EPLANT 2023

EPLANT-Piping

3D PIPING AND EQUIPMENT LAYOUT

TECHNICAL MANUAL

Version 2023.0 – August 02th, 2023

RELSOFT S.A.

Corrientes 1455 piso 3 of. 13
C1042AAA Buenos Aires – ARGENTINA
Telefax (5411) 4786 3923 – www.e-eplant.com

ZWSOFT
ZDN member

INDEX

1. INTRODUCTION

This **Technical Manual** contains reference information on the EPLANT-Piping system.

It is intended for people that setup the system and do special tasks, for example, define new parametric piping components or equipments.

It contains also detailed information on tables used by the system.

It is not intended for the regular designer.

References to the User Manual are indicated with UM.

2. SYSTEM REFERENCE TABLES

System reference tables contain general information shared between all projects. Most of this information is stored in binary files with Visual FoxPro format (DBF extension) placed in several system installation directories.

All these tables can be accessed from the REFERENCES menu bar in the database module. Next a detailed description of each table is given.

2.1 PIPING COMPONENT CODES

File name: \PD\STD**COD.DBF**
It contains the definition of all PIPING COMPONENT CODES and some of their characteristics.
All piping components must be defined in this table. Generally this table is modified only when a new component is added or a generic description is changed. It has the following structure:

FIELD	DESCRIPTION
COD	Piping Component Code. This code is used as the generic component code to associate a description, the parametric definition, dimensional tables and to define other important parameters. Length = 3 characters
DES_S	Piping Component Generic Description in Spanish language. This description is used in many reports. Length = up to 25 characters
DES_E	Piping Component Generic Description in English language. This description is used in many reports. Length = up to 25 characters Idem for other languages
PDL	3D Piping Parametric Code. It is the name of the 3D Parametric Definition script file and must be defined also in the PDL.DBF table where several general parameters are associated to it. Length = 3 characters
PID	P&ID Code. It is used to map each Component Code in EPLANT-Piping with the corresponding object in EPLANT-P&ID. This code must be the very same one declared in the CODE column of the project ATR.DBF table associated with the corresponding EPLANT-P&ID project. This mapping is used in Tag verifications. Length = 3 characters
CLS	Component Class, as defined in the \PD\STD\CLS.DBF table Length = 6 characters
ORD	Piping Component main Classification, as defined in the \PD\STD\ORD.DBF table Length = 1 character

PIC	Code used to define the snap mode to a line route, during a component placement. Possible values are: 1 Used for all components that can be placed on an arbitrary position along a line route or to connect them to a connection point of an existing component. It is the common case. 2 Used only for elbows: snapping is done to one of two intersecting line route segments. 3 Used only for tees: snapping is done on the branch side of two intersecting line route segments. C Used only for pipes. Length = 1 character
SNAP	Controls several parameters during component generation: first character: default option during snap selection: C = Connection, L = Line Route. second character: controls options available in the snap window: 0 = only default option, 1 = all. third character: default option in the reference point selection in case of line route snap: 1 = end 1, C = center, 2 = end 2. fourth character: controls options available in the reference point selection window in case of line route snap: 0 = only one option equal to default (the selection window doesn't even appear), 1 = all, 2 = only 1/C, 3 = only 1/2, 4 = only C/2. Length = 4 characters
SUP	Coefficient used for piping component painting area calculation. The painting area is calculated as the external area of a pipe having the same nominal diameter of the component and a length equal to the first parameter of the component. This value is then multiplied by the SUP coefficient. The value is expressed in m^2. Length = 4 numeric digits with a decimal
INSUL	Coefficient used in Insulation length. It is only used as multiplication factor of the real component length in case the corresponding option is set in the Project Setup: General Options 1. Length = 4 numeric digits with two decimals
DIA_N	Secondary Diameter Code. It is used to identify reductions. Possible values are: 1 The component has all diameters equal to the main one. 2 The component is a reduction. Value different than 1 or 2 During the manual loading of a component in the database module, both diameters are not checked. Length = 1 numeric digit.
DIA_1	If is equal to 2, the secondary diameter is copied to the main diameter and the secondary one deleted. The same operation is done on E1 and E2 fields in the project material table. This option is used for some reduction components (half couplings) that are defined only by the small diameter. Length = 1 numeric digit

IMP	Defines the implicit codes used by this component. Possible values are: 0: no implicit code of any type will be generated. 1: it generates the implicit element specified by the end code of each connection point of the component: a welding, a gasket, one set of studs or a clamp. 2: in joints with an end code specifying implicit studs it generates only one set of thru bolts for each component and one gasket for each joint. The stud length is calculated adding to the standard table length the component dimension, rounded to the upper 5 mm. This rounding value is set in the project setup using the STU_RND code. 3: equal to case 1, but it can generate a set of studs only and no gasket. 4: equal to case 2, but generates a set of thru bolts only and no gasket. 5: equal to case 1, but it generates a gasket only and no studs nor bolts. 6: equal to case 2, but it generates a gasket only and no studs nor bolts. 7: equal to case 2, but instead of the STUD.DBF table of the corresponding standard, a special table is used ([component_code]_STUD.DBF) that defines the type, quantity and length of stud for this component. It can have several different types of bolts associated to the same diameter and rating. It is currently used for the Knife Valve (KNF). 8: equal to case 7, but without generating gaskets. Length = 1 numeric digit
ANG	Reference angle, used by some components. For elbows it represents the nominal angle. In Olets it is the angle between the main piping axis and the branch axis. Length = 3 numerical digits without decimals
PC	If 0 the component is never checked against piping specifications (supports, etc.). If greater than 0 the component if searched in the active piping specification class. A value of 2 enables the reading of one dimensional parameter in the specification class, for example for gaskets and nipples. If greater of 2 the first parameter dimension can copied into the secondary diameter field as to be used as discriminant in MTO. This option can be disabled from the project setup. Length = 1 numeric digit
GRA	If 1 the component has a graphic representation and can be used in a 3D model, otherwise it can only be used as a manual component in the database module. Length = 1 character.

2.2 3D PIPING PARAMETRIC CODES

File name: \PD\STD**PDL.DBF**
It contains the definition of all 3D PIPING PARAMETRIC CODES and their characteristics, that is the shapes available to be used as 3D graphically representation for Piping Components. This table is used to associate some general characteristic to each of the PDL (Parametric Definition Language) script files. Generally this table is modified only when a new shape is defined. It has the following structure:

FIELD	**DESCRIPTION**
PDL	3D Piping Parametric Code. It has to be the same code used in the corresponding PDL script file. More than one Piping Component Code can use the same scrip: see column PDL in the COD.DBF table. Length = 3 characters

DES_S	3D Piping Parametric Generic Description in Spanish language. This description is only for clarity, it is not used in any reports. Length = up to 25 characters
DES_E	Piping Component Generic Description in English language. Length = up to 25 characters Idem for other languages
NCP	Number of connection points. Possible values: between 1 and 9. In the case of pipes components, more connection points can be added (up to 98). Length = 1 numeric digit
FACE	If is 0 (most common case) the entering normal to all end faces goes from each connection point through the center. If it is 1, the entering normal is defined by the connection point and an explicit opposite point, defined in the PDL file, and can take an arbitrary orientation. Length = 1 numeric digit
NPAR	It is the total number of Dimensional parameters used in the generation of 3D components. Valid values from 0 to 9. See chapter 5.2. Length = 2 numeric digits
ISO_PDL	If it has zero value the isometric symbol is generated in a static way (it is the more common case), if it has the value 1, the isometric symbol is generated dynamically according to the instructions contained in the [component code]_ISO.PDL placed in the project isometric symbology directory. See the chapter 7 in this manual. Length = 1 numeric digit
TOL_GAP	Used to define tolerance axis in the interference checking. Each character defines an axis and a direction with respect to the local axis of each component. First character = negative X, second character = positive X, third character negative Y, etc. Each character can be 0/1/2. 0: this direction is not increased with the tolerance. 1: this direction is increased with the tolerance. 2: this direction is diminished with the tolerance. It is used to avoid clashing the operator with its valve. Length = 6 characters.
NX	If 1, the component is axisymmetric with respect to the piping axis. Used in view extractions. Length = 1 numeric digit
IMAGE	To store an graphic image of the component parameters

2.3 PIPING COMPONENT ORDER CODES

File name: \PD\STD**ORD.DBF**
Contains the PIPING ORDER CLASSIFICATION CODES. They are used to assign properties to components and in the main sorting order during MTO in isometrics. It has the following format:

FIELD	DESCRIPTION
ORD	Piping Component Order Code. Length = up to 1 characters.
DES_S	Description of the Classification Code in Spanish language. Length = up to 25 characters
DES_E	Description of the Classification Code in English language. Length = up to 25 characters Idem for other languages

2.4 PIPING COMPONENT CLASSIFICATION CODES

File name: \PD\STD**CLS.DBF**
Contains the PIPING COMPONENT CLASSIFICATION CODES. They are used to group components in some commands. It has the following format:

FIELD	DESCRIPTION
CLS	Piping Component Classification Code. Length = up to 6 characters.
DES_S	Description of the Classification Code in Spanish language. Length = up to 25 characters
DES_E	Description of the Classification Code in English language. Length = up to 25 characters Idem for other languages

2.5 MATERIAL CODES

File name: \PD\STD**MAT.DBF**
Contains the description associated to the MATERIAL CODES. The material is referenced through the project using this code only. This table allows translating the material code into a meaningful description.

FIELD	DESCRIPTION
MAT	Material Code. Length = up to 6 characters.
DES_S	Material Description of the Material Code in Spanish language. Length = up to 25 characters
DES_E	Material Description of the Material Code in English language. Length = up to 25 characters Idem for other languages
SP_WEI	Specific Weight in Kg / cm3. Length = numeric 8 dec 3
OBSERV	Comments.

2.6 END CODE

File name: \PD\STD**END.DBF**
END CODE TABLE. Contains the allowed end codes. The blank code is allowed to connect with anything else. Used mainly for pipe ends. The end compatibility is explicitly established by means of the COMP_1/2/3/4 fields.

FIELD	DESCRIPTION
END	End Code. Length = up to 4 characters.
FLG_END	If 0 the end is undefined. If 1 the end is flanged. If 2 the end is top welded. If 3 the end is threaded. If 4 the end is socket welded. Length = 1 numeric digit
COMP_1	Compatible end. Length = up to 4 characters.
COMP_2	Compatible end. Length = up to 4 characters.
COMP_3	Compatible end. Length = up to 4 characters.
COMP_4	Compatible end. Length = up to 4 characters.
IMP_GAS	Generates an implicit Gasket element, using the default project gasket code. System default = GAS.
IMP_STU	Generates a set of implicit Stud or Bolt elements, using the default project stud code. System default = STU.

IMP_WEL	Generates an implicit Welding element, building the welding code with the welding prefix used by the project plus the first two characters of the end code. System default for prefix= W.
IMP_CLU	Generates an implicit Clamp element, using the default project clamp code. System default = CLU.
ISO_SYM	End code used in making the isometric symbol name. Length = up to 4 characters.
CODE	Code to generate an additional Material Codification (not currently used). Length = up to 3 characters.
IMP	Value = 1 The end code in the END field can generate implicit itself using the values associated to the IMP_ fields. Value = 0 The end code in the END field cannot generate implicit itself and must relay on the first compatible end code to do that. Length = 1 numeric digit.
DES_S	End Code Description Spanish language (not currently used). Length = up to 25 characters
DES_E	End Code Description English language (not currently used). Length = up to 25 characters Idem for other languages

2.7 RATING VALUES

File name: \PD\STD**RAT.DBF**
RATING VALUES table. Contains available rating values.

FIELD	DESCRIPTION
RAT	Rating Values. Length = up to 5 characters. Left justified.
CODE	Code to generate an additional Material Codification (not currently used). Length = up to 3 characters.

2.8 SCHEDULE VALUES

File name: \PD\STD**SCH.DBF**
SCHEDULE VALUES table. Contains available schedule values.

FIELD	DESCRIPTION
SCH	Schedule or Thickness. Length = up to 6 characters. Left justified.
CODE	Code to generate an additional Material Codification (not currently used). Length = up to 3 characters.

2.9 MESSAGES USED IN PARAMETRIC FILES

File name: \PD\STD**USR_MSG.DBF**
MESSAGES used in PARAMETRIC FILES. Contains text used as prompt generated from the PDL files. They used to customize data input in different languages. New variables can be defined.

FIELD	DESCRIPTION
MSG	Message Variable index. Length = 3 numerical digits.
DES_S	Text in Spanish language associated to the message variable. Length = up to 25 characters
DES_E	Text in English language associated to the message variable. Length = up to 25 characters Idem for other languages

2.10 PARAMETRIC EQUIPMENT TYPE DEFINITION TABLE

File name: \PD\EDL**EQU_DES.DBF**
This table defines the equipment types and it is used for the parametric equipment definition command.

FIELD	DESCRIPTION
CLASS	Equipment Type code. Used in the selection popup menu during the generation. Length = up to 6 characters.
CODE	Equipment specific code. It is the file name with the parametric definition with EDL extension. Length = up to 6 characters.
DES_S	Equipment Description in Spanish language. Length = up to 25 characters.
DES_E	Equipment Description in English language. Length = up to 25 characters. Idem for other languages
N_NOZ	Equipment total nozzle number. Up to 4.
N_PAR	Number of parameter used to define the equipment geometry. Up to 9.

2.11 PARAMETRIC EQUIPMENT DIMENSION TABLE

File name: \PD\EDL**EQU_DIM.DBF**
This table contains all equipment models and dimensions that can be selected during the generation of parametric equipment.

FIELD	DESCRIPTION
CODE	Equipment specific code. It is the file name with the parametric definition with EDL extension. It must be defined in the \PD\EDL**EQU_DIM.DBF** table. Length = up to 6 characters.
DES	Description of the specific Equipment model. Length = up to 25 characters.
DIA_1	Diameter in inches of Nozzle 1. Length = 6 characters.
END_1	End code of Nozzle 1. Length = 4 characters.
SER_1	Rating of Nozzle 1. Length = 5 characters.
SCH_1	Schedule of Nozzle 1. Length = 6 characters.
LEN_1	Length of Nozzle 1. Length = 6 numerical digits.
DIA_2	Diameter in inches of Nozzle 1. Length = 6 characters. The same schema is repeated up to nozzle 4.
P1	Parameter 1 value. Length = 6 numerical digits.
P2	Parameter 1 value. Length = 6 numerical digits. The same schema up to parameter 9.

2.12 NOMINAL DIAMETER TABLE

File Name: \PD\STD\ANSI**PIP.DBF**
Diameter definition table. Contains the nominal diameters available in the corresponding standard

FIELD	DESCRIPTION
DIAM	Piping Nominal Diameter. It needs the same format used in the dimensional tables of the same standard. It is also used as Line Nominal Diameter. Length = 6 characters
VAL	External Diameter in mm. Length = 10 numerical digits with 2 decimals
DD	Nominal Diameter Code, used for diameter sorting. Length = 6 characters
LINE_N	With 1 the diameter can be used as piping line nominal diameter, with 0 only can be used as stud diameter. Length = 1 numerical digit

2.13 PIPING COMPONENT DIMENSION TABLES

File names: \PD\STD\ANSI\
[component_generic_code][parameter_number][end_code].DBF

These tables, located in the standard directories, contain piping component dimensions. Each directory nested into the system STD directory is considered a different standard.
These tables have different formats, depending on the component type.
Three parts compose the name of these files:
[component_generic_code]: it is the component generic code (COD).
[parameter_number]: it is the parameter number stored in the table.
[end_code]: two or more characters representing the end code 1 followed by the end code 2 if differente. This rule can be changed in the parametric definition file (PDL).
All these tables have the first field named DIAM, with the nominal diameter and the value corresponding to the parameter stored in the field named VAL.
The following fields can also be used to specify other entry parameters: RAT (for rating), SCH (for schedule), DIAM2 (for the secondary diameter). These fields can be used in the same table as needed.

In case of valve operators only, the first part of the table name is the full component code (up to 6 characters) as read from the OPE field in the piping class or as could have been manually entered.

The simpliest format is:

One entry Table: value is associated to the Nominal Diameter only

FIELD	DESCRIPTION
DIAM	Nominal Diameter. Length = up to 6 characters.
VAL	Contains the component dimension, depends from the diameter only. Length = 10 numeric digit with two decimals.

The most complicated table could be: value depends of Nominal Diameter, Rating, Schedule, Secondary Diameter

FIELD	DESCRIPTION
DIAM	Nominal Diameter. Length = up to 6 characters.
RAT	Contains the Rating value. Length = up to 5 characters.
SCH	Contains the Schedule. Length = up to 6 characters.
DIAM2	Contains the Secondary Diameter. Length = up to 6 characters.
VAL	Contains the component dimension, depends from all the parameters defines in the preceding columns. Length = 10 numeric digit with two decimals.

The same combination of entry parameters can be repeted an arbitrary number of time to associate a different dimension. In this particular case, during the component placement a prompt will show the available values. This option is used, for example, to associated specified lengths to flanged pipes (glass or cast-iron pipes).

2.14 FLANGE DIAMETER TABLE

File Name: \PD\STD\ANSI**FLGD.DBF**
Flange Diameter definition table. Contains the Flange Plate Diameter and is function of the nominal piping diameter and rating

FIELD	DESCRIPTION
DIAM	Nominal Diameter. Length = up to 6 characters
RAT	Contains Flange Rating Length = up to 5 characters.
VAL	Contains the Flange Plate Diameter in mm. Length = 10 numeric digit with two decimals.

2.15 FLANGE THICKNESS TABLE

File Name: \PD\STD\ANSI**FLGT[End_Code].DBF**
Flange Thickness definition table. Contains the Flange Plate Thikness and is function of the nominal piping diameter, rating and of the end code (the latter is included in the file name).

CAMPO	DESCRIPTION
DIAM	Nominal Diameter. Length = up to 6 characters
RAT	Contains Flange Rating Length = up to 5 characters.
SER_n	Contains the Flange Plate Thickness in mm. Length = 10 numeric digit with two decimals.

2.16 STUDS AND BOLTS

File name: \PD\STD\ANSI**STUD.DBF**
STUDS and BOLTS table. Contains studs and bolts diameters and lengths by the nominal diameter and rating for the Ansi standard. Other standards use the same file name placed in their corresponding directories.

FIELD	DESCRIPTION
RAT	Rating. Length = 5 characters
DIAM	Nominal Diameter. Length = 6 characters
N_STU	Studs and Bolts quantity in a set used for a flanged joint. Length = 2 numeric digits
D_STU_IN	Studs and Bolts diameters in inches. Length = 6 characters
D_STU_MM	Studs and Bolts in mm. Not currently used. Length = 6 characters
L_STU_RF	Stud Length for the End Code Raised Face. Length = 3 numeric digits
L_STU_FF	Stud Length for the End Code Flat Face. Length = 3 numeric digits
L_STU_RJ	Stud Length for the End Code Ring Joint. Length = 3 numeric digits
L_BLT_RF	Bolt Length for the End Code Raised Face. Length = 3 numeric digits
L_BLT_FF	Bolt Length for the End Code Flat Face. Length = 3 numeric digits
L_BLT_RJ	Bolt Length for the End Code Ring Joint. Length = 3 numeric digits

Components that have the IMP field = 7 or 8 in the COD.DBF table do not use the general stud table and use instead a specific table for each component, which name is:
File name: \PD\STD\ANSI**[component_generic_code]_STUB.DBF**
For example the Knife valve hase this definition. The structure of these tables is the same that the generic tables except for the last field named COD (Character 6). This field is used to specify the implicit component code that has to be generated (STU or BLT). For the same nominal diameter and rating more than one component can be defined, that is an arbitrary combination of stud and bolts can be defined for a given nominal diameter.

2.17 PIPING COMPONENT WEIGHT TABLES

File names:
\PD\STD\ANSI\WEI**[component_generic_code][end_code].DBF**

These tables are located in the WEI directory inside the corresponding standard directory. They contain the component weight in Kg. They are binary files with dBASE format.
The weight depends always from the nominal diameter and either the rating or the schedule of the component. Field names follow the same rules that dimensional tables do. In case of thickness, apart to add the corresponding field in the required table, the same field must be added (if not already in) in the master WEI_SCH.DBF, that is automatically reconstructed each time a weight table is modified. A special case is the STUBW.DBF that contains the weight of studs and bolts expressed in Kg per mm of length.

In the case of External Code use, it is possible to set the project to take the weight directly from the External Code definition table. Is this the case when using the PUMA system to generate piping classes.

3. PROJECT REFERENCE TABLES

Project reference tables contains project specific information. They are also binary files with dBASE format, placed in the project DBF directory. Their names use the project code as a prefix.

Opening a new project, and any time an existing project is opened, each of these tables is checked for existence. If not found they are automatically copied from system masters.

Access to these tables is done from the database module. A detailed description of every table follows below.

3.1 PROJECT SETUP

File name: \[project_code]\DBF**[project_code]SET.DBF**
Contains configuration information of the project. See 5.3 of UM for details.

3.2 PIPING SPECIFICATIONS

File name: \[project_code]\DBF**[project_code]SP.DBF**
Contains the PROJECT PIPING SPECIFICATIONS. See 4.8 of UM for details. The table format is:

FIELD	DESCRIPTION
PCLA	Piping Class. Length = up to 9 characters. Left justified.
COD	Generic Piping Component code. The first three characters are checked with the COD file content in the \PD\STD\COD.DBF table. Longer codes can be used, for example CRE1. This allows assigning different parameters to the same type of component and in the same diameter range. This possibility is compatible with the naming conventions for dimensional tables. Length = up to 6 characters. Left justified.
OPE	Only valves use it. It represents the operator code associated to the valve specified in the COD field. If this field is void, the valve will be generated without operator. The operator code uses the first three characters (they are validated against the COD.DBF table). Additonal characters, if any, are used in the creation of the names used in dimensional tables only, whose names begin with the content of this OPE field followed by the parameter number and the end codes if required. Length = 6 characters.
D1A	Main diameter lowest range, in inches. It is checked against the DIAM field in the \PD\STD\ANSI\PIP.DBF table, for the Ansi standard. Length = 6 characters.
DD	Diameter code corresponding to the D1A field content. It is automatically generated. Length = 3 characters.
D1B	Main diameter highest range, in inches. It is checked as the D1A field. Length = 6 characters.
D2A	Secondary diameter lowest range, in inches. It is checked as the D1A field. Length = 6 characters.

D2B	Secondary diameter highest range, in inches. It is checked as the D1A field. Length = 6 characters.
RAT	Rating of the main diameter. Possible rating values are in \PD\STD\SER.DBF table. Length = 5 characters.
RAT2	Rating of the secondary diameter. Length = 5 characters.
SCH	Schedule of the main diameter. Possible rating values are in \PD\STD\SCH.DBF table. It can be also used to define thickness. In case of thickness, the value must contain a decimal point, to discriminate it from a schedule value. In the weight computation the thickness is considered in inches or mm depending on the project setup. Length = 6 characters.
SCH2	Schedule of the secondary diameter. Length = 6 characters.
E1	End code used for connection points 1 and 3. Possible values are contained in the \PD\STD\END.DBF table. This code is also used the dimensional and weight tables names. The position of connection points varies with the component. Length = up to 4 characters.
E2	End code used for connection points 2 and 4. Idem above. If this field is not specified, the system assumes it equal to E1. Length = up to 4 characters.
MAT	Material Code. Is checked with the MAT field in the \PD\STD\MAT.DBF table. If the code placed into the class doesn't exists, the material description won't appear. Length = up to 6 characters.
CODA	Additional Code. It is checked with the CODA field of the [project]CD.DBF file. If this text is defined in that table and it has a text associated in the corresponding Memo field, this text will appear in MTOs. This code has two main purposes: to complete the generic description associated with the COD code and to associate a description arbitrary large in material requisitions. Length = up to 16 characters.
THCK	Thickness/Dimension: only those components that have the PC field in the COD.DBF table = to 2 can read this field. In case of Gaskets, this field is associated with the gasket thickness and for all other components to the dimensional parameter specified by the reading code PCL in the corresponding PDL file. Length = Numeric of 8 with 2 decimals.
GAS	If this field is left blank and the component has flanged joints and these joints do generate implicit elements, a gasket with the GAS code is automatically generated. This code must be in the class definition, to be able to assign MAT and CODA if needed. Length = up to 6 characters.
STU	If this field is left blank and the component has flanged joints and these joints do generate implicit elements, a stud with the STU code is automatically generated. This code must be in the class definition, to be able to assign MAT and CODA if needed. If a different stud is needed, for example bolts (code BLT), its corresponding code must be loaded in this field. Length = up to 6 characters.
STD	If this field is left blank (it is the most common case) the system assumes that the component uses the dimensional standard defined in the project setup. If a name is found, this is interpreted as the standard name. In this case the corresponding directory must be already present to the \PD\STD directory. Length = up to 10 characters.
SPOOL	Used to load the Spool code in case this option is enabled in the Project setup. Length = up to 3 characters

File name: \[project_code]\DBF**[project_code]BRA.DBF**
Contains the BRANCH TABLES ASSOCIATED TO EACH PROJECT PIPING SPECIFICATION. See 4.8 of UM for details. The table format is:

FIELD	DESCRIPTION
PCLA	Piping Class. Length = up to 9 characters. Left justified.
DIAM	Nominal Diameter. Length = up to 6 characters.
DD	Diameter code corresponding to the DIAM field content. It is automatically generated. Length = 3 characters
DIAM2	Secondary Diameter. Length = up to 6 characters.
DD2	Diameter code corresponding to the DIAM2 field content. It is automatically generated. Length = 3 characters.
COD	Component Code used for the combination of DIAM and DIAM2. If more than one component is needed separate them with a +. If more than one alternative is available, separate them with ; Length = 3 characters.
STD	If this field is left blank (it is the most common case) the system assumes that the component uses the dimensional standard defined in the project setup. If a name is found, this is interpreted as the standard name. In this case the corresponding directory must be already present to the \PD\STD directory. Length = up to 10 characters.

File name: \[project_code]\DBF**[project_code]SPD.DBF**
Contains the DESCRIPTION ASSOCIATED TO EACH PROJECT PIPING SPECIFICATION. See 4.8 of UM for details. The table format is:

FIELD	DESCRIPTION
PCLA	Piping Class. Length = up to 9 characters. Left justified.
DES_S	Piping Class Description in Spanish language. Length = up to 25 characters.
DES_E	Piping Class Description in English language. Length = up to 25 characters. Idem for other languages

3.3INSULATION SPECIFICATIONS

File name: \[project_code]\DBF**[project_code]IN.DBF**
Contains the Project INSULATION SPECIFICATIONS. See 4.8.3 in UM to a detailed discussion of the reading of this table. The table format is:

FIELD	DESCRIPTION
ICLA	Insulation class, any name up to six characters. Identifies the insulation material. If this code is loaded in the Additional Codes table, a description can be loaded into the memo file and this description can appear in the material requisitions. Length = up to 6 characters.
DIAM	Nominal piping diameter. Ranges are not accepted, only specific values. Length = 6 characters.
DD	Diameter code corresponding to the D1A field content. It is automatically generated. Length = 3 characters.
I_TH	Specifies the insulation thickness corresponding to the diameter stored in the DIAM field. This parameter is used in Material Take Offs. Length = up to 6 characters.
I_MM	Contains the insulation thickness in mm corresponding to the diameter stored in the DIAM field. It is used in Clash Detection. Length = 6 numeric digits
MAT_INS	Contains the Material Code associated to the insulation. Length = up to 6 characters.
CODA_INS	Contains the Additional Code associated to the insulation. Length = up to 16 characters.
LINING	Lining Code. If it is equal to 1, Insulation Lining will be generated with the same quantity as the insulation. Length = 1 numeric digit.
MAT_LIN	Contains the Material Code associated to the Insulation Lining. Length = up to 6 characters.
CODA_LIN	Contains the Additional Code associated to the Insulation Lining. Length = up to 16 characters.

3.4ADDITIONAL PIPING CODES

File name: \[project_code]\DBF**[project_code]CD.DBF**
Contains the Project ADDITIONAL PIPING CODES, used to associate a text description of arbitrary length in material requisitions. The table format is:

FIELD	DESCRIPTION
CODA	Additional Piping Code. Length = up to 6 characters.
DESM_S	This is a MEMO type field: it can be used to store an arbitrary long description text in Spanish language.
DES_S	It is used to load a short description in Spanish language, of the full text stored in the previous field, as an index when browsing through all the codes. It can be included in the isometric MTO. Length = 60 characters.
DESM_E	Equivalent to DESM_S for the English language. Idem for other languages.
DES_E	Equivalent to DES_S for the English language. Idem for other languages. Length = 60 characters.

3.5PIPING MATERIAL EXTERNAL CODE

File name: \[project_code]\DBF**[project_code]CDE.DBF**
Contains the definition of the Project PIPING MATERIAL EXTERNAL CODE. This file is created and used only if the corresponding option in the project setup is enabled.
The operation of code assigment is automatically executed during the material update in the data base module, during material manual load in the data base module and during isometric MTO.
If the default structure is not compatible with the project requirements, the user can modify its structure, using the UTILITIES option. In this case, after the structure modification, the [project_code]CDE.IDX file must be deleted and the project opened again to automatically rebuild the index with the new structure. See 4.8.6 in UM for details.
The content of this table can be automatically generated using the suitable otion in the UTILITIES menu in the data base module, to be sure that all materials defined in the piping specs or all material generated by the project are represented in the CDE table.

This table has also the EXT_WEI field to store weight and EXT_DES to store a short description corresponding to each external code. These two fields are intended to be automatically loaded by an external program, such as Puma. They are available as selectable fields in the isometric MTO. Avoid to use them unless they are automatically loaded.

3.6PIPING MATERIAL ALTERNATE CODE

File name: \[project_code]\DBF**[project_code]CDA.DBF**
Similar to the External code, but using another table. Without EXT_WEI and EXT_DES fields.

3.7MATERIAL GROUPING CRITERIA

File name: \[project_code]\DBF**[project_code]REC.DBF**
This table specified the CRITERIA used to group material in the project requirements. Is format is:

FIELD	DESCRIPTION
ORDE	Group Code. The first character is the same as the field ORD in the \PD\STD\COD.DBF table. This field must be always filled with a code. Never leave it blank. Each requisition corresponds to a different group code. Length = 2 characters.
COD	Can contain a piping component code, as defined in the \PD\STD\COD.DBF table. It can be void. Length = 3 characters.
MAT	Can contain a material code, as defined in the \PD\STD\MAT.DBF table. It can be void. Length = 6 characters.
D1	Can contain a diameter, representing a lower range, as defined in the \PD\STD\ANSI\PIP.DBF table. It can be void. Length = 6 characters.
D2	Can contain a diameter, representing an upper range, as defined in the \PD\STD\ANSI\PIP.DBF table. It can be void. Length = 6 characters.
SURPLUS	If this field is different than zero, it is interpreted as the surplus (in %) to be added to the computed quantity to obtain the purchase quantity for this ORDE code. It takes precedence with respect the surplus value defined in the Requisition Title table. Length = 4 numeric digits with a decimal.

Data in this table, for a given record, are interpreted as follows:

- If both the COD and MAT fields have codes in them, the corresponding group code is assigned only to those components having both the specified codes.
- If only the COD field is filled, the group code will be assigned to those components having that COD and with any material code.
- If only the MAT code is filled, the group code will be assigned to those components having that MAT code and with any component code.
- If different group codes must be assigned for different diameter ranges, use the D1 and D2 fields in the same way. If these two fields are left void, diameters are not used to assign group codes.

This table is analyzed by the system in its sorted version (key = ORDE + COD + MAT) so if the same component codes or material have assigned different group codes, only the first one will be used.
If a piping component is not found in this table, it will receive a default group code with its first character equal to the ORD code (\PD\STD\COD.DBF table) followed by a zero.

3.8TITLES ASSOCIATED TO EACH GROUP CODE

File name: \[project_code]\DBF**[project_code]RED.DBF**
This table complements the previous one. It defines the TITLES ASSOCIATED TO EACH GROUP CODE. It has the following format:

FIELD	DESCRIPTION
ORDE	Group Code. Length = 2 characters.
DES_S	Associated Description in Spanish language. This text is used as Requisition Titles in reports. Length = 50 characters.
DES_E	Associated Description in English language. This text is used as Requisition Titles in reports. Idem for other languages Length = 50 characters.
N_DOC	To store the Document Number associated to each requisition. This text will automatically appear in the requisition report as reference. It is associated to the global variable **NDOC**. Length = 20 characters.
SURPLUS	During the generation of a new material total, the purchase quantity can be increased by the amount specified here for all the items belonging to this group code. The value is taken as %. Length = 3 numeric digits.

3.9SCHEDULING PHASES

File name: \[project_code]\DBF**[project_code]DAT.DBF**
Defines the SCHEDULING PHASES associated to Equipments and Lines. It has the following format

FIELD	DESCRIPTION
PHASE	Phase Name. Each phase defined in this table will be displayed as two fields in the Equipment and Line Schedule list, depending on the FL field value. Length = 16 characters.
COLOR_1	AutoCAD® Color Number associated to the phase when it is completed by the Schedule Date. Length = 3 numeric digits.
COLOR_2	AutoCAD® Color Number associated to a delayed phase when it is not completed by the Real Date. Length = 3 numeric digits.
FL	It is E for Equipment phases, L for Line phases. It is automatically loaded by the system. Length = 1 character.

3.10 EXTERNAL FILES LINK

File name: \[project_code]\DBF**[project_code]LNK.DBF**
Contains the names of external files with documents related to Equipments, Lines and components with Tag inside EPLANT models. It is used if enabled in the project setup.

FIELD	DESCRIPTION
TAG_EPLANT	It is the parameter contained inside EPLANT models used to associate external documents. For Equipment is the Equipment Name, for Lines is the Line Number and for components is the Tag value. More than one record can be assigned to the same parameter, to link multiple documents. Length = 25 characters.
FILE	File associated to the EPLANT element identified in the TAG_EPLANT field. It can be a file with any format, the name and extension must be specified. If the name does not specify the path, the file is searched for in the project LINK directory, otherwise it can be place anywhere. Length = 100 characters.

3.11 REFERENCE POINTS

File name: \[project_code]\DBF**[project_code]RPT.DBF**
Contains the Reference Points Coordinates that can be imported during equipment generation as the equipment center.

FIELD	DESCRIPTION
DES	Point Description. It can be the equipment name or any identification. Length = 25 characters.
X	X coordinate. It is regarded expressed in current UCS while generating the equipment Length = numeric 14 with 2 decimals.
Y	Y coordinate. It is regarded expressed in current UCS while generating the equipment Length = numeric 14 with 2 decimals.
Z	Z coordinate. It is regarded expressed in current UCS while generating the equipment Length = numeric 14 with 2 decimals.
ANG	Equipment insertion angle, counterclockwise from the X axis. Length = numeric 8 with 2 decimals.

3.12 PIPING COMPONENT COST

File name: \[project_code]\DBF**[project_code]CST.DBF**
Contains Piping Component Cost. This table is related to all other material tables and allow to associate the cost of each component in each Material Listing. The same rules used in Piping Specification apply. In any case, the **Table Cost Update** option in the data Base module allows to automatically load this table.

FIELD	DESCRIPTION
COD	Generic Piping Component code. The first three characters are checked with the COD file content in the \PD\STD\COD.DBF table. Longer codes can be used, for example CRE1. This allows assigning different parameters to the same type of component and in the same diameter range. This possibility is compatible with the naming conventions for dimensional tables. Length = up to 6 characters. Left justified.
D1	Main diameter in inches. It is checked against the DIAM field in the \PD\STD\ANSI\PIP.DBF table, for the Ansi standard. Length = 6 characters.
DD1	Diameter code corresponding to the D1 field content. It is automatically generated. Length = 3 characters.
D2	Secondary diameter, in inches. It is checked as the D1 field. Length = 6 characters.
RAT	Rating of the main diameter. Possible rating values are in \PD\STD\SER.DBF table. Length = 5 characters.
RAT2	Rating of the secondary diameter. Length = 5 characters.
SCH	Schedule of the main diameter. Possible rating values are in \PD\STD\SCH.DBF table. It can be also used to define thickness. In case of thickness, the value must contain a decimal point, to discriminate it from a schedule value. In the weight computation the thickness is considered in inches or mm depending on the project setup. Length = 6 characters.
SCH2	Schedule of the secondary diameter. Length = 6 characters.
E1	End code used for connection points 1 and 3. Possible values are contained in the \PD\STD\END.DBF table. This code is also used the dimensional and weight tables names. The position of connection points varies with the component. Length = up to 4 characters.
E2	End code used for connection points 2 and 4. Idem above. If this field is not specified, the system assumes it equal to E1. Length = up to 4 characters.
MAT	Material Code. Is checked with the MAT field in the \PD\STD\MAT.DBF table. If the code placed into the class doesn't exists, the material description won't appear. Length = up to 6 characters.
CODA	Additional Code. It is checked with the CODA field of the [project]CD.DBF file. If this text is defined in that table and it has a text associated in the corresponding Memo field, this text will appear in MTOs. This code has two main purposes: to complete the generic description associated with the COD code and to associate a description arbitrary large in material requisitions. Length = up to 16 characters.
STD	Component dimensional standard. Length = up to 10 characters.

NAME	Component dimensions in the case of a manually loaded component only. Generally this field is left blank. Length = up to 20 characters.
COST	Component Cost. For Pipes it represents the Cost per meter, for the rest of the components is the Unitary Cost. Length = numeric of 12 with 2 decimals.

3.13 FLUID AND COLOR TABLES

File name: [project_code]\DBF**[project_code]FLU.DBF**
Contains AutoCAD® Color Codes associated to each Fluid Code. It is used in the case the project setup is configured to use a Fluid dependent color for piping components.

FIELD	DESCRIPTION
FLU	Fluid Code, in the same way it appears in Line Numbers. Length = 10 characters.
COLOR	AutoCAD® Color Code. Length = numeric of 3.
DES_S	Fluid Description in Spanish Language. For description purposes, not used by the system. Length = 60 characters.
DES_E	Fluid Description in English Language. For description purposes, not used by the system Length = 60 characters.

3.14 PROJECT END CODES

File name: [project_code]\DBF**[project_code]E1.DBF**
Contains the Project Descriptive Code associated to each System End Code. It can be used in any reports. System codes must be used in piping specifications. From the data base module this E1.DBF table can be changed. A copy of this table with the E2.DBF name is automatically created for internal purposes.

FIELD	DESCRIPTION
END	System End Code, such as in the system table \eplant\pd\std\END.DBF. Length = 4 characters.
DES	Project Descriptive Code. Length = 10 characters.

3.15 EQUIPMENT STATUS

File name: \[project_code]\DBF**[project_code]STE.DBF**
Contains Status codes that can be assigned to each equipment. Equipment status is associated to each equipment using the corresponding command. See Equipment menu.

FIELD	DESCRIPTION
STATUS	Equipment Status Code. Length = 6 characters.
DES_E	Description in English Language. Length = 40 characters.
DES_S	Description in Spanish Language. Idem para los otros idiomas. Length = 40 characters.

3.16 LINE STATUS

File name: \[project_code]\DBF**[project_code]STL.DBF**
Contains Status codes that can be assigned to each line. Line status is associated to each line using the corresponding command. See Line Utilities menu.

FIELD	DESCRIPTION
STATUS	Line Status Code. Length = 6 characters.
DES_E	Description in English Language. Length = 40 characters.
DES_S	Description in Spanish Language. Idem para los otros idiomas. Length = 40 characters.

4. PROJECT MATERIAL TABLES

The information extracted from all project 3D models is stored in tables in the project DBF directory. The user can ignore the real distribution of this information. In any case, a summary of those tables is given.

4.1PROJECT COMPONENTS

File name: \[project_code]\DBF**[project_code]PIP.DBF**
Contains all piping components extracted from 3D models of the project. It contains also all the implicit elements generated during the update processing. It uses three indexes with suffixes PIP, PIL and PIF. It is automatically updated any time the project is opened from the database module. Its update status can be inspected from the option 3D MODELS in the main database menu bar.

4.2EQUIPMENTS

The following tables are used:

File name: \[project_code]\DBF**[project_code]EQU.DBF**
Contains all equipment names defined in the project 3D models. It is automatically updated as the PIP table.

File name: \[project_code]\DBF**[project_code]DLE.DBF**
Contains all equipment names defined in the project 3D models. It is used to load Schedule and Real Dates for each equipment and phase. It is automatically updated as the PIP table.

4.3EQUIPMENTS NOZZLES

The following table is used:

File name: \[project_code]\DBF**[project_code]NOZ.DBF**
Contains all nozzles defined in the project 3D models, with their characteristics and the equipment where they are placed.

4.4LINES

The following tables are used:

File name: \[project_code]\DBF**[project_code]LIN.DBF**
Contains all line names defined in the project 3D models. It is automatically updated as the PIP table.

File name: \[project_code]\DBF**[project_code]DLE.DBF**
Contains all line names defined in the project 3D models. It is used to load Schedule and Real Dates for each line and phase. It is automatically updated as the PIP table.

4.5PROJECT MATERIAL TOTALS FOR PURCHASE

File name: \[project_code]\DBF**[project_code]Tn.DBF**
Contains project piping material total quantities, used for material requisition generation. The number n in the file name is the revision total number (from 0 to 99) and it is automatically assigned by the system.

4.6PROJECT MATERIAL TOTALS FOR SPOOLS

File name: \[project_code]\DBF**[project_code]Sn.DBF**
Contains project piping material total quantities, separated by line and spool code. The number n in the file name is the revision total number (from 00 to 99) and it is automatically assigned by the system. See 4.7.15 and 5.8 in UM.

4.7PROJECT MATERIAL TOTALS FOR JOINTS

File name: \[project_code]\DBF**[project_code]Ln.DBF**
File name: \[project_code]\DBF**[project_code]Un.DBF**
These files contain information about piping components and joint codes. They can be used to track fabrication and welding certification. The number n in the file name is the revision total number (from 00 to 99) and it is automatically assigned by the system. See 4.7.17 and 5.8 in UM.

5. 3D COMPONENT PARAMETRIC GENERATION

5.1INTRODUCTION

All piping components have a parametric definition; that is, the shape depends of parameters. Parameter values are stored in dimensional tables, placed in the standard directory or can be input manually during the component generation.

The parametric definitions use the EPLANT PDL language. To each component defined in the \PD\STD\COD.DBF table there must be a corresponding file with the name specified in the PDL column of the same table and PDL extension, available in the \PD\PDL directory, or the one specified by the Project Setup.
Each PDL name must be defined also in the \PD\STD\PDL.DBF table where specific parameter values are defined. The same PDL definition can be used for several Component Codes.

The system is comes with an extensive library of piping components. PDL files are text files that the user can modify.

The parametric definition is automatically activated when placing a component with the command **(cmp "code" 0)**.

If the function second parameter is zero, the dimensional tables are automatically read, otherwise each dimension will be prompted from the keyboard. The maximum number of parameter per component is nine.
The dimensional table name associated to each parameter is defined with the following default rule (it can modified in the pdl file, see below for details):

- firsts three characters: component code
- fourth character: 1/2/3/4/5/6/7/8/9 depending on the parameter number stored inside the table
- following characters: end code of connection point 1.

An example can be: GAT1RF.DBF.

5.2PDL FILE SYNTAX

PDL files are text files with sentences for the parametric generation of piping components. They can be modified with any text editor.
Sentences must be written in the PDL language. The system verifies the syntax when reading the file, warning any error found.

The firsts two lines are a required header, each line begins with a key word, followed by a variable part. For example, the file for the GAT code is named GAT.PDL and contains the following:

GAT Gate Valve
TH=NO

First line: GAT is repeated as the first line and next, separated by a blank space, a descriptive text can be placed, usually to identify the component or the way it is rendered. The number of parameters used by the component is the value of the field NPAR in the PDL.DBF system

component definition table. It can be any number between 0 and 9. Parameters are referenced in the PDL file using fixed name variables: P1, P2, ... up to P9
Second line: TH=NO specifies that the piping thickness will not be read. The sentence TH=YES activates the reading of the piping thickness from the \PD\STD\ANSI\PIPT.DBF table. In this case the variable TH will be assigned the read value, otherwise it will be set to 0.

If the component is flanged, flanges tables are automatically read. These tables have fixed names and are placed in the corresponding directory standard. For ANSI standard their names are:
\PD\STD\ANSI\FLGD.DBF diameter of the flange plate.
\PD\STD\ANSI\FLGTRF.DBF thickness of the flange plate for RF end code.
\PD\STD\ANSI\FLGTFF.DBF thickness of the flange plate for FF end code and so on.

The following variables are then assigned:

FT1 plate flange thickness, corresponding to the nominal diameter and component rating
FR1 plate flange radius, corresponding to the nominal diameter and component rating
FT2 plate flange thickness, corresponding to the secondary diameter and component rating
FR2 plate flange radius, corresponding to the secondary diameter and component rating
Etc. for the remaining connection points up to FT9 and FR9.

> **Note**. The legacy (previous to version 2005.0) syntax:
> **FT=FLGT**
> **FR=FLGD**
> It is no longer honoured. These sentences can remain in the pdl file, but will be ignored. The same for the number of parameters that was formerly read from the first line of the pdl file.

If the component is a reduction, FT2 and FR2 will contain the values corresponding to the secondary diameter, otherwise all flange thickness will be equal to FT1 and all flange radius to FR1.
In the case of main diameter the piping class field RAT is used to read flange tables. In the case of other diameters the field RAT2 is used instead, if defined, otherwise the RAT value is used also in those cases.

If the component is not flanged FTi and FRi are set to zero.

If the component ends are welded or threaded the system tries to read penetration tables with names:
\PD\STD\ANSI\SC.DBF penetration for threaded end SC.
\PD\STD\ANSI\SW.DBF penetration for socket weld end SW, etc.

and the following variables receive their values:
PE1 = penetration end 1
PE2 = idem 2
Etc. up to PR9.

If these tables are not found no error message is issued and these variables are set to zero value.

After these firsts two sentences as many as n lines will follow, where n is the number of parameters. If no parameters are read, these sentences will not appear.
These sentences has the following minimum syntax, for example for the parameter 1:

P1

That is, the letter P followed by the number of the parameter, in this case 1. During the placement of the component, this variable will receive the parameter value read from the corresponding dimensional table.

> **Note**. The legacy (previous to version 2011.0) syntax that specified the format of the dimensional tables as well as other key words not specified below is ignored:
> **P1 BW=VAL RF=SER FF=SER SC=SER SW=SER MSG=n**

Alter the parameter name it is posible to place other key words (each separated by a single space) to modify the default behaviour. The supported syntax is the following:

The key word **MSG=** followed by a number, associates a text used as prompt in case of manual input of this parameter value. The reference number and the associated text must be defined in the \PD\STD\USR_MSG.DBF table.

The key word **NOEND** specifies that end codes will not be used to generate the table name to store the P1 parameter.
An example of this syntax can be found in the WHE.PDL file, used by the valve manual operator. The corresponding dimension table is in this case the \PD\STD\ANSI\WHE1.DBF table.

The key word **MAN** is used to specify that this parameter will be always loaded manually.

The key word **D1= i** specifies that the number i is assigned to the main diameter (default 1). For example D1=2 means that the first reading diameter (column DIAM in the dimension tables) is set to the secondary diameter (and not to the main diameter as default).

The key word **D2= j** specifies that the number j is assigned to the second reading parameter (default 2). For example D2=3 means that the reading secondary diameter (column DIAM2 in the dimension tables) is the diameter corresponding to the third connection point (and not to the deafault secondary diameter).

The key word **E1= k** specifies that the number k is assigned to the end of the first reading diameter (default 1).

The key word **E2= m** specifies that the number m is assigned to the end of the second reading diameter (default 2).

The key word **RAT=2** specifies that the main rating is changed with the secondary one during dimensional table reading. If the secondary rating is not defined this sentence is ignored.

The key word **SCH=2** specifies that the main schedule is changed with the secondary one during dimensional table reading. If the secondary schedule is not defined this sentence is ignored.

The key word **INTERNAL=LEN** is used in Pipes and automatically assigns the LEN length parameter (internally calculated) to the P1 parameter

The key word **PCL** specifies that the P1 parameter is read from the THCK (Thickness) field in the piping specification table. To enable this possibility the corresponding component must have the value 2 in the PC field of the PDL.DBF table. This is the default value for Nipples.

The key word **PAR=0** allows to input the parameter with a zero value. It is used in case the parameter represents a rotation angle.

After the parameter definitions, an arbitrary number of sentences follow, to define the shape and dimensions of the component.
Lines with a preceding * symbol are skipped as commentary.

Apart from the header, PDL sentences fit in the following four types:

- Key words: set the start and end of a group of sentences.
- Value setting to a variable: variables have fixed names.
- Conditional sentences. They control the execution of other sentences.
- Generation of a graphic element with characteristics defined by variables.

5.3KEY WORDS

START_3D
Marks the start of the graphical tridimensional definition of the component. It is placed after the header group.

STOP_3D
Marks the end of the graphical definition of the component.

The sentences START_3D and STOP_3D must be always present in any PDL file.

In case the component requires a special shape for insulation (different from the component itself), this shape can be defined inside the pdl file using pdl sentences between these key words:
START_INS
STOP_INS

5.4VARIABLES WITH VALUE ASSIGNED BY THE SYSTEM

R1 External radius of the nominal diameter.
R2 External radius of the secondary diameter, for reductions.
R3 External radius of the connection point 3.
R4, R5, R6, R7, R8, R9 External radius of the connection points 4, 5, 6, 7, 8 and 9
FT1 FR1 FT2 FR2 ... up to **FT8 FR8** thickness and radius of the flange plates, to each connection point.
PE1 PE2 ... up to **PE9** penetration value in case of threaded or socket welded end for each connection points
TH Pipe thickness in mm, if enabled.
P1 Parameter 1 value is defined, read from a table or input manually. Up to **P9**.
GAS Gasket thickness. If the end E1 is not flanged, this variable is set to zero.
WLD Welding thickness.

ANG Reference angle. Used for elbows. It if the value of the ANG field in \PD\STD\COD.DBF table. In the case of cut elbows is the cut angle.
LEN Reference Length. Used in pipes and olets.
GAP It is equal to: zero, GAS or WLD depending on the values of GAS and WLD.

5.5VARIABLES WITH VALUE ASSIGNED IN THE PDL FILE

The complete available variables are listed below:

S1= formula.
S2= up to **S20=**.
PT1= formula1 , formula2 , formula3
PT2= up to **PT40=**
SC1= formula1 , formula2 , formula3
SC2= up to **SC9=**
AN1= formula
AN2= up to **AN9=**

"formula" is used as an algebraic expression defined by:

- **Numbers**
- **Variables.** They can be one of the variable with automatically assigned values: Ri, FTi, FRi, Pi with i from 1 to 9 or a numerical variable with their values assigned in the PDL file: S1, S2, S3, S4, S5, S6, S7, S8, S9.
- **Arithmetic operator symbols** (+ - * /). A negative number must be expressed as the result of the operation: 0 - number.
- **Functions** Available functions are:
- set of parenthesis () : only one level is accepted, no nesting allowed
- SIN(formula) : returns the trigonometric sine of the angle expressed by formula
- COS(formula) : returns the trigonometric cosine of the angle expressed by formula
- TAN(formula) : returns the trigonometric tangent of the angle expressed by formula
- ASIN(formula) : returns the angle of the trigonometric sine expressed by formula
- ACOS(formula) : returns the angle of the trigonometric cosine expressed by formula
- ATAN(formula) : returns the angle of the trigonometric tangent expressed by formula
- SQRT(formula) : returns the square root of the number expressed by formula
- POW2(formula) : returns the power to 2 of the number expressed by formula
- FABS(formula) : returns the absolute value of the number expressed by formula

In the trigonometric functions the angle is expressed in radians. In all the functions, the first parenthesis is part of the function name and must be written with no spaces in between.
The numbers, variables, operator's symbols and function symbols may be separated by spaces or not. The computation of a formula is executed from left to right, with precedence of the parenthesis, multiplication and division over sum and difference. If different formulas are required by the syntax, a comma must separate each formula.

Si variables are used as internal variables, to store intermediate values, to be used later. They can reference to themselves.
PTi variables are used to define a point with its x,y,z coordinates, for example an insertion point.
SCi variables are used to define three scale components in the x, y, z direction. They are used to scale block insertions. No defaults are accepted in block insertions.

ANi variables are used to define angles, used in block insertions or by other commands. They are expressed in decimal degree units.

5.6INTERNAL USED VARIABLES WITH VALUE ASSIGNED IN THE PDL FILE

There are several types of these variables.

At any location of the PDL file, between the key words START_3D and STOP_3D the connection point definitions must be placed, using the following sentences:

CPi= formula1 , formula2 , formula3

The index i must go between 1 and the value defined by the NCP field in the \PD\STD\DDL.DBF table.
If the FACE field of the same table is equal to 1, also the opposite points of each connection points must be defined, using the sentence:

OPi= formula1 , formula2 , formula3

INTERF= formula_x- , formula_x+ , formula_y-, formula_y+ , formula_z- , formula_z+
Defines the circumscribed prism associated to the component for the interference checking. The six parameters are the dimension of the prism, along local reference axis.
In the case of linear components such as pipes, the firsts two parameter can be left zero, because the system automatically computes the correct values, based on the pipe length.

5.7CONDITIONAL SENTENCES

There is only one:

IF= formula

If the formula evaluates to an number <= zero, the following sentence to the IF= is skipped and not executed.

For example, the sentence:
IF= FT1
is not executed if the end is not flanged, because in this case FT1 = 0. This allows defining components in a very general way.

5.8 GRAPHIC ELEMENTS GENERATION

The following sentences are used to generate graphic elements:

INSERT block_name PTi SCj ANk
Inserts a block with its name indicated in the sentence. The insertion point is defined by the PTi variable, the scale SCj and the angle ANk. Acceptable block names are the following:
CYL Cylinder with unitary length and radius 1000. It is used in all cylindrical parts but flanges plates.
FLG Cylinder with unitary length and radius 1000. It is used only for flange plates.
SOC Cylinder with unitary length and radius 1000. It is used only for penetration in threaded and socket connections.
CONE Block with a cone shape with an opaque surface defined with 3DFACE and with the circle approximated by a polygon defined in the setup for the current diameter. Used in component definitions.
CONE_n Idem to CONE, but with the number of sides of the inscribed polygon equal to n. Used in equipment components.
BOX Rectangular Prism Block. Each side has unitary length. Origin is the center of the bottom face.
HESPH_n Hemisphere block, with the number of sides of the polygon approximating the circle equal to n.

LINE PTi PTj
Generates a line from the PTi point to the PTj point.

CIRCLE PTi Sj
Generates a circle with center in the PTi point and radius Sj.

ARC SCE PTi PTj PTk
Generates a circular arc using three points PTi PTj PTk. PTi is the start, PTj is the center, PTk is the end.

ARC SER PTi PTj Si
Generates a circular arc starting in the PTi point, ending in the PTj point and with the radius equal to Si.

PLINE PTi PTj ..PTk
Generates a polyline with vertex in the points received as parameters. With a final C it closes the polyline. Accepts ARC segments, using the ARC key word. To resume linear segment use the LINE keyword, as the analog PLINE AutoCAD® command does.
Example: PLINE PTi PTj ARC PTk LINE PTn ..
The polyline thickness is set to 0.

3DFACE PTi PTj PTk PTl
Generates a 3DFACE element with vertex in the points received as parameters. All sides will be visible. Use the following variations to set sides invisible:

3DFACE I PTi PTj PTk PTl -> invisible side: 1-2
3DFACE PTi I PTj PTk PTl -> invisible side: 2-3
3DFACE PTi PTj I PTk PTl -> invisible side: 3-4

CIRC_SEG PTi Si Sj ANi ANj 1 1 Sk
Generates a circular arc plane segment made by 3DFACES. PTi is the segment center, Si and Sj are the external and internal radius of the segment, ANi and ANj are the starting and ending angle of the segment (zero angle on the positive X axis, positive in counter clockwise direction, expressed in decimal DEGs). Staring and ending visibility codes follow (1=visible, 0=invisible). Sk is the number of segments used to approximate the circle.

CIRC_INV PTi Si ANi ANj 0 0 Sj
Generates an inverted circular sector made by 3DFACE elements. PTi is the segment center, Si is the segment radius, ANi and ANj the start and end angles (zero angle on the positive X axis, positive in counter clockwise direction, expressed in decimal DEGs). Staring and ending visibility codes follow (1=visible, 0=invisible). Sj is the number of segments used to approximate the circle.

ARC_SEG PTi Si Sj ANi ANj 1 1 Sk
Generates a circular arc made by 3DFACE elements perpendicular to the arc plane. PTi is the arc center, Si and Sj are the arc radius and height, ANi and ANj the start and end angles (zero angle on the positive X axis, positive in counter clockwise direction, expressed in decimal DEGs). Staring and ending visibility codes follow (1=visible, 0=invisible). Sj is the number of segments used to approximate the circle.

TRC_CONE S1 S2 S3 S4 PTi ANi S5
Generates a truncated cone made with 3DFACE elements. The normal to the greater base is aligned with the positive X axis, S1 is the diameter at the point PTi, S2 is the diameter at the second point, S3 is the cone height, S4 is the displacement along the Z axis of the center of the second point. ANi is the insertion angle in the XY plane (zero angle on the positive X axis, positive in counter clockwise direction, expressed in decimal DEGs) and S5 is the number of segments used to approximate the circles. If S5=0 uses the value defined in the setup for the current diameter.

TORUS_SEG S1 S2 S3 PTi ANi S4 S5
Generates a torus segment with variable section made with 3DFACE elements. S1 is the circle radius in the starting position 1, S2 is the circle radius at the end position 2, S3 is the torus radius, PT1 is the torus center, ANi is the segment angle (zero angle on the positive X axis, positive in counter clockwise direction, expressed in decimal DEGs), S4 is the number of segments used to approximate the circle, S5 is the number of segments used to approximate the torus. If S4=0 uses the value defined in the setup for the current diameter and number of segments for the torus.

COPY_MOVE PTi PTj
Copies the last graphic element drawn from the point PTi to the PTj point.

COPY_ROTATE PTi PTj ANi
Copies the last graphic element drawn from the point PTi to the PTj point and rotates it by the ANi angle with respect the PTj point.

5.9GRAPHIC ELEMENTS MODIFICATION

MOVE PTi PTj
Moves the last graphic element drawn from the point PTi to the PTj point.

ROTATE PTi ANj
Rotates the last graphic element drawn by the ANj angle around the current UCS Z axis, using the PTi point as reference.

ROTATE_X PTi ANj
Rotates the last graphic element drawn by the ANj angle around the current UCS X axis, using the PTi point as reference.

ROTATE_Y PTi ANj
Rotates the last graphic element drawn by the ANj angle around the current UCS Y axis, using the PTi point as reference.

UCS axis ANi
Rotates current UCS by the ANi angle around the selected axis. The "axis" code can only be X Y or Z.

UCS O PTi
Moves the current UCS origin to the PTi point.

UCS 3 PTi PTj PTk
Defines a new UCS using three points PTi, PTj and PTk. No defaults allowed: all points must be defined.

When the command starts generating graphic elements, the UCS is defined by the snap to a line route or a component. The origin of the local UCS is the component center, unless the UCS is modified inside the PDL file.
The variables that represent points and scales are relative to the UCS active at the moment the variable is used.
If the UCS is modified during the generation of the component, it has to be restored to its original UCS: in this UCS the connection points must be defined.

5.10 GENERIC COMMANDS

LAYER= name
Assigns the layer [name] to the first component that will be generated after this sentence. If the layer doesn't exist, it will be created. Following elements are generated in the default layer 0.

LINETYPE= name
Assigns the line type [name] to the first component that will be generated after this sentence. Following elements are generated with the CONTINUOS line type.

COLOR= number
Assigns the color [number] to the first component that will be generated after this sentence. Following elements are generated without setting the color.

THICKNESS= formula
Generates all elements from now on with this thickness. Starting the generation of a new component, the thickness is set to 0.0.

6 2D VIEW PARAMETRIC GENERATION

NOTE: THIS OPTION HAS BEEN TEMPORARILY DISABLED IN V2015.0

6.1INTRODUCTION

Orthographic Plans are generated from 3D models using the 2D View Extraction command. See User Manual 4.9.

In principle, there is no need to define a 2D representation for each 3D piping component or equipment element created by EPLANT, because both are built using EPLANT Primitive Solids which automatically have defined a 2D view representation for both the **Opaque** or **Transparent** options to the AutoCAD® Hide command. These options are set in the characteristics of each 2D view.

If the equipments use other elements created with plain AutoCAD®, these elements are copied to the extraction layer without any processing.

In case a different representation from the automatic one is needed, a parametric definition can be used using the same PDL language used to 3D piping component generation.

This possibility is also used to create simple line symbology for piping component.

2D parametric symbology definition uses files with names [component_code]_2D.PDL placed in the project 2D Plan Symbology directory, by default id P2D. The component code is the value of the COD field in the COD.DBF table.

The system installs a library of single line symbols quite extensive, with few double line symbology that is intended to be used only when the automatic representation is not well suited.

6.2SYNTAX OF 2D PDL FILE

2D PDL files are text files that contain instructions to generate 2D views of piping components using a parametric definition. They can be modified using any text editor, such as NotePad.
Sentences are written in PDL language. The system makes syntax checking during the reading of these files and warns of any error.

First two lines are a mandatory header, each line begins with a fixed part that may continue with a description. As an example, for the 90B code the file will be named 90B_2D.PDL and will contain the following:

90B Elbow 90 for SW and SC
TH=NO

First Line: 90B repeats the generic component code followed by a blank space and a description not interpreted by the system.

Second Line: TH=NO specifies that the piping thickness will not be read. The sentence TH=YES activates the reading of the piping thickness from the \PD\STD\ANSI\PIPT.DBF table. In this case the variable TH will be assigned the read value, otherwise it will be set to 0.

Generally a 2D PDL file will begin with those first lines followed by a sets of sentences between key words. Each set defines a particular view.

6.3 AVAILABLE VARIABLES

All variables described in chapter 5 can be used. The **Pi** variables contain here the dimensional parameter really used in the 3D component generation.
Point variables **PT1**, **PT2**, ... are assigned the position of connection points expressed in UCS E of the 3D component.
The following special variables are also defined using the names **PT1[0]**, **PT1[1]**, **PT1[2], PT2[0]**, etc. which contain the coordenates of all connection points to be able to use them separately.

6.4 KEY WORDS

Key words are used to specify set of sentences used to create a specific view, identified by the key word itself. There are two sets of keys: for Double Line and Simple Line Symbology. In case of Double Line there are two additional options: Opaque and Transparent to AutoCAD® Hide command. Opaque/Transparent option is selected in the extraction view command.

6.4.1 Double Line Symbology

VIEW_XH
Marks the start of the definition of an Opaque projection along the local component X axis. The end mark is taken as the first VIEW_ sentence.

VIEW_XT idem to VIEW_XH but with **Transparent** projection.

VIEW_YH
Idem to VIEW_XH for the **Opaque** projection along the local component Y axis.

VIEW_YT idem to VIEW_YH but with **Transparent** projection.

VIEW_ZH
Idem to VIEW_XH for the **Opaque** projection along the local component Z axis.

VIEW_ZT idem to VIEW_ZH but with **Transparent** projection.

VIEW_NXH
Idem to VIEW_XH for **Opaque** projection, along the oriented view of the current UCS. It is used only for axisymmetric components having the NX equal to 1 in the \PD\STD\COD.DBF table.

VIEW_NXT idem to VIEW_NXH but with **Transparent** projection.

6.4.2 Single Line Symbology

VIEW_X1
Marks the start of the definition of a Single Line Symbology projection along the local component X axis. The end mark is taken as the first VIEW_ sentence.

VIEW_Y1
Idem to VIEW_X1 for projection along the local component Y axis.

VIEW_Z1
Idem to VIEW_X1 for projection along the local component Z axis.

VIEW_A1
Single Line Symbology projection independent of the component orientation.

6.4.3 Single Line Symbology – Support Only

In the special case of 2D Projections generated with the Support Structure command, the following view names are recognized:

VIEW_SXT
For Transparent projection only.

In this case, the projection is generated scaled with the value specified in the corresponding command and available in the script as the variable **DSC**. Dimensions in these projection must be explicitly scaled by that amount.

6.5SINGLE LINE END SYMBOLOGY

En symbols generation for single line symbology is automatic and uses the following predefined blocks:

BW_X.DWG half disc for symbol BW for any view different from Z.
BW_X.DWG half disc for symbol BW for any other view.
RF_X.DWG half disc for symbol RF for any view different from Z.
RF_X.DWG half disc for symbol RF for any other view.
Etc. for other codes.

End codes are obtained using the same rule used in the name of isometric symbols.

NO_END_SYM=n
Disables the generation of the end symbol for the end n.

7. ISOMETRIC SYMBOLS PARAMETRIC GENERATION

7.1INTRODUCTION

Almost all piping components have one or more symbols associated to them that are used in isometric representation. These symbols are statically defined using a command placed in the isometric menu.
Only those components that have the value 1 in the field ISO_PDL of the COD.DBF table need a dynamic definition. This is achieved using a file containing the parametric definition of the symbol using the PDL language (EPLANT Parametric Definition Language).
To each of these components there must correspond a file with the generic component code followed by _ISO end extension .PDL. These files must be present in the project isometric symbology directory.

7.2_ISO.PDL FILE SYNTAX

_ISO.PDL files are text files that contain instructions to generate a dynamic isometric symbol in a parametric form. Those files can be edited with the NotePad or any equivalent text editor.
The instructions must be written in the PDL language, in the isometric flavor. It is basically the same syntax as accepted in the 3D definition files, with minor differences. The system performs a syntax verification reading the file, warning any inconsistency found.

Firts line is a header. For instance, for the SVB code the file will be named SVB_ISO.PDL and will contain the following:

SVB Sanitary Valve type B = 3 cylinders + 2 flanges

First line: SVB is the very component code, followed by a space and a description of the component, if any.

After the parameter definitions, an arbitrary number of sentences follow, to define the shape and dimensions of the component.
Lines with a preceding * symbol are skipped as commentary.

Apart from the header, PDL sentences fit in the following four types:

- Key words: set the start and end of a group of sentences.
- Value setting to a variable: variables have fixed names.
- Conditional sentences. They control the execution of other sentences.
- Generation of a graphic element with characteristics defined by variables.

7.3KEY WORDS

START_ISO
Marks the start of the graphical definition of the isometric dynamic symbol of the component. It is placed after the header group.

STOP_ISO
Marks the end of the graphical definition of the symbol.

The sentences START_ISO and STOP_ISO must be always present in any PDL file.

7.4VARIABLES WITH VALUE ASSIGNED BY THE SYSTEM

P1 Dimensional Parameter 1 used by the 3D component.
If this component has more than one parameter, the **Pi** with i from 1 to 9 variables will contain the corresponding additional parameters. Parameter ar not read from dimensional tables, are directly read from the definition of the 3D component.

E1 Isometric end code equivalent to the end code of the 3D component connection point 1.
If the component has more than one connection point, the variables **Ei** from 1 to 9 will contain the values of the iso end codes corresponding to the other connection points.

The name of the block used to represent the iso symbol generated with an _ISO.PDL file contains the values of those parameters that have in their definition (in the PDL file defining the 3D component) the PAR=0 key that allows to input a value equal to zero for that parameter. It is used when the parameter represents a rotation angle.

7.5VARIABLES WITH VALUE ASSIGNED IN THE _ISO.PDL FILE

The complete available variables are listed below:

S1= formula.
S2= up to **S9=**.
PT1= formula1 , formula2 , formula3
PT2= up to **PT40=**
SC1= formula1 , formula2 , formula3
SC2= up to **SC9=**
AN1= formula
AN2= up to **AN9=**

"formula" is used as an algebraic expression defined by:

- **Numbers**
- **Variables.** They can be one of the variable with automatically assigned values: Ei, Pi with i from 1 to 9 or a numerical variable with their values assigned in the PDL file: S1, S2, S3, S4, S5, S6, S7, S8, S9.
- **Arithmetic operator symbols** (+ - * /). A negative number must be expressed as the result of the operation: 0 - number.

- **Functions** Available functions are:
- set of parenthesis () : only one level is accepted, no nesting allowed
- SIN(formula) : returns the trigonometric sine of the angle expressed by formula
- COS(formula) : returns the trigonometric cosine of the angle expressed by formula
- TAN(formula) : returns the trigonometric tangent of the angle expressed by formula
- ASIN(formula) : returns the angle of the trigonometric sine expressed by formula
- ACOS(formula) : returns the angle of the trigonometric cosine expressed by formula
- ATAN(formula) : returns the angle of the trigonometric tangent expressed by formula
- SQRT(formula) : returns the square root of the number expressed by formula
- POW2(formula) : returns the power to 2 of the number expressed by formula
- FABS(formula) : returns the absolute value of the number expressed by formula

In the trigonometric functions the angle is expressed in radians. In all the functions, the first parenthesis is part of the function name and must be written with no spaces in between.
The numbers, variables, operator's symbols and function symbols may be separated by spaces or not. The computation of a formula is executed from left to right, with precedence of the parenthesis, multiplication and division over sum and difference. If different formulas are required by the syntax, a comma must separate each formula.

Si variables are used as internal variables, to store intermediate values, to be used later. They can reference to themselves.
PTi variables are used to define a point with its x,y,z coordinates, for example an insertion point.
SCi variables are used to define three scale components in the x, y, z direction. They are used to scale block insertions. No defaults are accepted in block insertions.
ANi variables are used to define angles, used in block insertions or by other commands. They are expressed in decimal degree units.

7.6INTERNAL USED VARIABLES WITH VALUE ASSIGNED IN THE _ISO.PDL FILE

There are two types of these variables.

At any location of the PDL file, between the key words START_3D and STOP_3D the connection point definitions must be placed, using the following sentences:

CPi= formula1 , formula2 , formula3

The index i must go between 1 and the value defined by the NCP field in the \PD\STD\COD.DBF table.
If the FACE field of the same table is equal to 1, also the opposite points of each connection points must be defined, using the sentence:

OPi= formula1 , formula2 , formula3

In both cases the UCS at the point of execution of any of those sentences must be the block UCS.

7.7CONDITIONAL SENTENCES

There are two types, the first one is (enabled in every other flavors of PDL language):

IF= formula

If the formula evaluates to an number <= zero, the following sentence to the IF= is skipped and not executed.

For example, the sentence:
IF= FT1
is not executed if the end is not flanged, because in this case FT1 = 0. This allows defining components in a very general way.

The other one is the following:

IFE Ei=isometric end code

Where Ei is the variable that represent the value of the isometric end code of the end number one. If both codes are the same the sentence following this one is executed, otherwise that sentence is ignored.
Exemple:

IFE E1=RF

7.8GRAPHIC ELEMENTS GENERATION

The same options available in chapter 5.8.

7.9GRAPHIC ELEMENTS MODIFICATION

The same options available in chapter 5.9.

7.10 GENERIC COMMANDS

The same options available in chapter 5.10.

8. EQUIPMENTS PARAMETRIC GENERATION

8.1INTRODUCTION

Equipments can be parametrically defined using the EDL language.
In this case, some of the equipment parameters must be defined in the \PD\EDL\EQU_DES.DBF and \PD\EDL\EQU_DIM.DBF equipment tables.
The EQU_DIM.DBF table is only used by equipment with preloaded dimensions, otherwise is left void.

To activate the generation of a parametric equipment, use the following sentence:

(eq_cmd "PARAM" "pump")

"pump" in this example is a defined class for pumps in the EQU_DES.DBF table. The following menu allows selecting the equipment specific name.

To each equipment defined by a name in the CODE field, in the \PD\EDL\EQU_DES.DBF table, a file with the same name and EDL extension in the directory \PD\EDL must correspond.
To add a new equipment definition, a slide with its image must be generated and included in the \PD\EDL\EQU_DES.SLB slide library.

8.2EDL FILE SYNTAX

EDL files are text files with sentences for the parametric generation of equipment with nozzles (up to 5). They can be modified with any text editor.
Sentences must be written in the EDL language. The system verifies the syntax when reading the file, warning any error found.

The first line is a required header, it must repeat the equipment code. The rest of the lines can be any combination of:

- Commenting line: it has * as the first character of the line.
- Key words.
- Instructions to define graphic elements that make the equipment.
- Instructions to define piping components belonging to an equipment.

8.3 KEY WORD

START_EQU
Marks the start of the equipment definition.

STOP_EQU
Marks the stop of the equipment definition.

START_PIPE
Marks the start of the definition of a piping component included in the equipment.

STOP_PIPE
Marks the end of the definition of a piping component included in the equipment.

8.4EQUIPMENT GEOMETRY DEFINITION SYNTAX

EDL language uses the same syntax as the PDL language, without the sentences CPi and OPi. If any parameters are used, they are associated to variables with names P1, P2, etc. up to P9.

8.5PIPING COMPONENTS DEFINITION SYNTAX

To assign a value to each component parameter a specific sentence is used for each different parameter. The sentence name is the same as the corresponding field name in the piping specification table. The available parameters are:

STD=
PCLA=
ICLA=
D1=
D2=
D3=
D4=
E1=
E2=
E3=
E4=
RAT=
RAT2=
SCH=
SCH2=
MAT=
CODA=
GAP1=
GAP2=
GAP3=
GAP4=
LEN=
POS_MODE=

Following the = symbol the value of the parameter can be directly placed or a specific EDL variable may be used to define nozzles, identified as:
DIA_i
END_i
RAT_i
SCH_i
LEN_i
GAP_i

with i identifying the nozzle number, from 1 to 5. The values of these variables are read from the equipment dimension table or manually input in the parametric equipment definition window.

At last, the sentence:

PIPE_GEN

Activates the component generation in the current UCS, that must be set according to each nozzle position and orientation.

9. LINE EXTRACTION TO PCF FILE SETUP

There is a command to generate PCF files (compatible to the Isogen isometric generator program). See the User Manual chapter 4.13.13 for details.

The extraction is setup using the parameters stored in several files stored in the /project/ISOGEN/SETUP folder. The files are the following:

ISOSET.DBF contains the main settings. Its content is detailed below:

CODE	VALUE_C	VALUE_N	MEANING
BORE	INCH		Nominal Diameters: Value_c = INCH / MM
CO-ORDS	MM		Dimensions: Value_c = INCH / MM
WEIGHT	KG		Weight unit: Value_c = KG is the only one supported so far
BOLT-DIA	INCH		Bolt Diameter: Value_c = INCH / MM
BOLT-LENGT	MM		Bolt Length: Value_c = INCH / MM Currently the CO-ORDS code is used
ITEM-CODE		1	Item Code is built using: Value_n = 1: EPLANT Internal Code Value_n = 2: EPLANT External Code Value_n = 3: EPLANT Alternate Code
MTO_LIST		70	Value_c = definition of the description text associated to each item-code Value_n = maximum number of characters per line in the description associated to each material
FL_NOZZLE		1	Not used

CODMAP.DBF contains the mapping between each EPLANT object and the corresponding Isogen codes. Default table that is created in a new project contains a mapping of currently defined EPLANT objects. Should the user create new ones, they need to be added to this table:

COLUMN	DESCRIPTION
COD	EPLANT COD code, as defined in the /pd/std/COD.DBF system table
DES_E	Generic description associated to each code: used for reference only
E1	EPLANT end code for main diameter: if left blank any end code is accepted
E2	EPLANT end code for reduction diameter: if left blank any end code is accepted
PCF_ID	PCF identification code
SKEY	SKEY code used to associate a given Isogen symbol

ENDMAP.DBF contains the mapping between each EPLANT end codes and the corresponding Isogen end codes. Default table that is created in a new project contains a mapping of currently defined EPLANT end codes. Should the user create new ones, they need to be added to this table:

COLUMN	DESCRIPTION
END	EPLANT END code, as defined in the /pd/std/END.DBF system table
PCF_END	Isogen code corresponding to the EPLANT one

All the remaining files are specific for Isogen configuration. Refer to Isogen documentation for details.