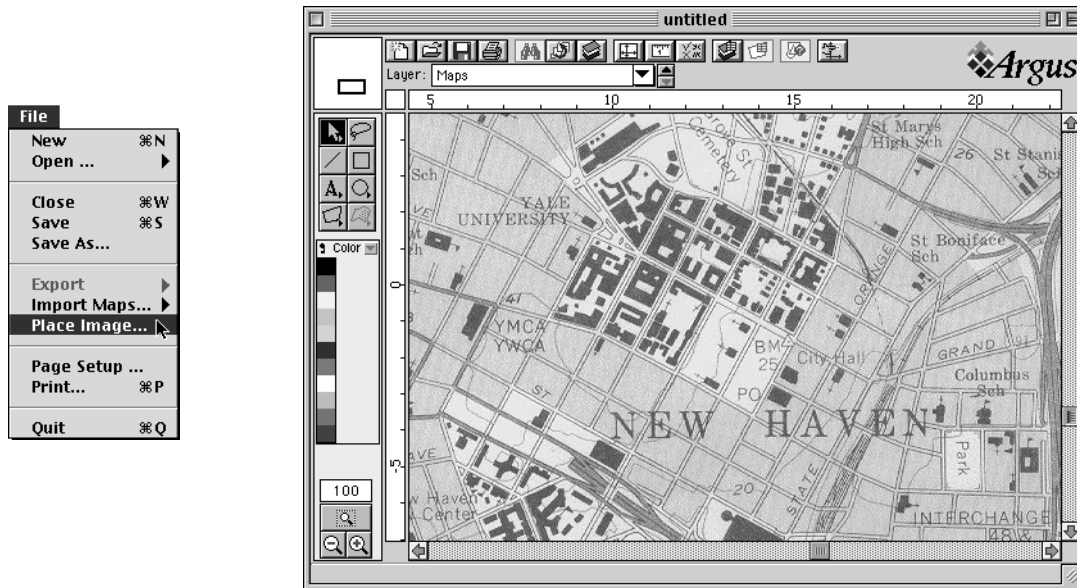
The following seven image formats are supported by Argus ONE.

Format name	Format file extension
GIF	gif
TIFF	tif or tiff
XBitmap	bm
MacPaint	mcp
Windows DIB	BMP
Sun Raster	RIS
Nexpert Objects Images	NBI

If you try to place a file which contains an image saved in a format other than those supported an alert will be presented.

To Place an image

1. From the File menu select Place Image...
2. From the dialog box select the image file you wish to place.
3. The image appears on the screen.



If the image is large it can take a while for it to be placed. Because of technical limitations a progress bar is not available to indicate the time left to complete the process.

Images and memory considerations

Contrary to vector graphics such as that of Argus ONE, DXF and Shape files, images consume large amounts of memory. Image size and the number of colors in the image are some of the factors effecting the amount of memory required by Argus ONE to present the image.

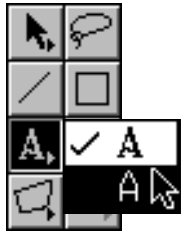
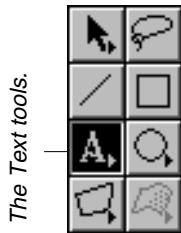
Editing Images

The following editing commands operate on images:

- Move
- Resize
- Scale
- Delete

The following editing commands are not applicable to images:

- Copy
- Rotate



Click and hold to select text type.

Text Annotation - System Font

The system-font text annotation tool, a variation of the vector font text annotation tool, uses a system font to present the text. Later versions of Argus ONE will allow you to select the system font from the available fonts installed on your system.

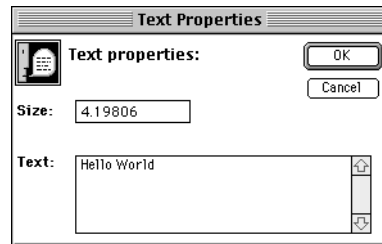
To choose system-font or vector-font annotation text

1. In the tool palette, click and hold the text tool to open the popup menu.
2. To select the system-font tool release the mouse button above the thick A icon.
3. To select the vector-font tool release the mouse button above the thin A icon.

Creating and editing system-font annotation text

System-font annotation text is created and edited in the same manner as vector-font text. However, system-font annotation text can not be rotated. As a result, all imported text (which might be rotated) is represented using the vector font.

System-font annotation text looks better but...



Hello World

Hello World

Hello World

...Can't be rotated as vector text can.

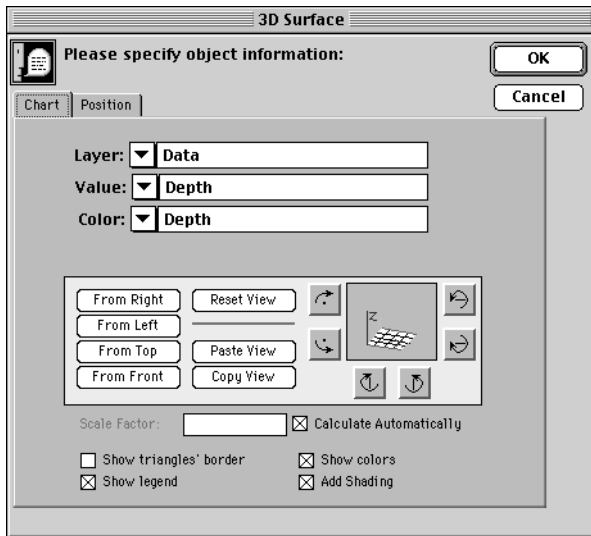
New Look of Post-Processing Objects' Dialogs

All post-processing objects' dialogs have been changed to tab dialogs. All dialogs have two panes. This change was introduced to allow for future panes offering more control of post-processing objects. The two panes available are the Chart pane and the Position pane.

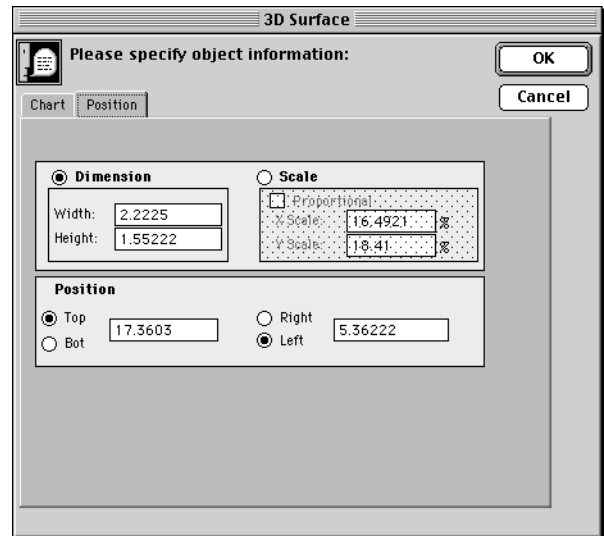
To move between the panes

- Click the tab its pane you want to open.

Click the Chart tab to bring its pane to the front



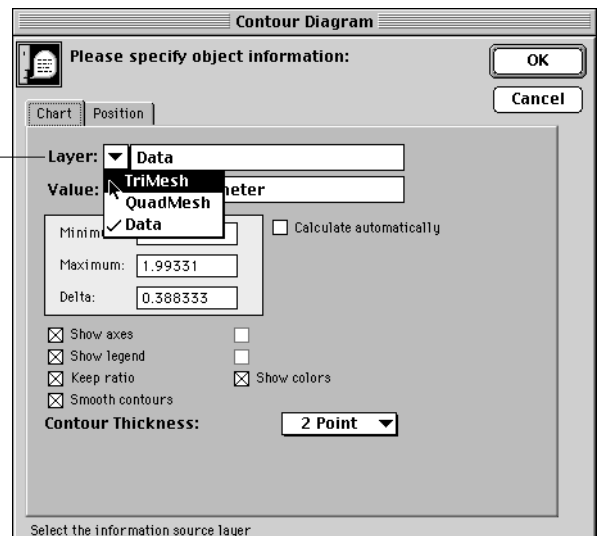
Click the Position tab to bring its pane to the front



Using Post-Processing Visualization Objects to View Mesh and Grid Data

Post processing objects can plot data directly read from mesh node parameters and grid and block parameters. Thus, to examine mesh and grid data you do not need to first read it into a data type layer. All grid, block, mesh layer and node parameters will be listed in the layer popup menu in the post-processing object dialog.

All mesh layers having node parameters and all grid layers having block parameters will be listed as well as all data type layers.

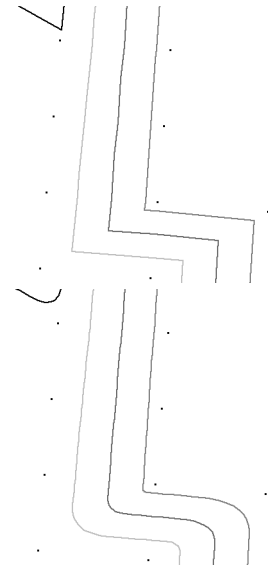
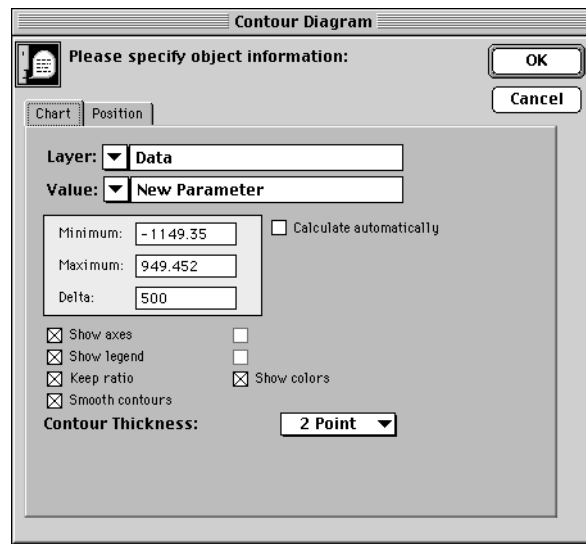


Smooth Contours in Contour Diagram Objects

Contours created in Contour diagram post-processing objects may be smoothed.

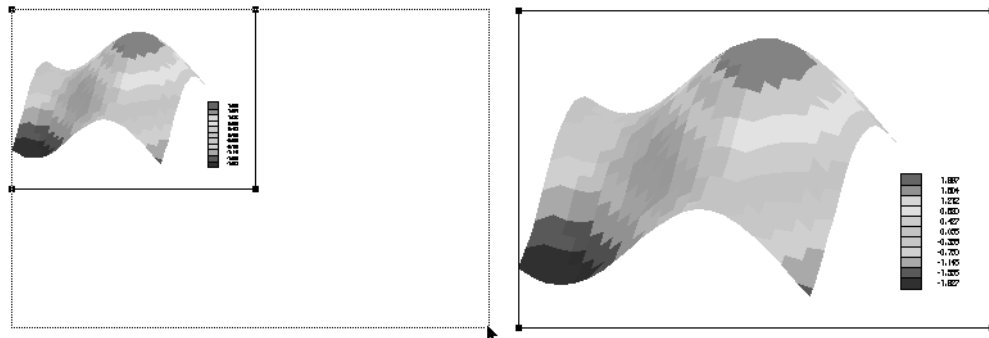
To smooth contours in contour maps

1. Open the Contour Diagram dialog box
2. Check the Smooth contours check box.
3. Click OK to close the dialog and accept the change.



Resizing Objects by Clicking and Dragging

All the objects available in Map type layers are resizable by click-drag.



To resize an object select it and click drag one of its bullets. Drag the stretch rectangle to the required size and...

...release the mouse button. The object is resized.

Values Floaters

As Data type layers, Information, Mesh and Grid type layers have the ability to show values assigned to objects in them in a table type window. These windows, which are floating windows, are referred to as Values Floaters. They also include charting capabilities which enable you to plot values using various methods. These enhanced Values floaters allow you to resize columns, select the range of information for charting and they also automatically update when you add or delete objects to your layer, or add or delete parameters in the layers they belong too.

In future versions, you will be able to select an object in your layer and see its line highlighted in the values-floater, and vice-a-versa.

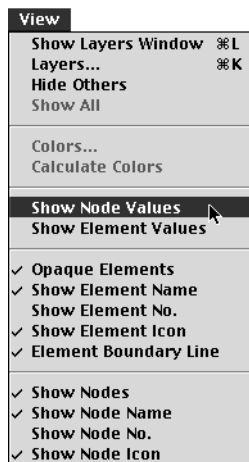
Working with Values Floaters

You can have as many open values-floaters as you need at any time. You open a values-floater from within the active layer. However, once it is opened, it will stay open even if you switch the active layer. This allows you to view information in layers that are not currently active.

Values floaters belong to the project in which they were opened. If you have a number values-floaters opened in one project, and you switch to another project they disappear from sight. Once you bring to the front the project they belong to they reappear.

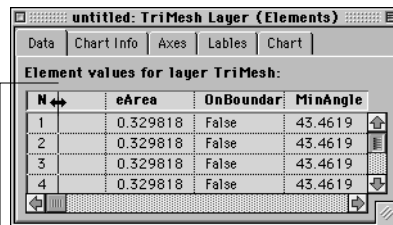
Opening and Working with Values Floater

Once you have objects or data in Data, Information, Mesh and Grid type layers, a menu command named Show <object type> Values will be available under the View menu.

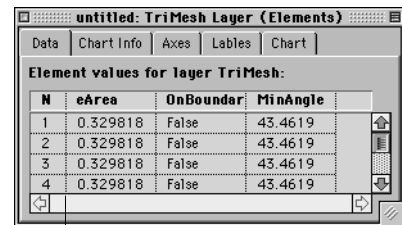


To resize the width of columns

1. Place the cursor above the dotted line between any two columns. The cursor changes into a two sided arrow.
2. Click-drag the cursor to the right or to the left to resize the column, a solid black line represents the new location of the column.
3. Release the mouse button.



Click-drag the cursor to resize the column width and...



...release the mouse button in the new location.

Values Floater - Information Type Layers

In information type layers the values-floater lists all contour objects and their values. If no parameters but the default one have been assigned to the layer, only the default's parameter values will be listed. The list updates as you add and delete objects or parameters. You may want to add parameters with contour functions such as ContourType() and ContourName() to display object specific information to assist you in identifying objects.

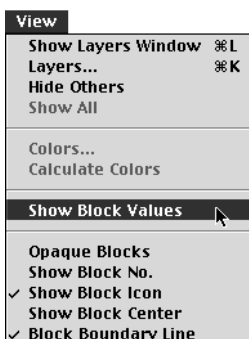


The dialog box titled 'untitled: Railroad_Name Layer' displays a table of contour values for the layer 'Railroad_Name'. The table has the following columns: Railroad_Name, CFCC, St_County, Record_Num, and ID.

Railroad_Name	CFCC	St_County	Record_Num	ID
Conrail Railroad	B11	11001	76226024	32.
Conrail Railroad	B11	11001	76226036	34.
Conrail Railroad	B11	11001	76226040	36.
Conrail Railroad	B11	11001	76226481	37.
Amtrak Railroad	B21	11001	76226648	38.
Amtrak Railroad	B11	11001	76226652	39.
Amtrak Railroad	B21	11001	76226655	42.
Csx Railroad	B21	11001	76226657	43.
Conrail Railroad	B11	11001	76226871	45.
Conrail Railroad	B11	11001	76226900	47.
Conrail Railroad	B11	11001	76226927	48.
Conrail Railroad	B11	11001	76226950	51.
Conrail Railroad	B11	11001	76227043	53.
Amtrak Railroad	B11	11001	76227116	56.

Values Floater - Grid Type Layers

In grid type layers the values-floater lists all blocks and their values. This might become useful if you need to view numerically all block values as they will be exported to your model. If no parameters have been assigned to the layer, only row and column numbers are listed. The list updates as you add and delete blocks or parameters. You may want to add parameters with block functions such as BlockArea() and BlockName() to display block specific information to assist you in identifying blocks.



The dialog box titled 'untitled: Grid Layer' displays a table of grid values for the layer 'Grid'. The table has the following columns: Row, Column, X, and SinX.

Row	Column	X	SinX
1	1	3.84969	-0.650388
1	2	4.77575	-0.997995
1	3	5.70177	-0.549207
1	4	6.62781	0.337846
1	5	7.55385	0.955299
1	6	8.4799	0.810428
1	7	9.40594	0.0188393
1	8	10.332	-0.787783
1	9	11.258	-0.965758
1	10	12.1841	-0.373063
1	11	13.1101	0.517335
1	12	14.0361	0.994902
2	1	3.84969	-0.650388

Values Floater - Mesh Type Layers

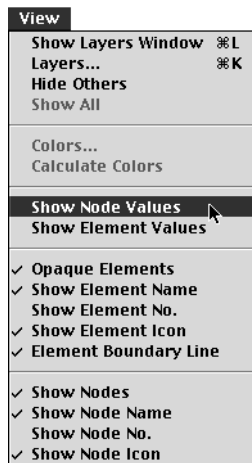
In mesh type layers two values-floaters are available, one for node values and the other for element values. This might become useful if you need to view numerically all node and element values as they will be exported to your model. You may want to add parameters with element or node functions such as ElementArea() and ElementName(), NodeName(), etc. to display element and node specific information to assist you in identifying elements or nodes.

The elements values-floater

The elements values-floater lists all elements and their values. If no element or layer parameters have been assigned to the elements, only the element numbers are listed. The list updates as you add and delete elements or parameters.

The nodes values-floater

The nodes values-floater lists all nodes and their values. If no node or layer parameters have been assigned to the nodes, only the node numbers and X and Y locations are listed. The list updates as you add and delete nodes or parameters.



untitled: TriMesh Layer (Elements)

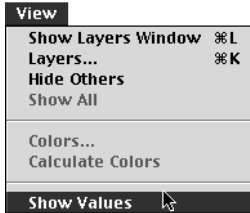
N	ElemArea	OnBoundary	MinAngle
1	0.454846	True	49.827
2	0.379491	True	50.5334
3	0.373753	True	53.3855
4	0.359836	True	55.4062
5	0.35946	True	53.3434
6	0.386168	True	56.0124
7	0.393949	True	55.3716
8	0.355884	True	53.8974
9	0.353845	True	53.1034
10	0.379402	True	55.6032
11	0.383019	True	56.5215
12	0.368068	True	51.5783
13	0.358784	True	53.0889
14	0.360585	False	52.8706
15	0.35677	False	54.6217

untitled: TriMesh Layer (Nodes)

N	X	Y	EffectiveArea	N_OnBoundary	EffectiveLen
2	10.4422	19.0536	0.151615	True	0.965943
3	9.5885	18.6479	0.151615	True	0.472607
4	5.13101	8.23118	0.729739	False	0.
5	5.08	7.20028	0.251081	True	0.927384
6	5.84435	7.76864	0.368781	True	0.952508
7	4.38326	7.77354	0.364349	True	0.90226
8	4.43162	8.76556	0.712875	False	0.
9	3.68653	8.34681	0.356396	True	0.90226

Values Floater - Data Type Layers

In data type layers the values-floater lists all values assigned to data points. If there are no data points in the layer, the Show Values menu is dimmed.



untitled: Data Layer

Data | Chart Info | Axes | Lables | Chart

Data values for layer Data:

X	Y	EffectiveArea	N_OnBoundary	EffectiveLeng
10.6589	18.091	0.151615	1.	0.493336
10.4422	19.0536	0.151615	1.	0.965943
9.5885	18.6479	0.151615	1.	0.472607
5.13101	8.23118	0.729739	0.	0.
5.08	7.20028	0.251081	1.	0.927384
5.84435	7.76864	0.368781	1.	0.952508
4.38326	7.77354	0.364349	1.	0.90226
4.43162	8.7656	0.712875	0.	0.
3.68653	8.34681	0.356396	1.	0.90226
2.7827	14.4909	0.772351	0.	0.
1.905	14.9967	0.260039	1.	0.932357
1.96968	14.0794	0.382863	1.	0.9195
2.75872	15.4024	0.390796	1.	0.945214

Values Floaters and Charts

Value-floaters also feature charts. Using charts you can graphically investigate the distribution of values assigned to objects. There are eight chart types available and these can be controlled using the labels and axes control panes. These charts are volatile, they are not saved once you close the values-floater they appear in.

Creating Charts

To create a chart you first need to select the cells to be charted.

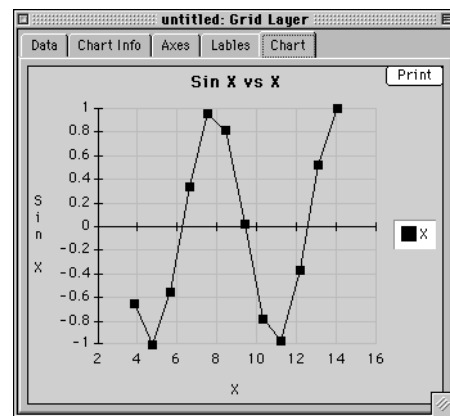
Once cells are selected you can click the Charts tab to view your chart. You may also click the Chart Info, Axes, and Labels tabs to adjust your chart.

untitled: Grid Layer

Data | Chart Info | Axes | Lables | Chart

Block values for layer Grid:

Row	Column	X	SinX
1	1	3.84969	-0.650388
1	2	4.77575	-0.997995
1	3	5.70177	-0.549207
1	4	6.62781	0.337846
1	5	7.55385	0.955299
1	6	8.4799	0.810428
1	7	9.40594	0.0188393
1	8	10.332	-0.787783
1	9	11.258	-0.965758
1	10	12.1841	-0.373063
1	11	13.1101	0.517335
1	12	14.0361	0.994902
2	1	3.84969	-0.650388



X and SinX are selected for row number 1 in the grid. Choosing XY Line chart type...

...and clicking the Chart tab produces a plot of the variation of SinX with X.

Selecting Cells

Selection is performed by clicking the mouse in the cells. Selected cells are highlighted. You may use several techniques for cell selection.

To select a single cell

- Click the cell. - The cell is highlighted.

To select a number of cells using click-drag

- Click-drag from the upper-left cell to the bottom-right cell. If you stretch the selection outside the floater size the window will scroll automatically to allow you to add cells outside the current view. All cells within the rectangle are selected.

To select a number of cells using click and shift-click

1. Click the first cell to be selected.
2. Shift-click the last cell to be selected.
The group of cells (across rows and columns) is selected.

To select discontinues cells or groups of cells

You may select discontinues groups of cells. This allows you to chart non-adjacent columns or rows.

1. Select the first group using one of the methods described above.
2. Hold the Control key and use any of the methods described above to add another non-adjacent group of cells to the selection.

Row	Column	X	Y	SinX
1	1	1.95674	10.2624	0.926443
1	2	2.90689	10.2624	0.232555
1	3	3.85704	10.2624	-0.655953
1	4	4.80719	10.2624	-0.99551
1	5	5.75733	10.2624	-0.50195
1	6	9.08285	10.2624	0.335302
1	7	13.8336	10.2624	0.954274
2	1	1.95674	11.1655	0.926443

Selecting discontinuous rows

Row	Column	X	Y	SinX
1	1	1.95674	10.2624	0.926443
1	2	2.90689	10.2624	0.232555
1	3	3.85704	10.2624	-0.655953
1	4	4.80719	10.2624	-0.99551
1	5	5.75733	10.2624	-0.50195
1	6	9.08285	10.2624	0.335302
1	7	13.8336	10.2624	0.954274
2	1	1.95674	11.1655	0.926443

Selecting discontinuous columns

To select whole columns

You may select whole columns of cells.

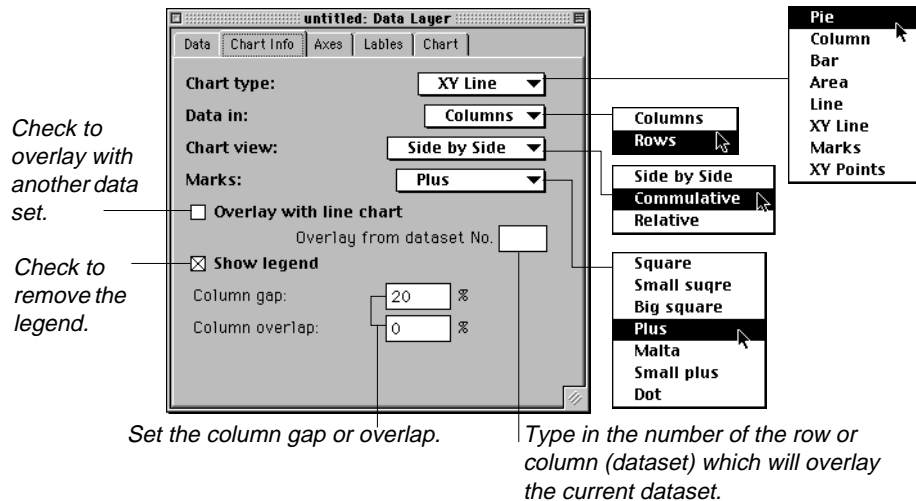
1. Click the column name cell.
2. To add more columns to the selection use the methods described above while clicking in the columns name cell.

Controlling Charts

In addition to controlling the values to be charted and the chart size, you can control the chart type, the chart axes and labels. These are performed in the Chart Info, Axes and Labels panes.

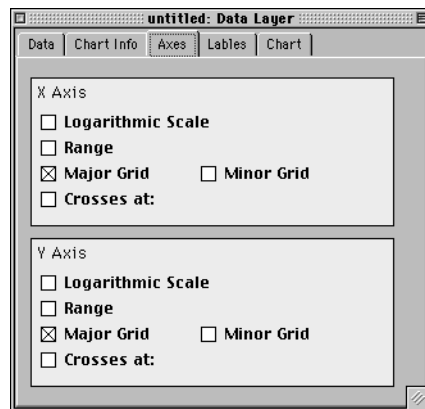
The Chart Info Pane

The chart info pane allows you to set the chart type, to select whether data is in rows or columns, to set the chart view, to add marks at data points, to overlay the chart with another chart and to add a legend.



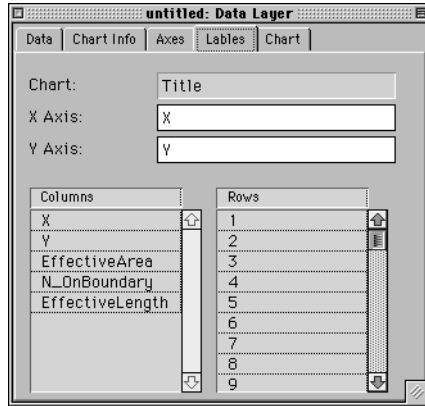
The Chart Axes Pane

The chart axes pane allows you to control the X and Y axes of the chart including the scale, range, background grid, and where the axis will cross.



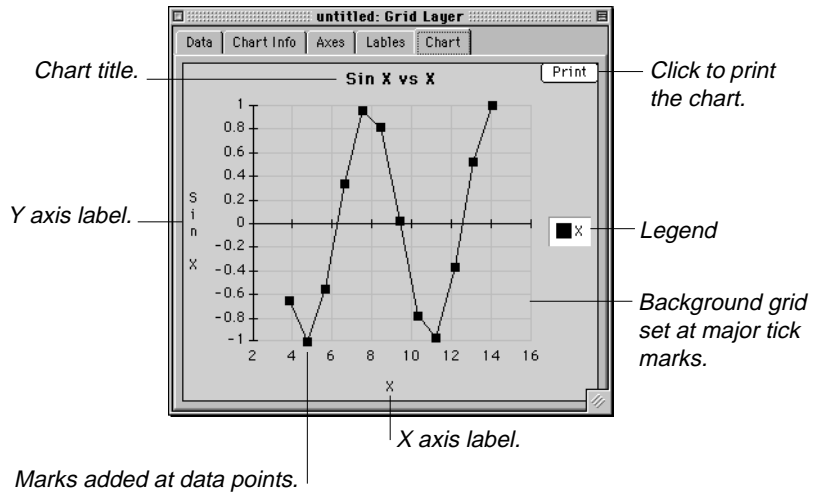
The Chart Labels Pane

The chart labels pane allows you to set the names for all labels, including the title, the axes and the legend. The default values presented are taken from the parameters names appearing in the data pane.



The Chart Pane

The chart pane is where your chart is presented. The chart automatically readjusts when you resize the window. You can print the chart by clicking the print button at the top-right corner of the window.



New Export Script Commands

Loop for Objects in Layer

Loop for objects in layer enables you to export objects stored in Information type layers from within a grid or a mesh export template. Future version which will feature export templates for information type layers, will support this loop type as well.

Using Information Type Layer Functions

You can use all available information type layer functions from within this loop. The following example exports all contours in a layer called Domain Outline and for each object exports its name, area, length, type and value of the parameter named Density. It also exports for every contour the X and Y coordinates of its vertices.

```

Define Variable: i [integer]
Redirect output to: contours.exp
# The following loop will loop for every
# Object (contour) in the Domain Outline layer
# and for each object export some contour information
# and the value of the density parameter.
Loop for: Objects in layer: Domain Outline
  Start a new line
    Export expression: ContourName(); [G0]
    Export expression: ContourArea(); [G0]
    Export expression: ContourLength(); [G0]
    Export expression: ContourType(); [G0]
    Export expression: Domain Outline.Density [G0]
  End line
# The following loop will export the X and Y coordinates
# of each vertex.
  Loop for: Variable i from: 1 to: NumVertices() step: 1
    Start a new line
      Export expression: NthVertexX(i); [G0]
      Export expression: NthVertexY(i) [G0]
    End line
  End loop
  Start a new line
  End line
End loop
End file

```


Nesting Loop for Objects in Layer within Loops for Blocks or Elements

You may nest a loop for Object in Layer within a loop over blocks and elements. This enables you to export all objects in a layer which intersect or are contained within a block or an element. Usually you will need to export information by block, however, if your code requires that you list all the objects in all layers per block or element you can nest a loop over objects in layer, within a loop over blocks or elements within a loop over layers.

The following example exports for each blocks the length of intersection of contours in each layer named rivers1, rivers2 and rivers3, as well as the area of intersection between each block and contours in each of the layers named lakes1, lakes2 and lakes3. The functions `ContourIntersectLength()` and `ContourIntersectArea()` which are unique to Loop for Object in Layer are described below. Also the use of square brackets as loop variable is also explained below.

```

Define variable: i [integer]
Define variable: j [integer]
Define variable: NumLayers [integer]
Set variable: NumLayers:= 3
#
Redirect output to: rivers_lakes.exp
# Loop from 1 to 3
  Loop for: Variable i from: 1 to: NumLayers() step: 1
#   Loop for Blocks
    Loop for: Blocks
#     Loop for all objects in layers named rivers1,
#     rivers2, and rivers3 and export the length of
#     rivers which intersects with the current block
      Loop for: Objects in layer: Rivers[i]
        Export expression: ContourIntersectLength() [G0]
      End loop
    End loop
  Loop for: Blocks
#   Loop for all objects in layers named lakes1,
#   lakes2, and lakes3 and export the area of
#   lakes which intersects with the current block
      Loop for: Objects in layer: Lakes[i]
        Export expression: ContourIntersectArea() [G0]
      End loop
    End loop
  End loop
End file

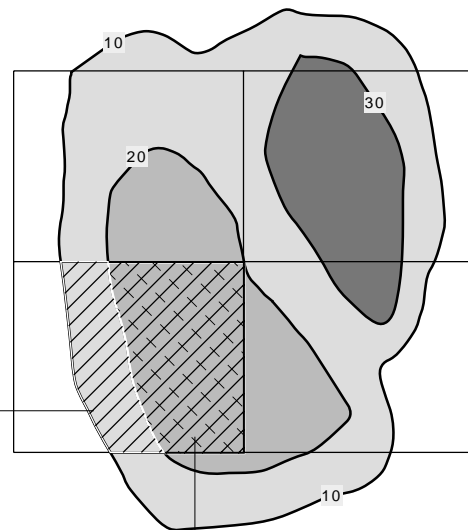
```

Functions unique to loop for objects in layer when nested within a loop for blocks or elements

Name & Syntax	Arguments	Description & Return Value
ContourIntersectArea()	<i>no arguments</i>	Returns the area defined by the intersection of a closed contour and a block or element. Operates on the information type layer the Loop for Objects in Layer is set to.

Example: In the following illustration the bottom-left block is tested for ContourIntersectArea() and returns for each of the two contours intersecting it, the area of intersection (marked by a different hatch pattern). The areas returned may be exported, added or manipulated using other functions.

Each block is tested in its turn (loop for blocks) with the 3 contours in the information layer (the layer the loop over objects in layer refers to), and returns the area of intersection. The bottom left block is intersected by the contour marked "10" and returns this hatched area.

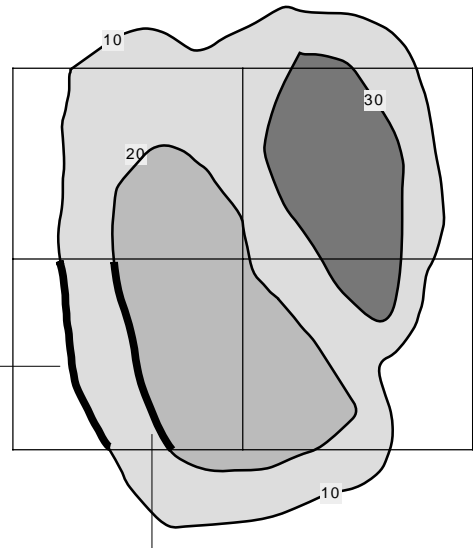


The bottom left block is also intersected by the contour marked "20" and thus returns the crossed area as well.

Name & Syntax	Arguments	Description & Return Value
ContourIntersectLength()	no arguments	Returns the length defined by the intersection of a closed or open contour and a block or element. Operates on the information type layer the Loop for Objects in Layer is set to.

Example: In the following illustration the bottom-left block is tested for ContourIntersectLength() and returns for each of the two contours intersecting it the length of intersection (marked by a thick line). The lengths returned may be exported, added or manipulated using other functions.

Each block is tested in its turn (loop for blocks) with the 3 contours in the information layer (the layer the loop over objects in layer refers to), and returns the length of intersection with each of the contours which intersect it. The bottom left block is intersected by the contour marked "10" and returns the length of this thick line.



The bottom left block is also intersected by the contour marked "20" and thus returns the length of intersection also marked by a thick line.

Looping Over Numbered Layers and Parameters

The square brackets [] notation allows one to automatically concatenate layers and parameter names with a loop variable. This allows you to repeat a sequence of an export template without prior knowledge of the numbers of layers to be exported.

In the following example NumLayers is either a global variable (see “Global Variables” on page 21s in this supplement) or a PIE variable which is set to the number of parameters sharing a name such as Grid.Perm Unit1, Grid.Perm Unit2, etc. As the loop progresses the loop variable i, the layer’s and parameter’s name, appearing before the square brackets, is concatenated with the value of i to produce the actual parameter name.

```
Loop for: Variable i from: 1 to: NumLayers step: 1
      Export matrix: Grid.Perm Unit[i] [G0]
End loop
```

Export Arrays

Using arrays in exported templates is supported through an Array PIE. This PIE allows one to initialize, set and get values of a one dimensional array. Any number of such arrays can be used. For details regarding the use of Array PIEs refer to “Array PIEs” on page 54.

New and Enhanced PIEs

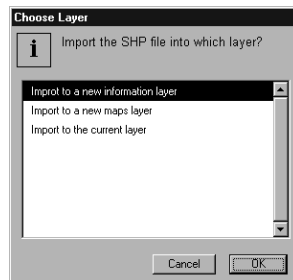
Although many PIEs are available, we describe here five PIEs which can be used by everyone. Other more specific PIEs are listed on the Argus web site.

Enhanced Import Shape Files PIE

The shape file PIE now automatically filters .shp files, notifies you of the types of fields stored in a shape file, and automatically creates new layers and parameters into which shapes will be imported with their respective values.

To import a shape file

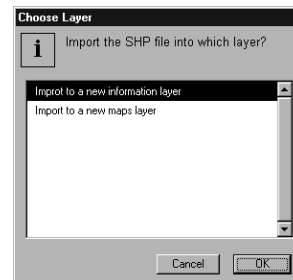
1. From the File menu select Import and Shape file.
2. Locate and open the file (.shp type)
3. Select the layer type into which the file will be read.
4. Select the fields to be imported. Fields' types are listed next to their names.



Choose whether to import into the current layer or into a new one. New layers are named after the name of the shape file.

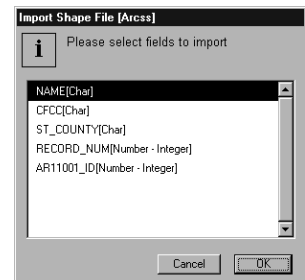
If you import into an existing information type layer, and allow intersection is not on, shapes which intersect other shapes will be skipped. An alert will pop up and allow you to either accept or cancel the import.

Also, if you import into an existing information layer, the existing parameters will be used, and their type will not be changed to complement the actual type being imported.



If the current layer is a domain layer, in which allow intersection is not allowed, the PIE will only permit import into a new map or information layer.

Although allow intersection will be turned on when importing into a new information type layer, shapes which intersect themselves (allowed in ArcView, but not in Argus ONE information layers) will be skipped. To avoid skipping such shapes, import into a Maps type layer.



This dialog lists all the fields related to objects in the shape file. Click to select those you wish to import.

If a new information type layer is selected for import, Argus ONE creates a new layer, names it after the name of the shape file, creates layer parameters using the fields' names and types, and imports the data.

If the objects imported reside outside the current drawing size, Argus ONE will alert you and offer to resize it so that all new shapes will fit in the resized drawing area.

Array PIEs

A PIE named ArrayPIE is available for use within export templates. It offers a one dimensional array which can be initialized, assigned values and return them on call. You can use as many arrays PIEs as you need within an export template, by creating duplicates of the original ArrayPIE. The name by which the PIE will be called from within the export template is the name you give it when you duplicate it. If you want to use more than one such PIE you must have a unique name for each of the Array PIEs.

Calling Functions for Array PIEs

There are four functions which are used to initialize the array, load its members, retrieve them, and dispose the array. The four functions are:

1. `ArrayInit(n)` - Initializes a one dimensional array of length `n`.
2. `ArraySet(i,v)` - Sets the member `i` to value `v`.
3. `ArrayGet(i)` - Retrieves the value stored in the `i` member and returns it.
4. `ArrayDispose()` - Disposes the array and the memory allocated to it.

If you wish to use two arrays, you may call the first `Array1` (on PC `Array1.dll`) and the second `Array2` (on PC `Array2.dll`). To use the above calls you should use the syntax (suffix the array name with the function): `Array1Init()`, `Array1Set()`, `Array1Get()`, `Array1Dispose()` for the first and `Array2Init()`, `Array2Set()`, `Array2Get()`, `Array2Dispose()` for the second.

The following example initializes a `MyArray1` of length 10. While doing so it tests if the Array PIE exists and if the initialization terminated successfully. It then sets values to each of the array members and exports these values.

```
Set Variable: NumLayers:=10
Set Variable: ok:=1
Set Variable: Test:= MyArray1Init(NumLayers)
If: IsNAN(Test)
  Alert: "The PIE MyArray1.DLL could not be found!"
  Set Variable: ok:= 0
End if
If ok
  Loop for: Variable i from: 1 to: NumLayers step: 1
    Evaluate expression: MyArray1Set(i,i*2)
  End Loop
  Loop for: Variable i from: 1 to: NumLayers step: 1
    Start a new line
    Export expression: MyArray1Get(i) [G0]
    End line
  End Loop
  Evaluate expression: MyArray1Dispose()
End if
```

The members of an N size array start at 0 and end at N.-1

Interpolator PIEs from SRI

Subterranean Research, Inc. has kindly allowed Argus to distribute its interpolators with the Argus ONE GIS module. If you find these PIEs useful or have any suggestions on how to enhance or improve them, please contact SRI (see address below).

Although the following documentation by Subterranean Research, Inc. (SRI) refers to use of their PIEs with data stored in data type layers, these interpolators are also available in information type layers. If an information layer contains close or open contours, those are sent to the interpolator PIEs as a list of their vertices assigned the contours' value.

Using SRI PIEs

If you experience problems using these PIEs please contact SRI as well as Argus. Argus does not warrant the use of these PIEs in any way.

Original Notes from SRI

NOTES FOR SRI_PIE2 Intended for Wintel Windows 95 or NT machines
Original Release date: March 14, 1997 Revision Date:N.A.

Copyright 1997. All Rights Reserved. Subterranean Research, Inc.
Burlington, VT 05401 USA (802)-658-8878 infodesk@subterra.com

What does it do

This PIE provides 3 different ways to interpolate data in Argus ONE data layers. The methods and the names by which they are referenced in the Interpolation Method popup menu are:

- InvDistSq - Inverse distance squared interpolator.
- NN2D - Nearest Neighbor interpolator.
- SCANN2D/v1 - Version 1 of 2-D Neural Network classifier/interpolator. (See technical notes below.)

How to choose an interpolator from SRI_PIE2

In the Layers Dialog select the data layer whose data you want to interpolate using SRI_PIE2. (See Argus ONE User's Guide p. 56s in Supplement for version 3.)

- From the Interpolation Method popup menu, select an interpolation method (InvDistSq, NN2D, SCANN2D/v1).

A simple example for testing the use of the PIE

In the following example a synthetic data set is created at mesh nodes, read into a data type layer, and then interpolated using NN2D. Simply substitute the reference name for any other interpolator desired.

Creating a synthetic data set

1. In a mesh type layer create a mesh. (See Argus ONE Manual for help.)
2. Open the Layers Dialog.
3. Create in the mesh layer a new node parameter named OBSERVATIONS and for values assign it the expression X(). (This will assign the nodes the nodal X coordinate values at each node.)
4. Close the Layers Dialog.

Reading the synthetic data set into a data layer

5. In the main window, select the DATA layer.
6. From the FILE menu select “Read Data from Layer”. A dialog box opens.
7. From the list, select OBSERVATIONS under the TriMesh layer. The data are read from all nodes to the data layer. The small dots indicate the position of data points.

Setting the interpolation method to NN2D

8. Open the Layers Dialog.
9. In the layers list select the appropriate parameter of the Data layer.
10. From the Interpolation Method popup menu (at the bottom of the dialog), select NN2D.
11. Close the Layers dialog.

Viewing the interpolated values at cursor position

12. As mouse is moved over the screen you will see, in the Information ruler, the X coordinate of the node nearest to the current location of the cursor.

Some Important Technical Notes for Users of SCANN2D/v1

- 1.0 SCANN2D/v1 uses an artificial neural network that (1) matches input data exactly (like any other interpolator) and (2) estimates values at spatial locations between input data points. See the paper by Rizzo and Dougherty, “Characterization of aquifer properties using artificial neural, Neural Kriging”, Water Resources Research, 1994, for a thorough description.
- 2.0 SCANN2D/v1 is a classifier. That is, it assumes that the quantity being interpolated is described by its “class”, which is stored numerically as an integer. In SCANN2D/v1, class numbers 1 through 24 are allowed. As an example, say the hydraulic conductivity tests at a site are partitioned among 8 bins (with the class numbers 1 through 8 corresponding to ranges of hydraulic conductivity values for gravel, coarse sand, medium sand, fine sand, silty sand, silt, silty clay, and clay). Then SCANN2D/v1 will predict the class number for hydraulic conductivity at any specified point. To use this information in groundwater simulation, the user must then convert the hydraulic conductivity class number into an effective

hydraulic conductivity value (say using Argus ONE's Expression facility in the Layers dialog).

- 3.0 SCANN2D/v1 uses "normalized coordinates", as described in the Rizzo and Dougherty paper cited above. Therefore, there are limitations on the amount of extrapolating that can be performed. The version released in SRI_PIE2 is limited as follows:
 - 3.1 SCANN2D/v1 allows interpolation/extrapolation to points that lie within an "estimation rectangle". This estimation rectangle is defined by bottom-left and top-right coordinates of (xmin - xrange, ymin - yrange) and (xmax + xrange, ymax + yrange), respectively. The terms xmin, ymin, xmax, and ymax are the minimum and maximum x and y coordinates in the observational data that are being interpolated. The terms xrange and yrange are (xmax - xmin) and (ymax - ymin), respectively.
 - 3.2 In SCANN2D/v1, any point outside of the allowed estimation rectangle is assigned the value 9999, which makes it easily detected during use.

Please contact Subterranean Research if your application requires an alternative estimation rectangle.

For Mac or Unix users

The SRI_PIE2.DLL is not currently available for Macintosh or Unix computers. If this is a critical need for your organization, we would be happy to discuss it with you.

Keeping Current with SRI's PIEs and Helper Applications for Argus ONE

This version of SRI's Plug-In Extensions for Argus ONE, SRI_PIE2.DLL, is available free for Argus Argus ONE users.

Subterranean Research, Inc. plans to update its PIE for Argus Argus ONE at least twice per year. Future versions, including updates, patches for bug corrections, and new features, will be available for an initial purchase and annual subscription fee. Pricing has not been set (as of March 14, 1997).

In addition, Subterranean Research, Inc. may be releasing one or more "Helper Applications" that can be used with Argus ONE, yet are not Plug-In Extensions. These will be available separately. Price and delivery are not yet available.

Contacting Subterranean Research, Inc.

Please contact David Dougherty at Subterranean Research, Inc. with any questions or comments. Telephone (802)-658-8878, Fax (802)-658-8878, Email David.Dougherty@subterra.com.

