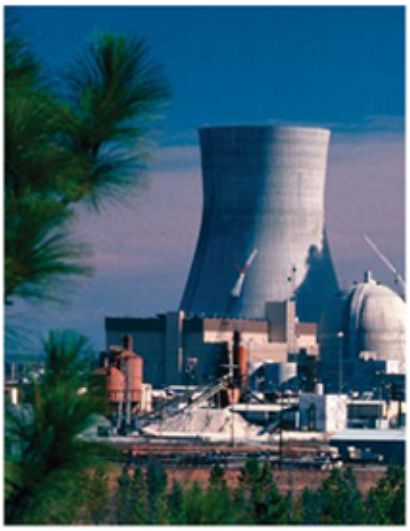
Hangers and Supports
Reference Data Guide

Process, Power & Marine


INTERGRAPH

Version 2007 January 2007 DSP3D-PE-200014G

Copyright

Restricted Rights Legend

Intergraph Corporation
Huntsville, Alabama 35894-0001

Warranties and Liabilities

Trademarks

Table of Contents

Preface

This document is a reference data guide for the SmartPlant® 3D Hangers and Supports task. The purpose of this document is to describe the reference data delivered with the software for this task.

Reference data includes both catalog data and specification data. Catalog data includes the parts that you place in the model, such as piping components and equipment. Specification data includes the rules that govern how those parts are placed and connected.

SmartPlant 3D Documentation Set

The SmartPlant® 3D documentation set is available as Adobe® PDF files. The content of the PDF files is the same content as online Help. To access these PDF documents in the software, click **Help > Printable Guides**.

The documentation set is divided into four categories:

- Administrative guides contain information about installing, configuring, customizing, and troubleshooting SmartPlant 3D.
- User's guides provide command reference and how-to information for working in each SmartPlant 3D task.
- Reference data guides define the reference data workbooks. Not all tasks have reference data.
- ISOGEN guides

Administrative Guides

Project Management User's Guide - Provides instructions for setting up the databases, creating permission groups, backing up and restoring project data, assigning access permissions to the model, managing interference detection, defining and managing locations for Global Workshare, controlling duplication and consolidation of plants, tools for synchronization, regeneration of report databases, and version upgrade.

SmartPlant 3D Database Integrity Guide - Provides information about the error messages in the database integrity reports, including meaning, cause, and possible corrective action.

SmartPlant 3D Global Workshare Guide - Provides instructions for setting up the software and the databases to work in a workshare environment.

SmartPlant 3D Installation Guide - Provides instructions on installing and configuring the software on both the client and server computers.

SmartPlant 3D/IntelliShip Programmer's Guide - Provides information about custom commands, naming rules, and symbol programming.

SmartPlant 3D Integration Reference Guide - Provides information about installing, configuring, and using SmartPlant 3D in an integrated environment.

SmartPlant 3D Interference Checking Guide - Provides information on installing, configuring, and using the interference detection service.

SmartPlant 3D Interpreting Human Piping Specifications - Provides information about how to interpret human piping specifications so that you can create the corresponding piping specification in the software.

SmartPlant 3D Plant Design System (PDS) Guide - Provides all information needed to use PDS with SmartPlant 3D. Topics include referencing active PDS projects in SmartPlant 3D, exporting PDS data and importing that data into SmartPlant 3D, converting PDS reference data to SmartPlant 3D reference data, and converting EDEN symbols to Visual Basic symbols.

SmartPlant 3D Release Bulletin - Provides what's new, hardware/software requirements, and support information for the current release.

SmartPlant 3D Troubleshooting Guide - Provides information on how to resolve errors that you may encounter in the software by documenting troubleshooting tips, error messages, and to do list messages.

User's Guides

Catalog User's Guide - Provides information about viewing, editing, and creating reference data and select lists (codelists).

Common User's Guide - Provides information about defining workspaces, navigating in the model, precision input, filtering, manipulating views, and running reports.

Drawings and Reports User's Guide - Provides information about creating drawing and report deliverables.

Electrical User's Guide - Provides information about routing electrical cable, cableway, cable tray, and conduit.

Equipment and Furnishings User's Guide - Provides information about placing equipment.

Grids User's Guide - Provides instructions for creating coordinate systems, elevation grid planes, vertical grid planes, radial cylinders, radial planes, grid arcs, and grid lines.

Hangers and Supports User's Guide - Provides instructions on placing piping, duct, cableway, and conduit supports in the model.

HVAC User's Guide - Provides instructions for routing HVAC duct.

Piping User's Guide - Provides instructions for routing pipe and placing valves, taps, and pipe joints.

Space Management User's Guide - Provides instructions for placing volumes (such as drawing volumes, obstruction zones) in the model.

Structural Analysis User's Guide - Provides instructions for defining loads, load cases, load combinations, and the importing and exporting of analytical data.

Structure User's Guide - Provides instructions for placing structural members such as: beams, columns, braces, slabs, openings, stairs, ladders, equipment foundations, and handrails.

Systems and Specifications User's Guide - Provides instructions for creating systems and their hierarchies and selecting which specifications are available for each system type.

SmartPlant 2D Symbols User's Guide - Provides instructions for creating cross section symbols.

Reference Data Guides

Drawings and Reports Reference Data Guide - Provides information about reports reference data.

Electrical Reference Data Guide - Provides information about electrical cable, cableway, cable tray, and conduit reference data.

Equipment and Furnishings Reference Data Guide - Provides information about equipment reference data and name rules.

Hangers and Supports Reference Data Guide - Provides information about hangers and supports reference data.

HVAC Reference Data Guide - Provides information about HVAC reference data.

Piping Reference Data Guide - Provides information about piping reference data including piping specifications, piping specification rules, piping parts, piping symbols, and name rules.

SmartPlant 2D Symbols Reference Data Guide - Provides information about the two-dimensional symbols used in all tasks.

SmartPlant 3D Reference Data Guide - Provides instructions about the Bulkload utility, codelists, and the reference data common to several disciplines.

SmartPlant 3D Symbols Reference Data Guide - Provides information about the Visual Basic Part Definition Wizard and the three-dimensional symbols used in all tasks.

Space Management Reference Data Guide - Provides information about space management reference data.

Structure Reference Data Guide - Provides information about structural reference data and name rules.

ISOGEN Guides

AText Reference Guide - Provides information about alternative text for isometric drawings. This guide is from Alias, the makers of ISOGEN®.

Option Switches Reference Guide - Provides information about the ISOGEN option switches for isometric drawings. This guide is from Alias, the makers of ISOGEN.

Symbol Keys Reference Guide - Provides information about the symbol keys for isometric drawings. This guide is from Alias, the makers of ISOGEN.

Documentation Comments

Send documentation comments or suggestions to PPMdoc@intergraph.com.

What's New in Hangers and Supports Reference Data

The following changes have been made to the Hangers and Supports reference data.

Version 2007

- Added *Supports and Local Coordinate System (LCS): An Overview*. This topic explains the LCS associated with each support and assembly. Using the LCS facilitates creating drawings with consistent views.
- The **OffsetY** column has been added to **HS_System.xls**. This provides a way to define a rule or value for Y-direction (perpendicular to face) object placement. See *Face Position Selection Sheet* for more information.
- Additional Cooper B-Line cable tray support parts are now available. Also, there are additional cable tray supports available based on the Cooper B-Line tray supports.
- Additional Halfen Powerclick support parts and assemblies are now available. The parts and assemblies can be found in *<product directory>\3D\CatalogData\BulkLoad\AdditionalDataFiles.*
- You can define loading property values for hangers using the **Support Properties** dialog box. Values can be defined for loads and pipe restraint stiffness.
- Added the Pipe Supports Limited (PSL) pipe support parts catalog. The catalog can be found in *<product directory>\3D\CatalogData\BulkLoad\AdditionalDataFiles\HS_PSL.xls*. The codelist file is available in the same directory and is named *HS_PSL_Codelist.xls*.
- A metric utility support part library is now available. The file can be found in *<product directory>\3D\CatalogData\BulkLoad\AdditionalDataFiles\HS_Util_Metric.xls.* The codelist file is available in the same directory and is named *HS_Util_Metric_Codelist.xls*.
- The AISC steel library is now updated to the AISC v3.1 standard.

Hangers and Supports Reference Data: An Overview

Before working with hangers and supports reference data, you must be familiar with how SmartPlant 3D handles reference data in general. If you have not already done so, read and understand the following important concepts and procedures described in the *SmartPlant 3D Reference Data Guide*:

- Custom attributes
- Symbol creation
- Codelists (also referred to as "select lists")
- Naming rules
- Bulkloading

For example, before you can define support parts, you must understand how part data relates to data on the custom interfaces sheet, and how that information relates to parameters defined when the part symbol is created.

Hangers and Supports Workbooks

The hangers and supports reference data is defined in several Microsoft Excel workbooks. The software delivers these workbooks to the *[Product Directory]*\CatalogData\Bulkload\DataFiles folder. Additional workbooks are delivered to the *[Product Directory]*\CatalogData\Bulkload\AdditionalDataFiles folder and the the [*Product Directory]*\CatalogData\Bulkload\AdditionalDataFiles\Delta61to2007\ExampleWorkbooksPerFeature folder. The following tables list tthe delivered workbooks and gives a brief description of their content.

Tip

- The filenames all start with "HS" for easy recognition.

The following table contains workbooks available in the *[Product Directory]*\CatalogData\Bulkload\DataFiles folder.

Workbook	Content
HS_Anvil.xls	Defines part classes from the Anvil catalog.
HS_Anvil_Codelist.xls	Defines codelists for the Anvil catalog.
HS_Assembly.xls	Defines assembly part classes.
HS_Assembly_Codelist.xls	Defines codelists for assemblies.
HS_GTypeComponent.xls	Defines the imported GType component symbol.
HS_HgrAisc-LRFD-3.1.xls	Defines connection support component classes.

Workbook	Content
HS_LRParts.xls	Defines load-rated parts for rigid rod assemblies.
HS_System.xls	Contains information from the previous **HangersandSupports.xls** workbook, such as designed supports, hangers and supports rules, and hanger discipline filters.
HS_Utility.xls	Defines utility classes, which include concrete parts and steel plates.
HS_Utility_Codelist.xls	Defines codelists for the utility classes.

The following table contains workbooks available in the [*Product Directory]*\CatalogData\Bulkload\AdditionalDataFiles folder.

Workbook	Content
HS_Bline_Assy.xls	Defines the Copper B-Line Cable Tray Assemblies catalog.
HS_Bline_Assy_Codelist.xls	Defines the Copper B-Line Cable Tray Assemblies codelist.
HS_Bline_Tray.xls	Defines the Copper B-Line Cable Tray Parts catalog. See also [*Product Directory]\CatalogData\Bulkload\AdditionalDataFiles\Delta61 to2007\ExampleWorkbooksPerFeatureHS_System Add Cable Tray Parts Compatiblity.xls for additional information.*
HS_Bline_Tray_Codelist.xls	Defines the Copper B-Line Cable Tray Parts codelist.
HS_Halfen_Assy.xls	Defines the Halfen Assemblies catalog.
HS_Halfen_Assy_Codelist.xls	Defines the Halfen Assemblies codelist.
HS_Halfen_PC.xls	Defines the Halfen Parts catalog.
HS_Halfen_PC_Codelist.xls	Defines the Halfen Parts codelist.
HS_Hgr_BlineChannel.xls	Defines the Hanger Bline Channels for Bline Assemblies.
HS_Hgr_BlineChannel_Codelist.xls	Defines the Hanger Bline Channels for Bline codelist.
HS_HgrAisc.xls	Defines the Aisc hanger beams.

Workbook	Content
HS_HgrCisc.xls	Defines hanger beams in metric. If you intend to use HS_HgrCisc.xls, you must first bulk load StructCrossSections-CICS-7.2.xls. The StructCrossSections-CICS-7.2.xls workbook can be found in the *[Product Directory]\CatalogData\Bulkload\AdditionalDataFiles folder*.
HS_HgrEuro_OTUA.xls	Defines the Hanger Euro OTUA catalog.
HS_Lisega.xls	Defines part classes from the Lisega catalog.
HS_Lisega_Codelist.xls	Defines codelists for the Lisega catalog.
HS_PSL.xls	Defines the PSL catalog.
HS_PSL_Codelist.xls	Defines the PSL codelist.
HS_Util_Metric.xls	Defines the Utility Metric parts.
HS_Util_Metric_Codelist.xls	Defines the Utility Metric codelist.

The following table contains workbooks available in the [*Product Directory]*\CatalogData\Bulkload\AdditionalDataFiles\Delta61to2007\ExampleWorkbooksPerFeature folder.

Workbook	Content
HS_Assembly Strict Face Selection.xls	Defines the use of Strict Face Selection for assemblies.
HS_Assembly to specify OffsetY for Assy_FR_LS_LS.xls	Provides information about defining Y offset for parts.
HS_Assembly_restraints.xls	Provides instructions for adding stiffness restraints to sections.
HS_Face Position Offset in Y Direction.xls	Defines information about defining Y offset for the face position.
HS_Gtype Component Translator.xls	Explains how to translate supports from other software (for example, PDS) for use with SP3D.
HS_Restraint Properties for Standard Support Assemblies.xls	Defines restraint properties for support assemblies.
HS_System Add Cable Tray Parts Compatiblity.xls	Provides instructions for use with the HS_Bline_Tray.xls workbook when using cable tray parts.

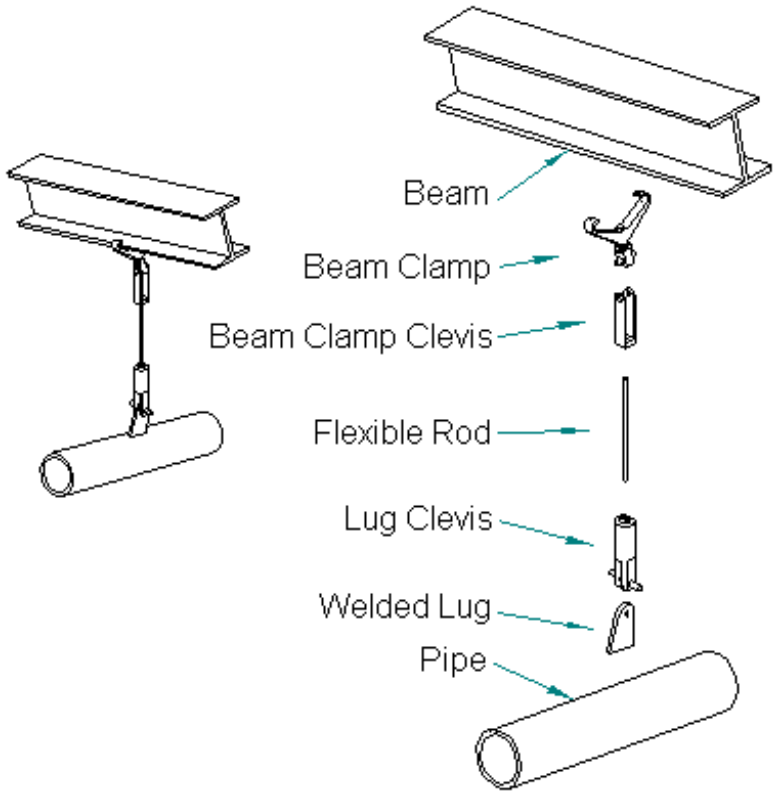
Workbook	Content
HS_System AddRoundingRule and Remove ref to ISPSMember and FacePosSelection.xls	Defines instructions for implementing rounding for supports.
HS_System StrictFaceSelection.xls	Defines strict face selection for assemblies.
HS_System to specify OffsetY for Assy_FR_LS_LS.xls	Defines instructions for using Y offset for assemblies.

Related Topics

- *Understanding Assemblies and Parts*, page 15

Understanding Assemblies and Parts

The hangers and supports that you place in your model are based on individual *parts*, such as a beam clamp or a welded lug. A single support may be comprised of multiple parts. This grouping of parts is called a *support assembly*. The following figure shows a typical support with an assembly that contains five unique parts.



Hanger Assembly with Constituent Parts

The assembly of a support's constituent parts is done using information defined in the reference data. Each assembly defined in the reference data is assigned an *assembly information rule*. The software uses the logic contained in the assembly information rule to put together the individual support parts to form an assembly.

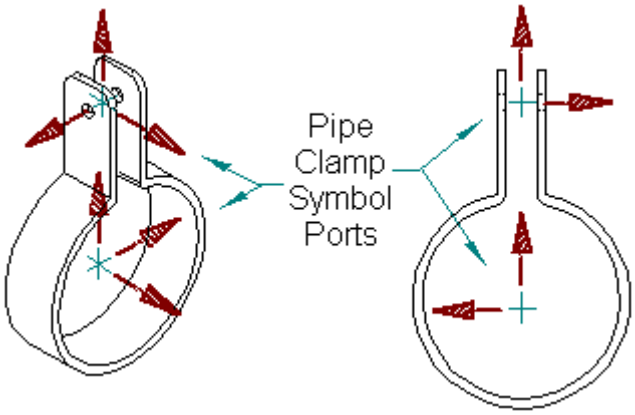
The software properly positions the parts in relation to each other, the supported object (such as a pipe), and the supporting object (such as a beam) by applying *constraints* between *ports*. The assembly information rule is responsible for identifying all constraints necessary to position the parts correctly. Specifying individual constraints provides the most flexibility and control over the assembly process. However, it is a tedious and repetitive process. To simplify things, relationships between parts are described using *joints* instead of individual constraints. Each joint represents a series of constraints that simulate the mechanical connection between parts. The assembly information rule provides a list of joints between assembly parts. The software parses this list and applies the appropriate constraints when creating the assembly.

Related Topics

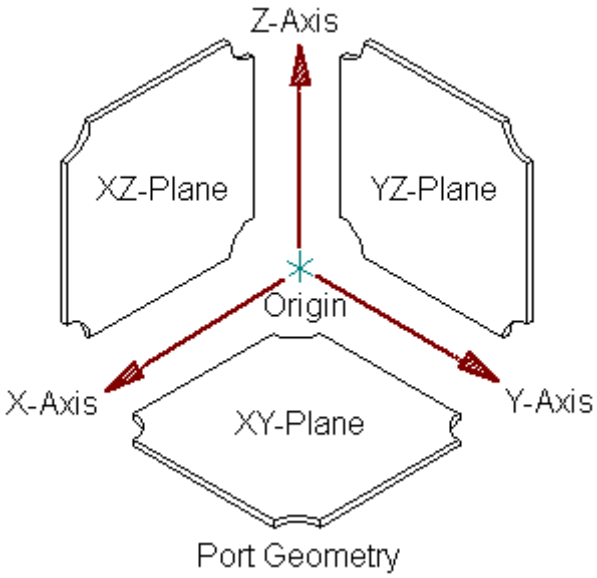
- *Understanding Constraints*, page 21
- *Understanding Joints*, page 22
- *Understanding Part Ports*, page 17
- *Understanding Reference Ports*, page 20

Understanding Part Ports

When creating part symbols that you intend to use in support assemblies, you should specify outputs called *ports* to aid in the placement of the part symbol. The following figure shows the output ports for a pipe clamp part symbol.

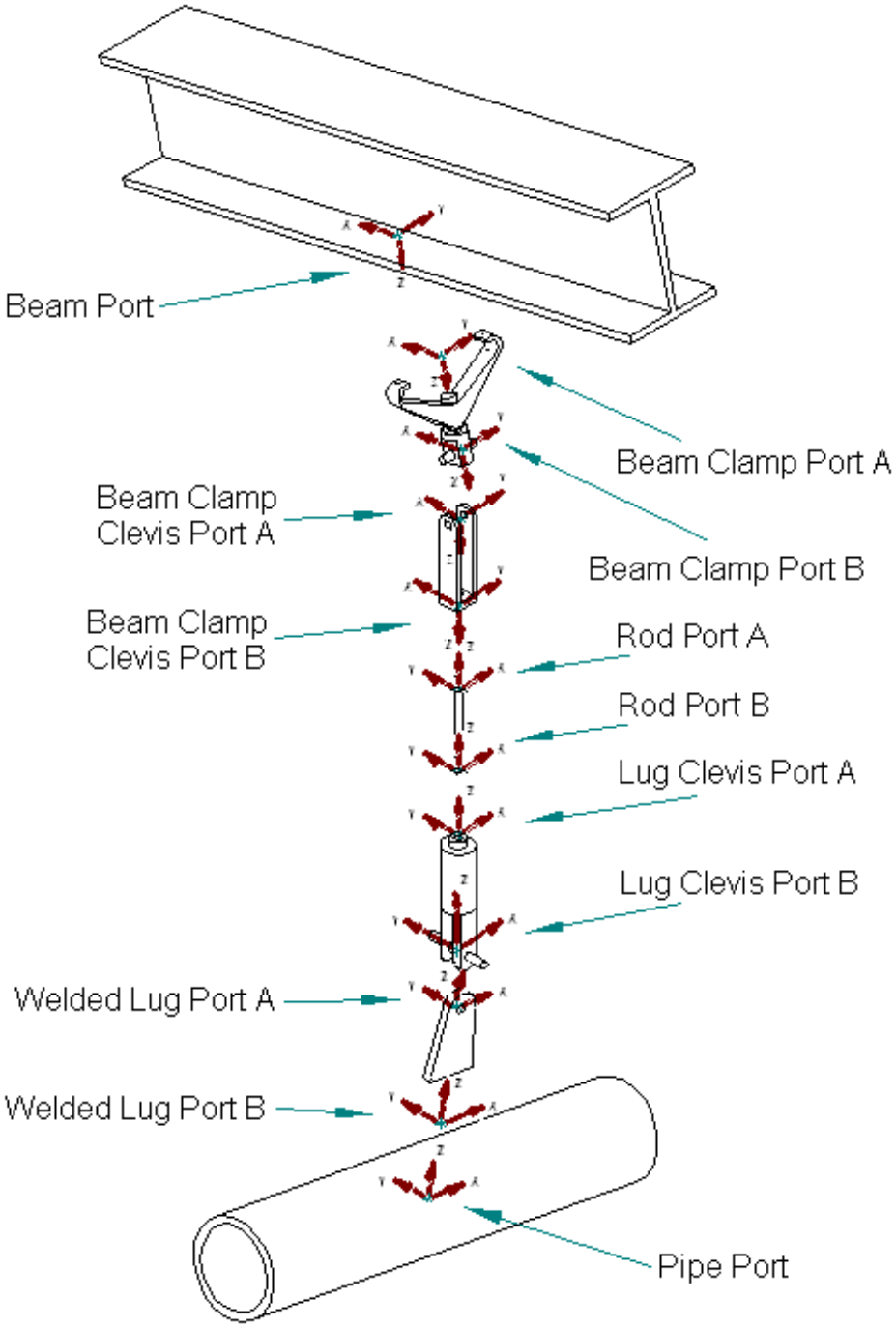


The geometric information associated with each port is, in essence, a coordinate system. The port defines an origin, three axes, and three planes.



Ports convey information through their relative location with respect to the symbol graphics. In the pipe clamp part symbol, one port is defined coincident with the centerline of the clamp's associated pipe. The second port is positioned where the clamp is connected to the other assembly part. Understand, however, that ports can be defined at any position in space and therefore do not need to be attached to the part symbol's graphical representation. In addition, the location of a port can be a function of the input of the part symbol. For example, you can define a port position for a welded lug based upon the pipe radius.

The following figure shows the part symbols and their ports arranged in an exploded view of the assembly. Each part has two ports defined; however, any number can be defined as required. The orientation and location of each part symbol port is independent of the other ports. During placement, however, a symbol's collection of ports are treated as rigid and will always retain their location with respect to the other ports and the geometry of the part symbol.



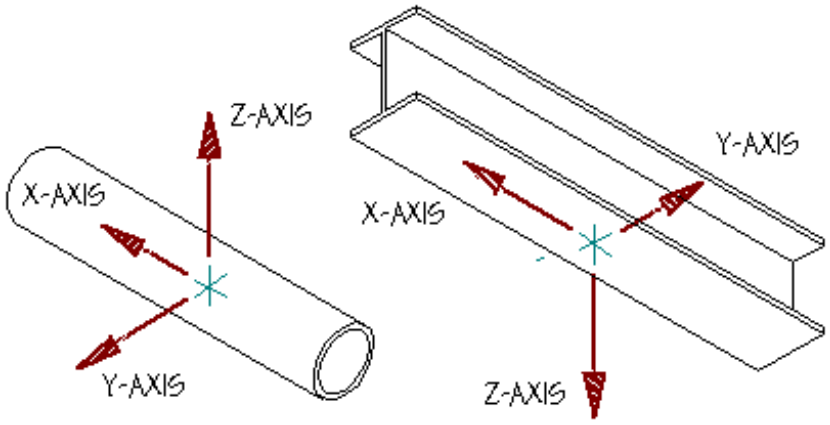
Related Topics

- *Understanding Assemblies and Parts*, page 15
- *Understanding Constraints*, page 21
- *Understanding Joints*, page 22
- *Understanding Reference Ports*, page 20

Understanding Reference Ports

While part ports define the connection relationship between parts in an assembly, *reference ports* define how a support assembly attaches to the supporting object (such as a beam) and the supported object (such as a pipe or cable tray). Unlike part symbol ports, reference ports are provided by the software and cannot be moved.

The example support that we have been discussing needs two reference ports, one on the beam and one on the pipe. The Z-axis of the supported object (in this case, the pipe) always points toward the supporting object (the beam). Conversely, the Z-axis of the supporting object (the beam) port always points toward the supported object (the pipe).



When you are defining assembly information rules in Visual Basic, the reference port on the pipe is called -1, ROUTE. The reference port on the beam is called -1, STRUCTURE. If you are connecting to more than one beam or structure, those ports are called -1, STRUCTURE; -1, STRUCT_2; and so forth.

Related Topics

- *Understanding Assemblies and Parts*, page 15
- *Understanding Constraints*, page 21
- *Understanding Joints*, page 22
- *Understanding Part Ports*, page 17

Understanding Constraints

The software uses constraints to control how part symbol ports relate to each other and to reference ports. There are four constraints that can be defined by themselves or in conjunction with another constraint.

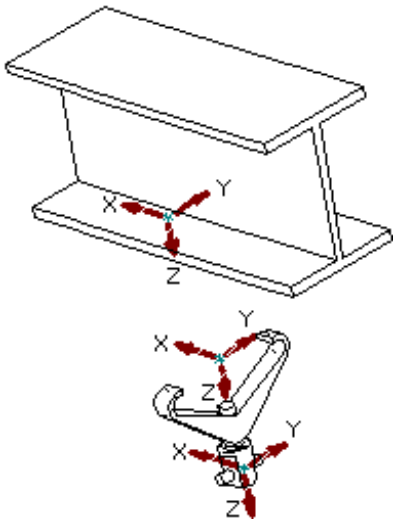
Parallel - Restricts geometries so that they are parallel. The constraint can only be applied between geometries with an associated direction. For ports, these geometries are axes and planes. A plane's direction is defined by its normal vector.

Perpendicular - Restricts geometries so that they are perpendicular. The constraint can only be applied between geometries with an associated direction. For ports, these geometries are axes and planes. A plane's direction is defined by its normal vector.

Coincident - Makes two geometries identical. Coincident constraints are only valid if one geometry can be made to touch everywhere with the other geometry.

Distance - Defines the minimum separation in three dimensions between two geometries. Unlike the other three constraints, this constraint has an associated value. A distance constraint can have different meanings depending on the geometry types and any other applied constraints.

As an example of constraints, consider this beam clamp and beam.



Two constraints allow the beam clamp to stay flush with the face of the beam while still allowing it to slide along the axis of the beam. The first constraint is parallel between the Beam XY-plane and the Beam Clamp XY-plane. The second constraint is coincident between the Beam X-axis and the Beam Clamp X-axis.

Related Topics

- *Understanding Assemblies and Parts*, page 15

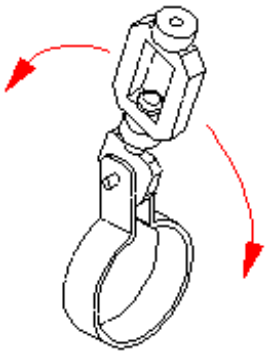
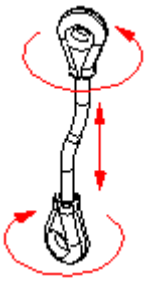
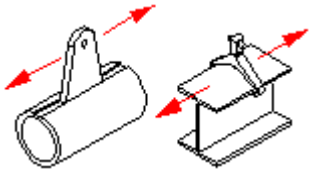
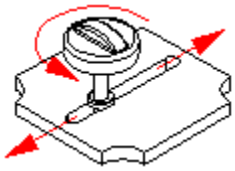
Understanding Joints

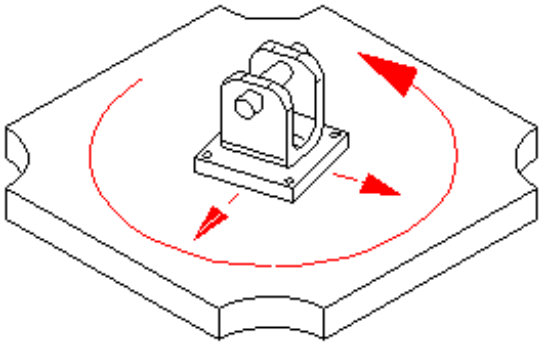
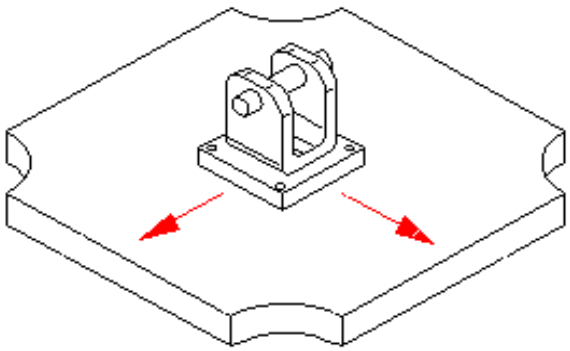
Constraints used between part symbols frequently model the mechanical attachment techniques used to connect the parts. These parts interact in a limited number of ways. The interaction of these part ports, the way in which each part can move with respect to the other part, are defined by *joints*.

In three-dimensional space, there are six degrees of freedom: X, Y, Z, RX, RY, and RZ. A joint restricts the motion of parts with respect to each other in one or more of the these degrees of freedom. By using joints, you do not have to specify individual constraints. However, you do need to specify the joint type, the part ports to join, and the required port geometry. For example, for the revolute joint shown, you need to specify the joint type (revolute), the pipe and turnbuckle ports to join, and the port axis of rotation for the joint (X-axis).

You can define more than one joint per port connection as long as the joints do not conflict with one another.

The following table provides pictures of some common joints.

	
Revolute Joint	*Cylindrical Joint*
	
Prismatic Joint	*Planar Slot Joint*

	
Plane Joint	*Translation Joint*

The next table lists joint and coupling types along with the associated degrees of freedom and a description of each joint.

Joint or Coupling Type	**Input Geometry**	**Associated Offset**	**Translational DOF**	**Rotational DOF**	**Description**
Revolute	1) Axis 2) Origin (implied)	1) No 2) No	0	1	Objects spin about provided axis but do not translate.
Planar	Plane	Yes	2	1	Objects free to rotate and translate while provided planes remain flush.
Translation	1) Plane 2) Axis (in specified plane)	1) Yes 2) No	2	0	Objects may translate but NOT rotate while provided planes remain flush.
Prismatic	1) Plane 2) Axis (in specified plane)	1) Yes 2) Yes	1	0	Objects may slide along provided axis but NOT rotate about the axis.

Joint or Coupling Type	Input Geometry	Associated Offset	Translational DOF	Rotational DOF	Description
Rigid	1) Plane 2) Axis (in specified plane) 3) Origin (implied)	1) Yes 2) Yes 3) Yes	0	0	Objects are rigidly fixed relative to each other.
Spherical	Origin (implied)	No	0	3	Object origins remain in contact with each other while objects are free to spin.
Cylindrical	Axis	Yes	1	1	Objects are free to spin about and translate along provided axis.
Planar Slot	1) Plane 2) Axis (one object only)	1) Yes 2) Yes	1	1	While provided planes remain flush, objects can rotate about their normal and slide along the axis.
Horizontal	Axis/Plane (one object, world XY-plane implied for other)	No	n/a	n/a	Object becomes coincident with or lies in the horizontal plane.

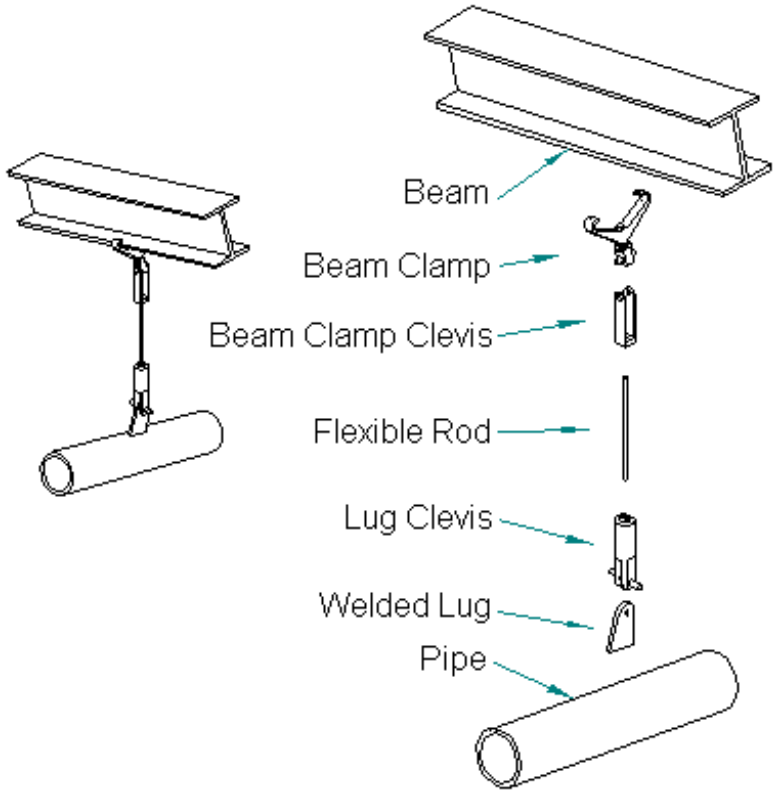
Joint or Coupling Type	Input Geometry	Associated Offset	Translational DOF	Rotational DOF	Description
Vertical	Axis/Plane (one object, world Y-axis implied for other)	No	n/a	n/a	Axis is made parallel to vertical axis; plane is forced to contain vertical axis.

Related Topics

- *SupportJoints Sheet*, page 51
- *Understanding Assemblies and Parts*, page 15
- *Understanding Constraints*, page 21
- *Understanding Part Ports*, page 17
- *Understanding Reference Ports*, page 20

Defining Assemblies: An Overview

Assemblies are a collection of parts that form the physical frame to provide the support. The assembly of a support's constituent parts is done using information defined in the reference data. Each assembly defined in the reference data is assigned an *assembly information rule*. The software uses the logic contained in the assembly information rule to put together the individual support parts to form an assembly.



There are several assembly sheets already defined in the delivered hanger and support reference data. You can find the delivered workbook at *[Product Directory]*\CatalogData\Bulkload\DataFiles\HS_Assembly.xls.

When defining assemblies, the basic parameters are:

Part Class Type - Declare the type of supports. There are several types of supports: PipeSupportDefinitionClass, DuctSupportDefinitionClass, CableTraySupportDefinitionClass, ConduitSupportDefinitionClass, and CombinedSupportDefinitionClass.

The part classes whose names begin with "Assy" are in the PipeSupportDefinitionClass.

Occurrence Attributes - Specify any optional occurrence attributes on the part class level for the support.

Part Number - Type a part number for the assembly. This name must be unique across the entire reference data catalog.

Part Description - Enter a description for the assembly.

Symbol Definition - Not used for assemblies. Leave empty.

Symbol Icon - Not used for assemblies. Leave empty.

Discipline Type - Enter the code that represents the type of object that this assembly can support. Valid codes are listed on the **HngSupDiscipline** sheet in the **AllCodeLists.xls** workbook in the **Codelist Number** column.

Command Type - Enter the code that specifies which Hangers and Supports task command can use the assembly. Valid codes are listed on the **HngSupCommand** sheet in the **AllCodeLists.xls** workbook in the **Codelist Number** column.

Type Selection Rule - Enter the code that specifies the location of the support with respect to the route being supported. Valid codes are listed on the **HngSupTypeSelectionRule** sheet in the **AllCodeLists.xls** workbook in the **Codelist Number** column.

Assembly Info Rule - Specifies the custom Visual Basic Assembly Information Rule to use to construct the assembly. For more information about Assembly Information Rules, see *Assembly Information Rule Example*, page 31.

Max Insulation - Not used. Set to **-1**.

Load Range Low - Enter the minimum load for the assembly. This assembly is not used for loads lighter than this value. Include the units of measurement, lbf for example, when specifying this value.

Load Range High - Enter the maximum load for the assembly. This assembly is not used for loads heavier than this value. Include the units of measurement, lbf for example, when specifying this value.

Minimum Assembly Length - Enter the minimum length that this assembly can be. Include the units of measurement, in or mm for example, when specifying this value.

Maximum Assembly Length - Enter the maximum length that this assembly can be. Include the units of measurement, in or mm for example, when specifying this value.

Supporting Count - Specify the maximum number of supporting objects, a beam for example, to which the assembly can be attached. Enter **1+** to indicate that the support can be attached to a variable number of supporting objects.

Supported Count - Specify the maximum number of supported objects, a pipe for example, to which the assembly can be attached. Enter **1+** to indicate that the support can be attached to a variable number of supported objects.

ND From - Enter the minimum nominal diameter for the supported object. This option is only used for conduit and pipe assemblies.

ND To - Enter the maximum nominal diameter for the supported object. This option is only used for conduit and pipe assemblies.

ND Unit Type - Enter the units for the **ND From** and **ND To** values.

Nominal Width From - Enter the minimum width of the supported object for the assembly. Include the units of measurement, in or mm for example, when specifying this value.

Nominal Width To - Enter the maximum width of the supported object for the assembly. Include the units of measurement, in or mm for example, when specifying this value.

Nominal Depth From - Enter the minimum depth of the supported object for the assembly. Include the units of measurement, in or mm for example, when specifying this value. Enter **0** if the supported object is circular.

Nominal Depth To - Enter the maximum depth of the supported object for the assembly. Include the units of measurement, in or mm for example, when specifying this value. Enter **0** if the supported object is circular.

Supported Family - Enter the type of supported object. Valid types are STRAIGHT and TURN.

Face Selection Type - Enter the face selection combination that you would like to use for the assembly. This column specifies which faces on a structure member that the assembly attaches. These combinations are defined on the **Face Selection** sheet. You enter a number from the **Selection Key** column on the **Face Selection** sheet in this column.

Insulation Purpose - Enter the insulation purpose code for the supported object. Valid codes are listed on the **Insulation Purpose** sheet in the **AllCodeLists.xls** workbook in the **Codelist Number** column.

Support Type - Enter the type for the assembly. Valid codes are listed on the **HngSupSupportType** sheet in the **AllCodeLists.xls** workbook in the **Codelist Number** column.

Mirror Behavior Option - Enter the code that represents the mirror behavior for the assembly. Valid codes are listed in the **AllCodeLists.xls** workbook on the **Mirror Behavior Option** sheet in the **Codelist Number** column.

Related Topics

- *Assembly Information Rule Example*, page 31
- *Hangers and Supports Reference Data: An Overview*, page 11
- *Supports and Local Coordinate System (LCS): An Overview*, page 29

Supports and Local Coordinate System (LCS): An Overview

Every support placed whether it is a Designed Support (DS) or a Standard Support Assembly (Assembly) has a LCS associated with it.

By default, the LCS are always orientated with LCS North along the pipe, LCS east to the side of the pipe and LCS Elevation in the global Up direction. For standard supports on sloped pipe, the LCS North is horizontal, not sloped with the pipe. There is one exception to this, for assemblies on vertical pipe. The LCS Elevation axis is not global Up, but rather to the side of the pipe.

These LCS are provided to facilitate creating drawings with consistent views. When creating a drawing, you can use the **Use object coordinate system** checkbox when creating a view so that that view is always looking at the same side of the route object. For example, if you set the **Orientation** to **Looking North**, that view will always be looking at the end of the route object.

The Assembly Information code for any Standard Support can be modified to change the orientation of the LCS for that particular assembly. To do so, you must add an interface to the module: Implements IJHgrLocalCoordinateSystem and you must implement that interface in the assembly code. For example:

```
Implements IJHgrLocalCoordinateSystem
Private Function IJHgrLocalCoordinateSystem_GetLocalCoodSystem(ByVal
pDispInputConfigHlpr As Object, ByVal pDispPartOccCollection As
Object) As Object
Const METHOD = "IJHgrLocalCoordinateSystem_GetLocalCoodSystem"
On Error GoTo ErrorHandler
   'Create a Joint Factory
   Dim JointFactory As New HgrJointFactory
   'Create an object to hold the Joint
   Dim LocalCSInfo As Object

'=================================================================
=
   '------------Define the local coordinate system configuration----
---
   'NOTE: The format is strictly defined 'Only X axis and XY plane
of local
   coordinate system needs to be defined
   'The first pair is always the reference hanger port index/name
   'It can be reference port ("Route", "Structure", "BBSR_Low") or
existing symbol port
   'Local coordinate system will be referred as -1, and with port
name "LocalCS"
   'Following is an example of the allowed format:
   '=============reference port ================= local coordinate
system=========
   '===index===== 1 ================ -1 ========
```

```
   '===port name= Structure ================= LocalCS =========
   '===Plane===== YZ ================= XY =========
   '===Align===== False
   '===Axis====== Y ================= X =========
   '===Align===== True
   '===ConfigureIndex No: 9510
   Set LocalCSInfo = JointFactory.MakeRigidJoint(1, "Route", -1,
"LocalCS", 11501)
   Set IJHgrLocalCoordinateSystem_GetLocalCoodSystem = LocalCSInfo
Exit Function
ErrorHandler:
   Err.Raise LogError(Err, MODULE, METHOD, "").Number
End Function
```

Related Topics

- *Defining Assemblies: An Overview*, page 26

Assembly Information Rule Example

The assembly information for each support type is defined in assembly reference data using the AssmInfoRule attribute.

The AssmInfoRule attribute (short for Assembly Information Rule) contains the ProgId of a COM object implementing the IJHgrAssmInfo interface. In most cases, the rule is a Visual Basic class. The purpose of an Assembly Information Rule is to provide any required information about the assembly during the evaluation of a support. Information is acquired by calling the methods defined on the IJHgrAssmInfo interface.

There are five methods defined within the Visual Basic program:

- IJHgrAssmInfo::GetAssemblyCatalogParts() defines the part class and retrieves the parts.
- IJHgrAssmInfo::GetAssemblyJoints() defines the constraints between the parts.
- IJHgrAssmInfo::GetRouteConnections() defines which part is connected to the route.
- IJHgrAssmInfo::GetStructConnections() defines which part is connected to the structure.
- IJHgrAssmInfo::GetMaxRouteConnectionValue() defines the route toggle configurations supported by the Assembly Information Rule.

GetAssemblyCatalogParts Method

The GetAssemblyCatalogParts method of the IJHgrAssmInfo interface is responsible for specifying the catalog parts constituting the assembly. The method is defined as:

```
Function IJHgrAssmInfo_GetAssemblyCatalogParts (
         ByVal pDispInputConfigHlpr As Object
         ) As Object
```

The method should return a collection of objects implementing the IJDPart interface. The following shows a sample implementation of this method for the hanger support assembly shown in Figure 1.

```
Implements IJHgrAssmInfo
Private Function IJHgrAssmInfo_GetAssemblyCatalogParts(ByVal
pDispInputConfigHlpr As Object) As Object
'Get IJHgrInputConfig Hlpr Interface off of passed Helper
Dim my_IJHgrInputConfigHlpr As IJHgrInputConfigHlpr
Set my_IJHgrInputConfigHlpr = pDispInputConfigHlpr
'Create a new collection to hold the catalog parts
Dim CatalogPartCollection As New Collection
'Create the list of part classes required by the support assembly
```

```
Dim PartClasses(3) As String
PartClasses(1) = "Beam_Clamp"
PartClasses(2) = "Rod"
PartClasses(3) = "Pipe_Clamp"
'Dimension Object to temporarily hold a Catalog Part
Dim PartProxy As Object
'Use the default selection rule to get a catalog part for each part
class
Dim index As Integer
For index = 1 To UBound(PartClasses)
     Set PartProxy =
my_IJHgrInputConfigHlpr.GetPartFromClass(PartClasses(index))
     CatalogPartCollection.Add PartProxy
Next
'Return the collection of Catalog Parts
Set IJHgrAssmInfo_GetAssemblyCatalogParts = CatalogPartCollection
End Function
```

Arguments for the method are limited to a single object named the Input Configuration Helper. This object offers access to the input specified by the user during the creation of the support as well as any services required by the Assembly Information Rule. In the above example, the GetPartFromClass method of the Input Configuration Helper is used to obtain a catalog part given a part class name.

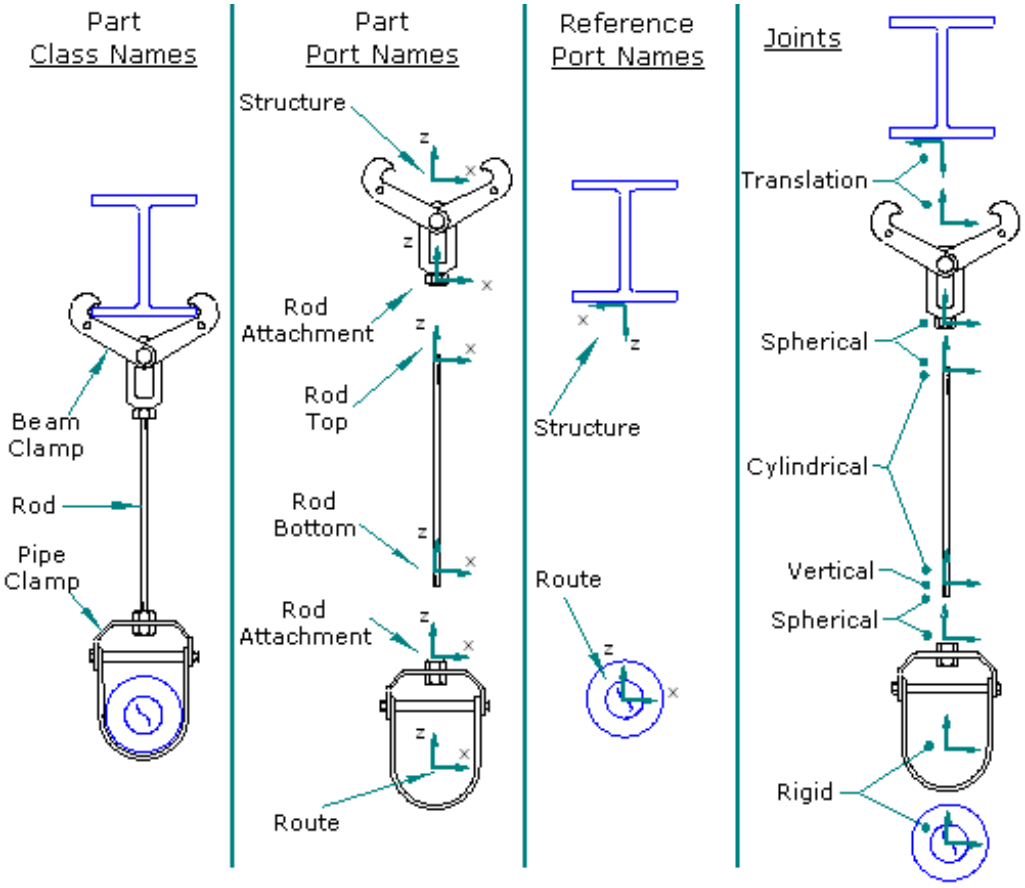


Figure 1: Hanger Support with Associated Ports and Joints

GetAssemblyJoints Method

The GetAssemblyJoints method of the IJHgrAssmInfo interface provides the joints required to assemble the support. The method is defined as:

```
Function IJHgrAssmInfo_GetAssemblyJoints (
         ByVal pDispInputConfigHlpr As Object
         ByVal pDispPartOccCollection As Object
         ) As Object
```

The method should return a collection of objects implementing the IJHgrJoint interface. The following shows a sample implementation of this method for the hanger support assembly shown in Figure 1.

```
Implements IJHgrAssmInfo
Private Function IJHgrAssmInfo_GetAssemblyJoints( ByVal
pDispInputConfigHlpr As Object,
ByVal pDispPartOccCollection As Object ) As Object
'Get IJHgrInputConfig Hlpr Interface off of passed Helper
Dim my_IJHgrInputConfigHlpr As IJHgrInputConfigHlpr
Set my_IJHgrInputConfigHlpr = pDispInputConfigHlpr
'Set Attributes on Part Occurrence
'====================
'Get the Diameter of the Primary Pipe
Dim PipeRadius As Double
PipeRadius = (my_IJHgrInputConfigHlpr.PrimaryPipeDiameter) / 2#
'Get interface for accessing items on the collection of Part
Occurrences
Dim IJElements_PartOccCollection As IJElements
Set IJElements_PartOccCollection = pDispPartOccCollection
'Get the Pipe Clamp from the collection. It will be the third Part
Occurrence.
Dim PipeClamp As Object
Set PipeClamp = IJElements_PartOccCollection.Item(3)
'Set the Pipe Radius Attribute on the Pipe Clamp Symbol
Dim strPipeRadius As String
strPipeRadius = "PipeRadius"
my_IJHgrInputConfigHlpr.SetSymbolInputByName PipeClamp,
strPipeRadius, PipeRadius
'Create Joints
'========
'Create a collection to hold the joints
Dim JointCollection As New Collection
'Create a Joint Factory
Dim JointFactory As New HgrJointFactory
'Create an object to hold the Joint
Dim AssemblyJoint As Object
'Create the Flexible (Cylindrical) Joint between the ports of the
rod
Set AssemblyJoint = JointFactory.MakeCylindricalJoint(2, "RodTop",
2, "RodBottom")
JointCollection.Add AssemblyJoint
'Add a Rigid Joint between Pipe and Pipe Clamp
Set AssemblyJoint = JointFactory.MakeRigidJoint(3, "PipeAttachment",
-1, "Pipe")
JointCollection.Add AssemblyJoint
```

```
'Add a Translation Joint between the Beam and the Beam Clamp
Set AssemblyJoint = JointFactory.MakePrismaticJoint(1,
"BeamAttachment", -1, "Structure")
JointCollection.Add AssemblyJoint
'Add a Spherical Joint between Beam Clamp and Top of Rod
Set AssemblyJoint = JointFactory.MakeSphericalJoint(1,
"RodAttachment", 2, "RodTop")
JointCollection.Add AssemblyJoint
'Add a Spherical Joint between Pipe Clamp and Bottom of Rod
Set AssemblyJoint = JointFactory.MakeSphericalJoint(3,
"RodAttachment", 2, "RodBottom")
JointCollection.Add AssemblyJoint
'Add a Vertical Joint to the Rod Z axis
Set AssemblyJoint = JointFactory.MakeVerticalJoint(2, "RodBottom")
JointCollection.Add AssemblyJoint
'Return the collection of Joints
Set IJHgrAssmInfo_GetAssemblyJoints = JointCollection
End Function
```

Two objects are passed as input to the method. The first is the Input Configuration Helper. The second is a collection of part occurrences representing the support. The part occurrences in this collection are listed in same order as the catalog parts returned by the GetAssemblyCatalogParts method. Hence, the first part occurrence corresponds to the first catalog part, the second part occurrence corresponds to the second catalog part, and so on. The part occurrences are provided to allow the Assembly Information Rule an opportunity to set inputs on the part occurrences' associated symbol. In the previous example, the "PipeRadius" input is set on the pipe clamp using information provided by the Input Configuration Helper.

The collection returned by the method contains a list of Hanger Joint objects. These objects are created using a Joint Factory (a HgrJointFactory object). The Factory implements the IJHgrJointFactory interface, which contains the following methods:

```
Function MakeRevoluteJoint (
         Integer PartIndex_A, String PortName_A,
         Integer PartIndex_B, String PortName_B,
         Integer ConfigIndex ) As Object
Function MakePrismaticJoint (
         Integer PartIndex_A, String PortName_A,
         Integer PartIndex_B, String PortName_B,
         Integer ConfigIndex,
         Double PlaneOffset, Double AxisOffset ) As Object
Function MakeRigidJoint (
         Integer PartIndex_A, String PortName_A,
         Integer PartIndex_B, String PortName_B,
         Integer ConfigIndex, Double PlaneOffset,
         Double AxisOffset, Double OriginOffset ) As Object
Function MakeTranslationJoint (
         Integer PartIndex_A, String PortName_A,
         Integer PartIndex_B, String PortName_B,
         Integer ConfigIndex,
         Double PlaneOffset ) As Object
Function MakeVerticalJoint (
         Integer PartIndex_A, String PortName_A,
         Integer ConfigIndex ) As Object
```

```
Function MakeSphericalJoint (
         Integer PartIndex_A, String PortName_A,
         Integer PartIndex_B, String PortName_B ) As Object
Function MakeCylindricalJoint (
         Integer PartIndex_A, String PortName_A,
         Integer PartIndex_B, String PortName_B,
         Integer ConfigIndex,
         Double AxisOffset) As Object
Function MakePrismaticJoint (
         Integer PartIndex_A, String PortName_A,
         Integer PartIndex_B, String PortName_B,
         Integer ConfigIndex,
         Double PlaneOffset, Double AxisOffset,
         Double OriginOffset ) As Object
Function MakePlanarJoint (
         Integer PartIndex_A, String PortName_A,
         Integer PartIndex_B, String PortName_B,
         Integer ConfigIndex,
         Double PlaneOffset ) As Object
```

These methods create CHgrJoint objects. A CHgrJoint object retains sufficient information to define the constraints implementing any joint. Member data for the CHgrJoint object includes:

Joint Type (1): an enum describing the type of joint. The IJHgrJointFactory method names indicate the enum stored.

Symbol Indices (2): Symbols will be indicated as indices into the catalog part list provided as output to GetAssemblyCatalogParts method. These are the PartIndex_A and PartIndex_B arguments within the IJHgrJointFactory method prototypes.

Port Names (2): Port names indicate which output of the specified symbol to use when constructing the joint. These are the PortName_A and PortName_B arguments within the IJHgrJointFactory method prototypes.

Configuration Index (1): Each joint type requires a defining set of port geometries. For example, the prismatic joint requires a plane and axis be specified for each of the two participating ports. Because the geometry associated with each port is limited (3 planes and 3 axes), all possible combinations can be encoded in a single 32-Bit integer referred to as the Configuration Index. Configuration Indices can be calculated using a utility provided by development called the Configuration Index Calculator. If no Configuration Index is provided when calling an IJHgrJointFactory method, a default representing the most common case is used.

Offsets (3): Several joint types permit the specification of offsets between their defining geometries. For the most complex case, three offsets are allowed. Offsets are represented using three doubles. These are the PlaneOffset, AxisOffset and OriginOffset arguments within the IJHgrJointFactory method prototypes.

GetRouteConnections Method

The GetRouteConnections method of the IJHgrAssmInfo interface provides information specifying those components of the assembly that physically connect to the supported input objects. The method is defined as:

```
Function IJHgrAssmInfo_GetRouteConnections ( ByVal
pDispInputConfigHlpr As Object
) As Object
```

The method should return a collection of integer arrays. The following shows a sample implementation of this method for the structural support assembly shown in Figure 2.

```
Private Function IJHgrAssmInfo_GetRouteConnections( ByVal
pDispInputConfigHlpr As Object ) As Object
'Create a collection to hold the ALL Route connection information
Dim RouteCollofConnInfo As New Collection
'Create an ARRAY to hold Route Connection information for the Cross
Bar
Dim CrossBarConnectionInfo(4) As Integer
CrossBarConnectionInfo (1) = 3 'Third Part return by
GetAssemblyCatalogParts() is Cross Bar
CrossBarConnectionInfo (2) = 1 'Connects to First Route Reference
Port
CrossBarConnectionInfo (3) = 2 'Connects to Second Route Reference
Port
CrossBarConnectionInfo (4) = 3 'Connects to Third Route Reference
Port
'Create an ARRAY to hold Route Connection information for the Left
Ubolt
Dim Left_UBolt_ConnectionInfo(2) As Integer
Left_UBolt_ConnectionInfo(1) = 7 'Seventh Part return by
GetAssemblyCatalogParts() is left most UBolt
Left_UBolt_ConnectionInfo(2) = 1 'Connects to First Route Reference
Port
'Create an ARRAY to hold Route Connection information for the Middle
Ubolt
Dim Middle_UBolt_ConnectionInfo(2) As Integer
Middle_UBolt_ConnectionInfo (1) = 8 'Eight Part return by
GetAssemblyCatalogParts() is middle UBolt
Middle_UBolt_ConnectionInfo (2) = 2 'Connects to Second Route
Reference Port
'Create an ARRAY to hold Route Connection information for the Right
Ubolt
Dim Right_UBolt_ConnectionInfo(2) As Integer
Right_UBolt_ConnectionInfo (1) = 9 'Ninth Part return by
GetAssemblyCatalogParts() is right most UBolt
Right_UBolt_ConnectionInfo (2) = 3 'Connects to Third Route
Reference Port
'Add the Connection Information arrays to the returned Collection.
RouteCollofConnInfo.Add CrossBarConnectionInfo
RouteCollofConnInfo.Add Left_UBolt_ConnectionInfo
RouteCollofConnInfo.Add Middle_UBolt_ConnectionInfo
RouteCollofConnInfo.Add Right_UBolt_ConnectionInfo
'Return the collection of Route connection information.
```

```
Set IJHgrAssmInfo_GetRouteConnections = RouteCollofConnInfo
End Function
```

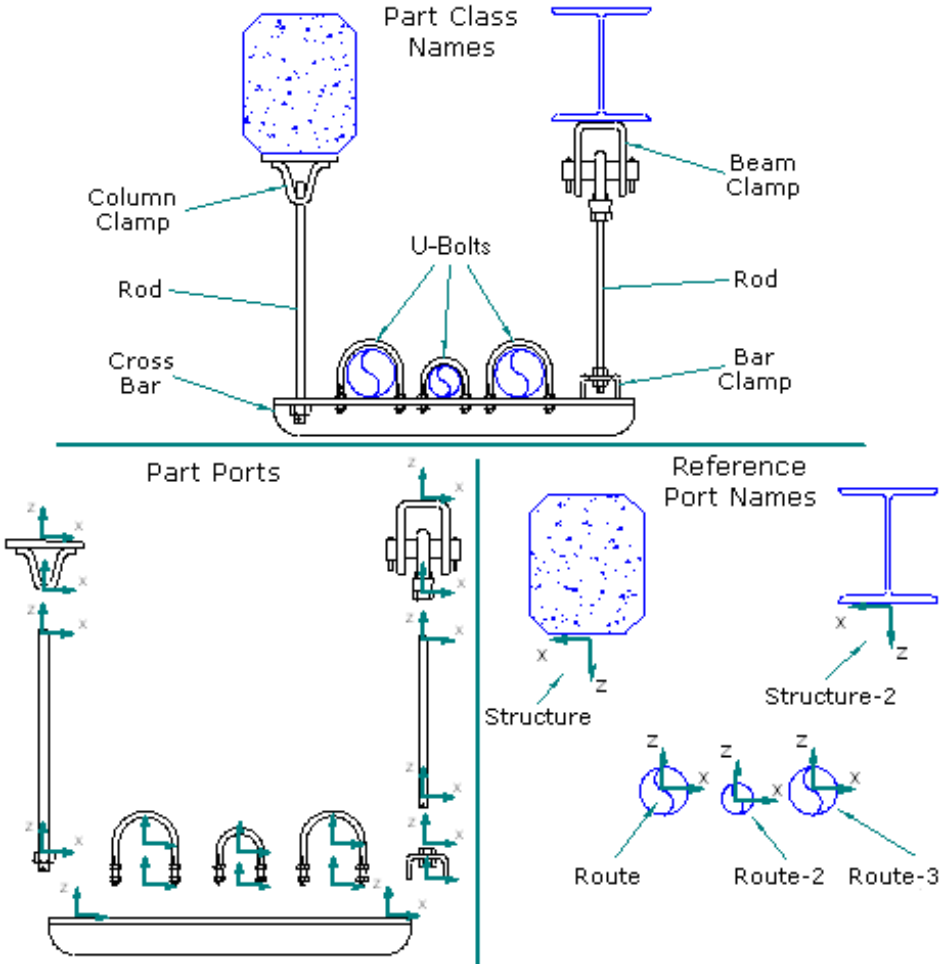


Figure 2: Trapeze Support with Associated Ports

In previous example, four arrays of integers are returned. These arrays correspond to the four assembly components physically touching the pipes: the cross bar and the three U-bolts. The first entry in each array is the index of the assembly part as referenced from the list returned in the GetAssemblyCatalogParts method. For the example, we assume the following part order (see Figure 2):

1. Column Clamp
2. Rod
3. Cross Bar
4. Bar Clamp
5. Rod
6. Beam Clamp
7. Left U-Bolt

8. Middle U-Bolt
9. Right U-Bolt

All entries following the first in each array refer to the route reference ports to which the assembly component connects. In the example, the first array returns: (3, 1, 2, 3). This is interpreted as: the third part returned by GetAssemblyCatalogParts (the cross bar), connects with the first, second, and third route reference ports (the left, middle, and right pipes). Similar arrays are returned for each of the U-Bolts; however, these connect with only a single pipe: (7, 1). This is interpreted as: the seventh part connects with the first route reference port.

GetStructConnections Method

The GetStructConnections method of the IJHgrAssmInfo interface provides information specifying those components of the assembly that physically connect to the supporting input objects. The method is defined as:

```
Function IJHgrAssmInfo_GetStructConnections ( ByVal
pDispInputConfigHlpr As Object) As Object
```

The method should return a collection of integer arrays. The following shows a sample implementation of this method for the structural support assembly shown in Figure 2.

```
Private Function IJHgrAssmInfo_GetStructConnections( ByVal
pDispInputConfigHlpr As Object ) As Object
'Create a collection to hold the ALL Structure connection
information
Dim StructCollofConnInfo As New Collection
'Create an ARRAY to hold Structural Connection information for the
Column Clamp
Dim ColumnClampConnectionInfo(2) As Integer
ColumnClampConnectionInfo (1) = 1 'First Part return by
GetAssemblyCatalogParts() is Column Clamp
ColumnClampConnectionInfo (2) = 1 'Connects to First Structural
Reference Port
'Create an ARRAY to hold Structural Connection information for the
Beam Clamp
Dim BeamClampConnectionInfo(2) As Integer
BeamClampConnectionInfo (1) = 6 'Sixth Part return by
GetAssemblyCatalogParts() is Beam Clamp
BeamClampConnectionInfo (2) = 2 'Connects to Second Structural
Reference Port
'Add the Connection Information arrays to the returned Collection.
StructCollofConnInfo.Add ColumnClampConnectionInfo
StructCollofConnInfo.Add BeamClampConnectionInfo
'Return the collection of Struct connection information.
Set IJHgrAssmInfo_GetStructConnections = StructCollofConnInfo
End Function
```

The format of the returned integer arrays is identical to those returned in the GetStructConnections method described above. For example, the two arrays returned in the above example correspond to the two parts physically touching the structural inputs, the column clamp and the beam clamp. The second of these arrays contains the following entries: (6, 2). This is interpreted as: the sixth part returned by GetAssemblyCatalogParts (the beam clamp), connects with the second structural reference ports (the beam).

Related Topics

- *Defining Assemblies: An Overview*, page 26

Assembly Selection Rules

Assembly selection rules help the person placing supports in the model by selecting appropriate support assemblies based on the feature (pipe, HVAC duct, or cable tray) and the structure that the person selected and then displaying those selected assemblies in the **Type** box on the ribbon. The person placing supports can control whether or not assembly selection rules are used based on the setting of the **Rule** option on the ribbon.

The standard hangers and supports commands implement the GetAttributeByRule function of the IJHgrInputObjectInfo interface. This function calls the GetAttributeByRule function of the CatalogServices. The catalog service reads the HgrRule table, gets the progID of the feature rule, creates the feature rule object, and finally calls the function AttributeValue on the IJHgrSupportRule interface, which returns the feature type of the selected feature.

```
Private Function IJHgrInputObjectInfo_GetAttributeByRule(
        ByVal RuleName As String,
        Optional ByVal pInputContext As Object = Nothing) As
Variant()
     Const METHOD = "IJHgrInputObjectInfo_GetAttributeByRule"
     On Error GoTo ErrorHandler
     Dim oHgrCatalogServices As IJHgrCatalogServices
     Set oHgrCatalogServices = New CatalogServices
     IJHgrInputObjectInfo_GetAttributeByRule =
     oHgrCatalogServices.GetAttributeByRule(RuleName, pInputContext)
     Set oHgrCatalogServices = Nothing
     Exit Function
ErrorHandler:
     ReportUnanticipatedError MODULE, METHOD
End Function
```

Therefore, when the **Rule** option is selected on the ribbon, the command creates the assembly selection rule object, and then calls the GetAssemblies function on the interface IJHgrAssmSelectionRule by passing its IJHgrInputObjectInfo interface. The assembly selection rule calls GetAttributeByRule function on IJHgrInputObjectInfo interface to obtain the feature type. The assembly selection rule then gets and returns the assembly based on the feature type from the catalog.

Related Topics

- *Defining Assemblies: An Overview*, page 26

Defining Parts: An Overview

Parts are used to create the hanger and support assemblies that are placed in the model.

The delivered reference data contains a few workbooks whose main purpose is to define parts. As an example, we will take the **Utility_VARIABLE_CYL** sheet in **HS_Utility.xls**. This workbook defines part classes that have the following common properties. Depending on the attributes that you define when you create a part symbol, you may have to specify additional or different attributes than the ones listed below.

Notes

- Various custom attributes are often listed for the parts. An example is IJOAHgrUtility_VARIABLE_CYL::RADIUS.
- For some of the properties listed below, *(i)* is an integer corresponding to a port. When you type the port properties into the workbook, substitute a number for *(i)*.

Part Class Type - Specifies the class type of the part.

Symbol Definition - Specifies the symbol used to represent the part. For more information on creating part symbols, refer to the *SmartPlant 3D Symbols Reference Guide*.

Symbol Icon - Specifies the graphic file that contains a picture of the part symbol. The graphic file is used as a preview in the software. Specify the path to the graphic file relative to the symbols folder that is shared on your software server.

Occurrence Attributes - Depending on the part symbol, there may be one or more occurrence attributes to define.

Class Type - Enter the class type code for the part. Valid codes are listed on the **HngSupPartClassType** sheet in the **AllCodeLists.xls** workbook in the **Codelist Number** column.

Part Selection Rule - Specify the part selection rule. The **Part Selection Rule** automates the selection of items from the catalog using certain criteria. For example, pipe size (HgrPipePartSelRule.PartByPipeSize.cls), rod attachment size (HgrPipePartSelRule.CPartByRodSize.cls), the beam profile rule (HgrPipePartSelRule.CHgrBeamForVPad.cls), by load factor (HgrPipePartSelRule.CPartByLoadFactor.xls), and so forth.

Port Type - Enter the type of port. For example, type **HgrSymbolPort** to specify that the symbol port is a hangers and supports port as opposed to a piping port.

User Class Name - Specify the user class name.

Part Number - Type a part number. This part number must be unique across the entire reference data catalog.

Part Description - Enter a description for the part.

Symbol Definition - Specifies the symbol used to represent the part. You can use this field to override the symbol definition for the part class and to instead use a different symbol for an individual part number.

ND From - Enter the minimum nominal diameter for which the part can be used. This option is only used for conduit and pipe assemblies.

ND To - Enter the maximum nominal diameter for which the part can be used. This option is only used for conduit and pipe assemblies.

ND Unit Type - Enter the units for the **ND From** and **ND To** values.

For every port *(i)* in the symbol, several attributes must be defined as below.

HgrSymbolPort(i):Name – Enter the name of this port. Any name is valid. This name must be unique for each port name column. The name will be used when defining joints in Assembly Information Rules.

HgrSymbolPort(i):Category – Enter the code that represents the category for the port. This attribute is no longer used but exists for backward compatibility. Valid codes are listed in the **AllCodeLists.xls** workbook on the **HngSupPortCategory** sheet in the **Codelist Number** column.

HgrSymbolPort(i):MatingType – Enter the code that represents the mating type for the port. This attribute is no longer used but exists for backward compatibility. Valid codes can be listed in the **HS_System.xls** workbook on the **SupportJoints** sheet.

HgrSymbolPort(i):ConnectionInfo – Enter the code that represents the connection information for the port. This attribute is no longer used but exists for backward compatibility. Valid codes are listed in the **AllCodeLists.xls** workbook on the **HngSupPhysicalConnection** sheet in the **Codelist Number** column.

HgrSymbolPort(i):PortType – Enter the code that represents the port type for the port. Valid codes are listed in the **AllCodeLists.xls** workbook on the **HngSupPortType** sheet in the **Codelist Number** column.

HgrSymbolPort(i):Size – Enter the physical size of this port. For example, for a pin, enter its diameter.

HgrSymbolPort(n):MinSize – Enter the minimum size of the part port that can connect to this port. For example, for a pin port, the minimum size of the eye that can connect to it.

HgrSymbolPort(i):MaxSize – Enter the maximum size of the part port that can connect to this port. For example, for a pin port, enter the maximum size of the eye that can connect to it.

HgrSymbolPort(n):UnitType – Enter a string representing the units for the port Size, MinSize and MaxSize attributes. Currently, only **in** and **mm** are valid strings for these sizes.

Dry Weight - Enter the dry weight of the support component.

Water Weight - Enter the wet weight of the support component.

DryCogX - Enter the X-axis location of the dry center-of-gravity.

DryCogY - Enter the Y-axis location of the dry center-of-gravity.

DryCogZ - Enter the Z-axis location of the dry center-of-gravity.

Mirror Behavior Option - Enter the code that represents the mirror behavior for the part. Valid codes are listed in the **AllCodeLists.xls** workbook on the **Mirror Behavior Option** sheet in the **Codelist Number** column.

IJHgrBOMDefinition::BOMType (Optional) – Enter the code that defines how the **BOMDescription** attribute for this part will be set. Valid codes are listed in the **AllCodeLists.xls** workbook on the **HngSupBOMType** sheet. If no value is entered, the BOM description for the part will default to the **PartDescription** text.

IJHgrBOMDefinition::BOMDefinition (Optional) - Specifies the BOM description. If **BOMType** is set to 4 (FromBOMDefProgID), enter the ProgID to use for calculating the BOM description. If **BOMType** is set to 5 (FromBOMDef), enter the text to use for the BOM description. You do not need to enter anything in this column for other values of the **BOMType** attribute.

Related Topics

- *Hangers and Supports Reference Data: An Overview*, page 11

Lookup Tables: An Overview

Lookup tables are used by the software to place supports in the model. The lookup tables are in the **HS_System.xls** workbook.

Related Topics

- *Face Position Selection Sheet*, page 49
- *Face Selection Sheet*, page 45
- *Rules Sheet*, page 50
- *SupportJoints Sheet*, page 51

Face Selection Sheet

The **Face Selection** sheet in the **HS_System.xls** workbook defines the faces of a structural member to which a part can attach itself.

Face Name - Enter a unique name for the face selection option. This name must be unique across the entire catalog.

Description - Type a description for the face selection option.

Cross Section - Enter the cross section shape to which the face selection option applies. For example, W or L.

Selection Key - Enter a code to link this face selection option to parts. All face selection options with the same selection key code, 1 for example, will be made available to the part that has this code defined.

Face 1 - Enter the member face with the highest priority. This face will be default when the assembly is placed.

Group 1 - Enter the group key code. Valid codes are listed on the **Face Position Selection** sheet in the **Group Key** column. For more information about this code, see *Face Position Selection Sheet*, page 49.

Face 2 - Enter the member face with the next highest priority. This face will be the second choice when the assembly is placed. Enter 0 if you do not want to define another choice.

Group 2 - Enter the group key code. Valid codes are listed on the **Face Position Selection** sheet in the **Group Key** column.

Face 3 - Enter the member face with the second highest priority. This face will be the third choice when the assembly is placed. Enter 0 if you do not want to define another choice.

Group 3 - Enter the group key code. Valid codes are listed on the **Face Position Selection** sheet in the **Group Key** column.

Face 4 - Enter the member face with the third highest priority. This face will be the fourth choice when the assembly is placed. Enter 0 if you do not want to define another choice.

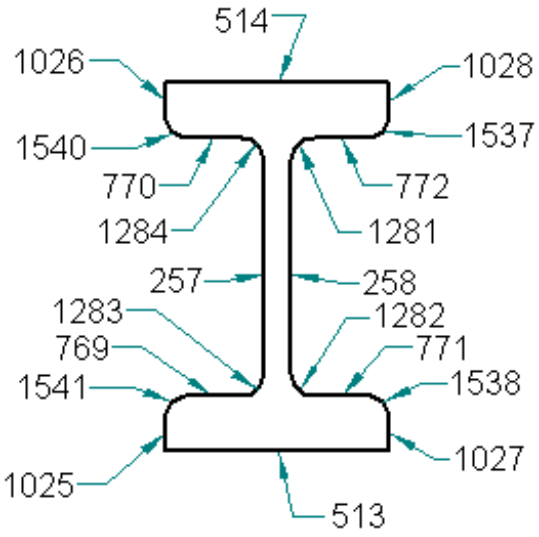
Group 4 - Enter the group key code. Valid codes are listed on the **Face Position Selection** sheet in the **Group Key** column.

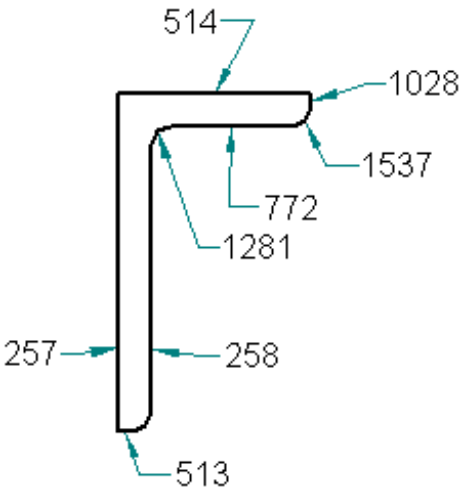
Face 5 - Enter the member face with the lowest priority. This face will be the last choice when the assembly is placed. Enter 0 if you do not want to define another choice.

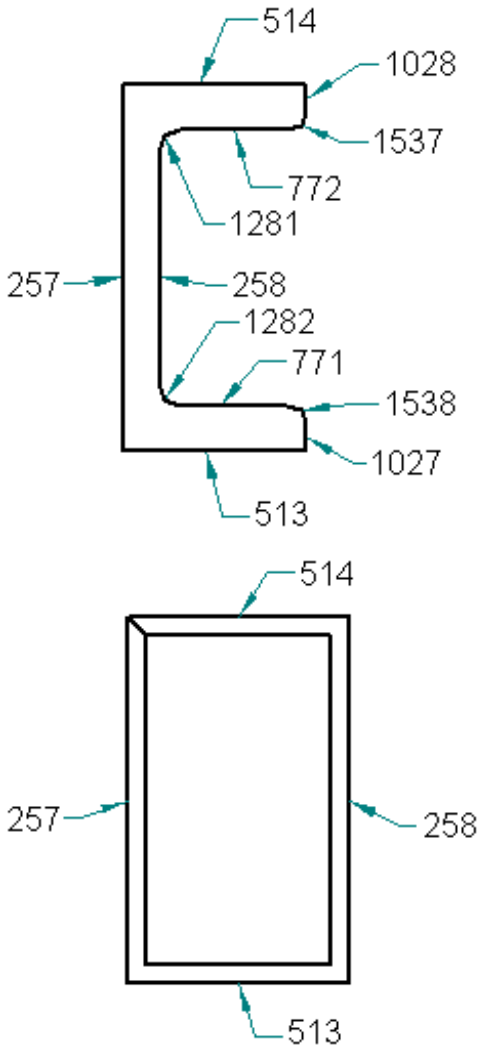
Group 5 - Enter the group key code. Valid codes are listed on the **Face Position Selection** sheet in the **Group Key** column.

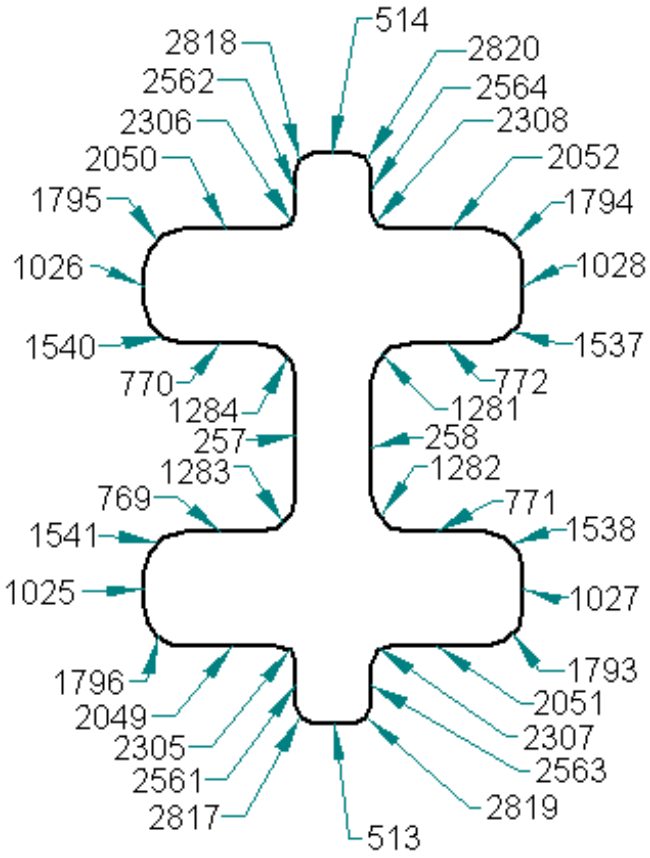
Available Face Numbers

These figures show the available face numbers for common section shapes that you can enter in the **Face 1-5** boxes.






514
1028
1537
772
1281
257
258
1282
771
1538
1027
513
514
257
258
513



Related Topics

- *Lookup Tables: An Overview*, page 44

Face Position Selection Sheet

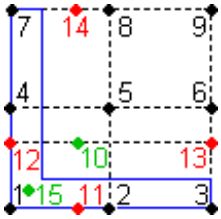
The **Face Position Selection** sheet in the **HS_System.xls** workbook defines the position on a face of a structure member where the part can locate itself. An example of when to use this sheet would be a U-bolt connected to the bottom of an I-shaped member. You would want the U-bolt to be placed on the out edges of the bottom flange so that the U-bolt did not interfere with the web of the member.

Point Sel Name - Enter a unique name for the face position selection option. This name must be unique across the entire catalog.

Description - Type a description for the face position selection option.

Group Key - Enter the group key for the face position selection option. This key is used to group face positions selection options together from which the user can select one of the face positions. This group key is also entered on the **Face Selection** sheet in the **Group 1-5** columns.

Cardinal Point - Enter the member cardinal point from which to base the face position. There are 15 cardinal positions available. The location of cardinal points 10 (center-of-gravity) and 15 (shear center) depend on the section shape. The local z-axis of the member and the center-of-gravity point of the section define cardinal points 11 and 14. The local y-axis of the member and the center-of-gravity point of the section define cardinal points 12 and 13.



Offset - Enter the offset rule to use for the face position selection option (X-direction). The offset is measured from the cardinal point that you specified. Offset and OffsetY do not require a defined progID. The column can contain a numeric value.

OffSetY - Enter the offset rule to use for the Y-direction (perpendicular to face). Offset and OffsetY do not require a defined progID. The column can contain a numeric value.

Related Topics

- *Lookup Tables: An Overview*, page 44

Rules Sheet

The **Hgr Rules** sheet in the **HS_System.xls** workbook is a lookup table for rules.

Rule Name - Enter a unique name for the rule. This name must be unique across the entire catalog.

Rule Description - Type a description for the rule.

Rule Type - Enter the code for the rule type. Valid codes are listed on the **HngSupRuleType** sheet in the **AllCodeLists.xls** workbook in the **Codelist Number** column.

Rule Value - Enter a value for the rule. For example, if you entered **5** in the **Rule Type** column, you enter the program ID in this column.

Related Topics

- *Lookup Tables: An Overview*, page 44

SupportJoints Sheet

The **SupportJoints** sheet in the **HS_System.xls** workbook defines the joints that you can use to connect two parts. The sheet is used when you are defining supports by adding parts using the **Place Part** command.

Joint Key - Specifies the name of the joint key. This name must be unique within the column.

Mating Type 1 - Enter the first mating type code for the joint. Valid codes are listed on the **HngSupPortType** sheet in the **AllCodeLists.xls** workbook in the **Codelist Number** column.

Mating Type 2 - Enter the second mating type code for the joint. Valid codes are listed on the **HngSupPortType** sheet in the **AllCodeLists.xls** workbook in the **Codelist Number** column.

Mating Option - Specifies the mating option. This value is not currently used in the software.

Command Type - Enter the code that specifies which Hangers and Supports task command can use the mating type combination. Valid codes are listed on the **HngSupCommand** sheet in the **AllCodeLists.xls** workbook in the **Codelist Number** column. This value is not currently used in the software.

Joint Type - Specifies the joint type by the joint code. Valid codes are listed on the **HngSupJointType** sheet in the **AllCodeLists.xls** workbook in the **Codelist Number** column. For more information about joints, see *Understanding Joints*, page 22. This value is not currently used in the software.

A short description of each joint type is listed below:

Code	Joint	Description
1	Revolute	Objects can spin about the provided axis but do not translate.
2	Planar	Objects are free to rotate and translate while provided planes remain flush.
3	Translation	Objects can translate but not rotate while provided planes remain flush.
4	Prismatic	Objects can slide along provided axis but cannot rotate about the axis.
5	Rigid	Objects are rigidly fixed relative to each other.
6	Spherical	Object origins remain connected while objects are free to spin.
7	Cylindrical	Objects are free to spin about and translate along provided axis.

Code	Joint	Description
8	Planar Slot	While provided planes remain flush, objects can rotate about their normal and slide along the axis.
9	Vertical	Object axis is made parallel to vertical axis, and the plane is forced to contain the vertical axis.
10	Horizontal X	Object becomes coincident with or lies in the horizontal plane.
11	Horizontal Y	Object becomes coincident with or lies in the horizontal plane.
12	Point On	Objects are point onto each other.

Configuration Index - Enter the configuration index number to use for the joint. You can calculate the configuration index number using the *Configuration Index Calculator*, page 52. This value is not currently used in the software.

Offset 1 - Specifies the plane offset between the parts at the joint. This value is not currently used in the software.

Offset 2 - Specifies the axis offset between the parts at the joint. This value is not currently used in the software.

Offset 3 - Specifies the origin offset between the parts at the joint. This value is not currently used in the software.

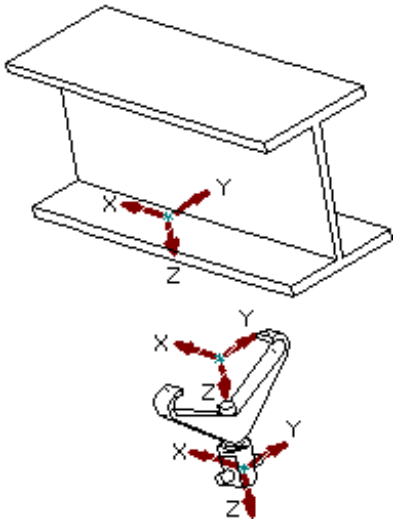
Related Topics

- *Lookup Tables: An Overview*, page 44
- *Understanding Joints*, page 22

Configuration Index Calculator

The Configuration Index Calculator calculates the configuration index number used on the **Support Joint** sheet and in assembly information rules. The configuration index is an integer that specifies the geometry associated with each port in the joint. For example, the prismatic joint requires a plane and an axis be specified for each port. The configuration index tells the software which port plane and which port axis to use for each part and how those planes and axes are constrained. The Configuration Index Calculator is delivered with the software in the *[Product Directory]*\HangersAndSupports\Tools\Bin folder.

For an example of how to use the Configuration Index Calculator, look at the connection between this beam clamp and the beam. Two constraints allow the beam clamp to stay flush with the face of the beam while still allowing it to slide along the axis of the beam. The first constraint is parallel between the Beam XY-plane and the Beam Clamp XY-plane. The second constraint is coincident between the Beam X-axis and the Beam Clamp X-axis.



Configuration Index - Displays the configuration index number that you should enter on the **Support Joint** sheet in the **Configuration Index** column.

Extract Data - Calculates the configuration index number based on the inputs that you provide.

Geometry Type A - Select the geometry type for one of the port constraints in the connection. Using the example, you would select XY-Plane to represent the XY-Plane of the beam clamp port.

Geometry Type B - Select the geometry type for the other constraint port in the connection. Using the example, you would select XY-Plane to represent the XY-Plane of the beam port.

A and B Alignment - Select how you want the Geometry Type A and Geometry Type B selections to align. Using the example, you would select Opposite / Mate because you want the two XY-Planes to be co-planar, or mate.

Geometry Type C - Select the geometry type for the other port constraint in the connection. Using the example, you would select X-axis for the X-axis of the beam clamp port. If the connection does not have a second constraint, you can leave this box blank.

Geometry Type D - Select the geometry type for the other constraint port in the connection. Using the example, you would select X-axis to represent the X-axis of the beam port. If the connection does not have a second constraint, you can leave this box blank.

C and D Alignment - Select how you want the Geometry Type C and Geometry Type C selections to align. Using the example, you would select Aligned / Flush because you want the two X-axes to be coincident.

Related Topics

- *SupportJoints Sheet*, page 51

HgrRefSupportingFilter Sheet

The **HgrRefSupportingFilter** sheet in the **HS_System.xls** workbook is a lookup table for selecting the supporting object filter.

Filter Name - Enter a unique name for the supporting filter. This name must be unique across the entire catalog.

Description - Type a description for the filter.

Connectable Type - Enter the code that specifies if the supporting or supported filter is used to find out the port information for the supporting objects. Valid codes are listed on the **HngSupConnectObj** sheet in the **AllCodeLists.xls** workbook in the **Codelist Number** column.

Discipline Type - Enter the code that represents the type of object that can be supported. Valid codes are listed on the **HngSupDiscipline** sheet in the **AllCodeLists.xls** workbook in the **Codelist Number** column.

Command Type - Enter the code that specifies which Hangers and Supports task command can use the filter. Valid codes are listed on the **HngSupCommand** sheet in the **AllCodeLists.xls** workbook in the **Codelist Number** column.

Defining Interface - Specifies the interface that the filter is to use.

Port Service - Specifies the port service that the software is to use to locate the reference port on the supporting object.

Related Topics

- *Lookup Tables: An Overview*, page 44

HgrDisciplineFilter Sheet

The **HgrDisciplineFilter** sheet in the **HS_System.xls** workbook is a lookup table for selecting the supported object filter.

Filter Name - Enter a unique name for the supported filter. This name must be unique across the entire catalog.

Description - Type a description for the filter.

Symbol Definition - Not used.

Discipline Type - Enter the code that represents the type of object that can be supported. Valid codes are listed on the **HngSupDiscipline** sheet in the **AllCodeLists.xls** workbook in the **Codelist Number** column.

Defining Interface - Specifies the interface that the filter is to use.

Port Service - Specifies the port service that the software is to use to locate the reference port on the supporting object.

ASR - Specifies the assembly selection rule to use when the **Rule** option on the ribbon is active.

Support Prog ID - Specifies the support services program.

Assembly Path - Defines the path in the catalog to the assembly definitions.

Parts Path - Defines the path in the catalog to the hanger and support parts.

Related Topics

- *Lookup Tables: An Overview*, page 44

Index