

System documentation
for

C2I

Interface between CADD5[®] 5i and ISOGEN[®]

Contents

1	Introduction	3
2	How does it work?	4
3	System components.....	5
4	cadds2pcf	6
4.1	Dialog	6
4.2	Output contents	7
4.3	Settings file	8
5	C2IClient for Windows.....	10
6	C2IClientSetup for Windows	11
6.1	Settings	11
6.1.1	Server tab.....	12
6.1.2	CADDs output tab.....	13
7	C2IClient for unix.....	14
8	C2IRun	15
8.1	Component mapping	17
8.2	Property mapping	19
8.3	End type mapping	20
8.4	Component definition file	21
8.5	Log file	22
9	C2ISetup.....	24
9.1	Main	25
9.2	Mapping	26
9.3	Isogen	27
	Aknowledgements	28
	References	28

1 Introduction

CADDS[®] 5i is a 3D CAD/CAM solution with modules to facilitate the design and detail of complex plant layouts and ships. Typically the design and detail of piping layouts requires the production of large amounts of isometric detail drawings. The automatic generation of these isometric drawings will allow engineering staff to concentrate on design activities therefore reducing the design cycle time span.

Alias ISOGEN[®] is the industry standard solution for the automatic generation of isometric piping detail drawings. The Cadds-2-Isogen interface offers support to ISOGEN[®] as a powerful alternative to the existing CADDS[®] 5i isometric drawing generation module.

This document provides an overview of the capabilities of the Cadds-2-Isogen interface (C2I), including details of each component part.

2 How does it work?

The interface supports CADD5[®] 5i in both the UNIX and Windows[®] environments. The actual ISOGEN[®] program is available as a dynamic-link-library (exeisogen.dll) on Windows[®] only. Therefore there must be a PC running Windows[®] in the network to use ISOGEN[®].

The interface operates as follows:

- An extract is generated from the design containing the elements forming the required piping segment for the isometric, for example a spool.
- A CVMAC[™] routine is run in the active piping part which does the following:
 - Creates a file with the contents of the piping data (geometry, connections...)
 - Starts a process (C2IClient) which processes this file through ISOGEN[®]
 - Waits until the result is produced
 - If required, recalls the result into CADD5[®] 5i (when added to C2IClient). As the output from Isogen is a DXF file, a DXF interface must be available to CADD5
- On the Windows[®] machine, a program (C2IRun) is running to:
 - See if a request to make an isometric has been made
 - Converts the data file to a PCF file (input for ISOGEN[®])
 - Calls ISOGEN[®] with the appropriate options
 - Places the result (one or more DXF files) in the correct location for C2IClient to see that a successful run has been made.

The ISOGEN[®] options to produce each type of isometric are part of the Isogen project and are fully customisable through interactive menus. Also the mapping-table of the ISOGEN[®] symbols can be adapted to the need of each project. The relevant ISOGEN[®] manuals for this are available on line at <http://www.alias.ltd.uk/documentation.htm>.

3 System components

The interface uses the following components:

At each client:

- CVMAC™ RunTime for C2I (cadds2pcf.cm + cadds2pcf.dm)
- C2IClient (script on UNIX, program on Windows®)
- C2IClientSetup (only on Windows®) for control of C2IClient
- Optional: A conversion program to read the DXF file into CADD5® 5i (when not part of your release of CADD5® 5i)

At the server (must be a Windows® -system)

- C2IRun.exe for conversion
- C2ISetup for controlling C2IRun
- ISOGEN® software with proper project settings to suit your requirements.
- The mapping files for components, properties and end types.

This interface has been tested with ISOGEN® exeisogen.dll version 1.1.0.4.

4 cadds2pcf

This macro starts the conversion process when called by the CADDs user. The runtime converts the piping content of the active model to an ASCII file. A number of properties for this process can be set in a settings file: `c2isettings` (see 4.3), which must be readable through the CADDs search path.

When the ASCII file has been made, the `C2IClient` script or program is called with the ASCII file as parameter. `C2IClient` will wait until the conversion process is terminated and the DXF-files are generated (or an error has occurred). `C2IClient` generates a results file and this file is read by the macro. The user also gets a result message at the screen.

4.1 Dialog

The `cvmac` is started by typing the following command

```
Run CVMAC cadds2pcf
```

If the settings file is incorrect, not present or cannot be read, the following message is given:

```
Cannot read c2isettings
```

And the program will terminate. The user must then make sure that the settings file can be read properly. When there is no project defined in the settings file, the program will ask for a project to be applied by `ISOGEN®`. This project must have been previously created using the `Isogen Project Manager Utility`. If required the project may be set in the `C2IRUN` setup process, which will override the project set here.

```
Please enter project to be applied:
```

If there is no style defined in the settings file, the program will ask for a style to be applied by `ISOGEN®`. If required the style may be set in the `C2IRUN` setup process, which will override the style set here.

```
Please enter style to be applied:
```

When this information is entered, the program will create the output file.

When ready the file will be sent to the server for processing to DXF. The following message is displayed to indicate this:

```
Waiting for isogen-server to process file  
<hostname> <part name> <result file name>
```

When processing is successful, and the DXF-files created by ISOGEN[®] have been moved back to the workstation, the following message is given:

DXF-files have been made

If an error occurs, or a time-out occurs, appropriate messages are displayed:

Isogen-server returned the following message:

<message content>

Depending on the error, actions should be taken, e.g. modifying the model. In some cases inaccuracies in positioning can cause errors during processing by ISOGEN[®].

4.2 Output contents

The whole content of the active model is converted to ISOGEN[®]. If there are not un-connected cnodes, this will result in multiple DXF-drawings from ISOGEN[®]. The name of the output file is the name of the model (until first dot) followed by the extension '.adr'. When CADDs entities have properties, these will be written to the output file first. If a text property contains a <cr>, then this will be replaced by a <\$>, which will be transformed back to a <cr> by ISOGEN[®].

First all cnodes are written to the output file with their properties.

After the cnodes the nlines are written to the output file. Lines shorter than 0.5 mm are not processed by ISOGEN[®]. If you enter a value for MINLINE larger than 0.5 in the settingsfile, a warning will be given for each line shorter than MINLINE.

When an nline has a BRAD property, the middle of the line will be written as an extra point. This point is required to determine the proper position of the bend.

Text related to the line are written as messages that will appear in a rounded box on the Isometric drawing.

Finally all nfigures will be written to the outputfile. The orientation and parent/child information will be written to the file, followed by a list of all connected cnodes and the figure properties.

Example of an output file:

```
# Written by CVMAC cadds2pcf - version 29-01-2001
UNITS
  3 ISOGEN-FILES = ISOGEN.FLS
  3 UNITS-BORE = INCH
  3 UNITS-CO-ORDS = MM
  3 UNITS-BOLT-LENGTH = MM
  3 UNITS-BOLT-DIA = MM
  3 UNITS-WEIGHT = KGS
MODEL
  3 PIPELINE-REFERENCE = ppibelw1
  3 DATE-DMY = 01/02/01
  3 ISOGEN-PROJECT = FROMNAME
  3 ISOGEN-STYLE = fab
CNODE      5
  POSITION      3730.64   -870.81     0.00
CNODE      6
  POSITION      3730.64   -870.81    3000.00
```

```

CNODE      9
  POSITION   3730.64   -870.81   1142.19
  0 CONNECTOR
  3 ENDTYPE   = "BW"
CNODE     11
  POSITION   3730.64   -870.81   1392.19
  0 CONNECTOR
  3 ENDTYPE   = "BW"
NLIN      4
  CNODES    5      9
  TEXT 2 PP-01-6"-1C
  2 OUTERDIAM = 0.16830
  2 INNERDIAM = 0.15410
  3 STOCKNO   = "PIPE-BE-40-SA106B-06"
NLIN     10
  CNODES    9      11
  0 INTERNAL
NLIN     13
  CNODES   11      6
  2 OUTERDIAM = 0.16830
  2 INNERDIAM = 0.15410
  3 STOCKNO   = "PIPE-BE-40-SA106B-06"
NFIG      8
  POSITION   3730.64   -870.81   1142.19
  XORIENT    0.00   1000.00    0.00
  YORIENT    0.00    0.00   1000.00
  ZORIENT   1000.00    0.00    0.00
  CNODES     9      11
  TEXT 3 LGLGK
  POSITION   3543.14   -683.31   1267.19
  1 FITTYPE   = 10
  3 COMPNAME  = "BELW"
  3 STOCKNO   = "BELW"

```

4.3 Settings file

Below is an example of the settings file c2isetngs with an explanation of the variables:

```

OUTFILEDIR =tmp.isogen.
CHECKFILEDIR /tmp/isogen/
OUTFILEDRIVE C:
CLIENTPATH /dvl/cs/c2iclient
MINLINE 20
PROJECT FROMNAME
STYLE ourstyle
BOREUNIT MM
END

```

```

# This file contains the settings required for proper operation of the
# CADD2PCF macro, part of the C2I system.
# The lines consist of a property name and a value, optionally followed
# by any other information. The property names are fixed and should not
# be changed.
# The set is terminated by an END line. This line should not be changed,
# since it is used by the program to terminate reading as soon as possible.
# The name of this file should not be changed, and the file must be placed
# at a directory in the cadds-search path (e.g. usr/apl/cadds/data/_bcd/).
#
# OUTFILEDIR   : The name of the directory to place the output file.
#               Must be specified in the CADD2PCF file format.
# CHECKFILEDIR : The name of the directory where c2iclient should place
#               its result file. Should be specified in the unix file
#               format.
# OUTFILEDRIVE : Name of the drive for the output and check files. This
#               is only required for NT environment when the files are
#               not placed at the same drive as the home drive of Cadds.
# CLIENTPATH   : full path to the c2iclient program or script. Must be
#               specified in unix format.

```

```
# MINLINE      : Minimum required line length. If a line is smaller, a
#               warning will be generated. Lines smaller than 0.5 mm
#               are never processed as pipelines. Leave out value or
#               set to value smaller than 0.5 to omit warnings.
# PROJECT      : ISOGEN® project to be applied for DXF-output. This should
#               only be added when the project is always the same, since
#               it suppresses the dialog to input the projet. If value
#               FROMNAME is given, then the project name is derived from
#               the first 3 characters of the partname.
# STYLE        : Isogen style to be applied for DXF-output. This should
#               only be added when the style is always the same, since
#               it suppresses the dialog to input the style.
# BOREUNIT     : Determines which unit is used for bore diameters. Can
#               be MM or INCH
#
# Name: c2isettings
# Created: 2001-08-09/AMV
```

5 C2IClient for Windows

The C2IClient for Windows is implemented as a VB-exe. The program has no user interface, and the local settings can be set or modified through the C2IClientSetup program.

The program is started from the cadds2pcf CVMAC. The settings are read from the registry. If the settings are not present, an error message will be written to the result file.

The output file is converted to a dos-format file (if not already in that format). The hostname of the client is determined, and added to the name of the .adr file. Then the file is moved to the server, followed by a .cmpl file to indicate that the file has been moved completely.

It will then wait for resulting DXF-files. The maximum waiting time can be set through the settings. If the completion file is found (hostname.cmpl) at the server, the resulting DXF files are copied to the client result directory.

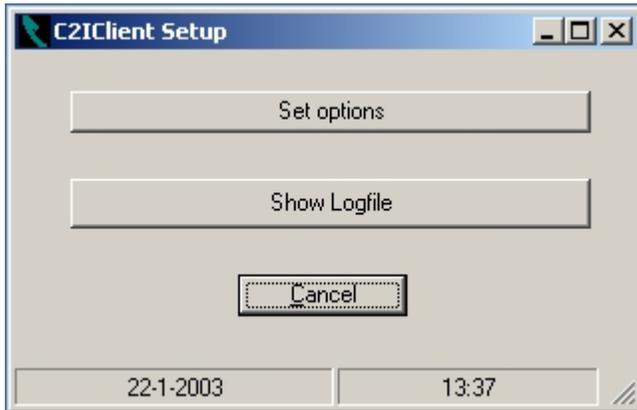
The result is written to the result file, so the macro can inform the user about the result.

C2IClient will write log messages to a log file when processing files. The contents of the log file can be use to solve problems.

6 C2IClientSetup for Windows

C2IClientSetup controls the settings for C2IClient. The settings are stored in the current user section of the registry. This allows for different settings for different users at the same computer.

When started, the main form will appear:



The form shows the current date and time at the bottom. The first button will display the settings dialog, the second the current log file. The log file can be useful to determine causes of problems.

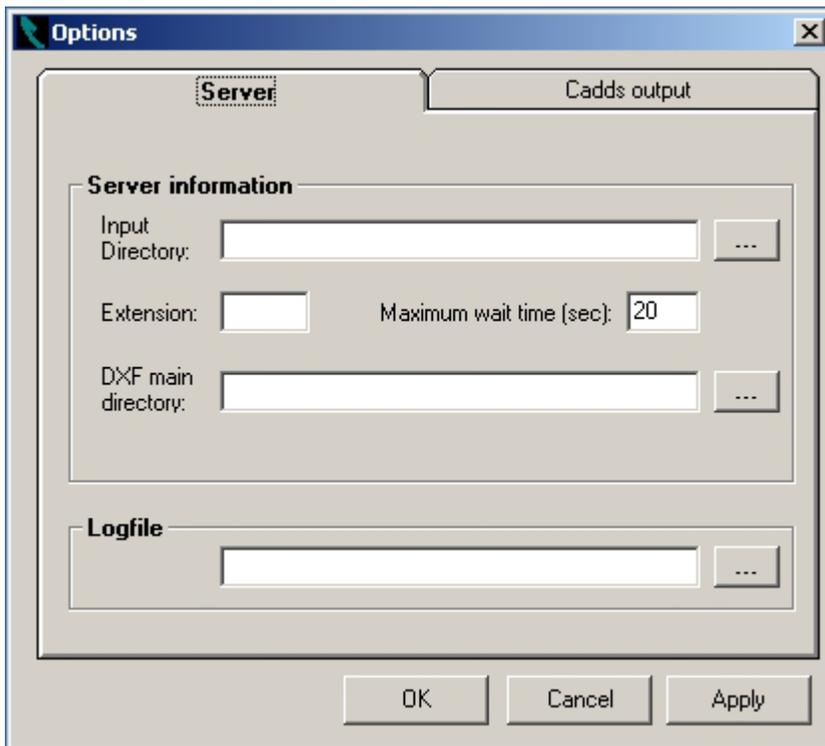
6.1 Settings

The options setting dialog has two tabs, one to set the options for the server side (where to copy to, what is the server expecting) and the log file, and one to set the options for the CADDSS side.

The directories and filenames can be typed, or can be selected through standard windows dialogs.

To store the settings click 'OK' or 'Apply'. To leave the dialog without saving values click 'Cancel'.

6.1.1 Server tab



The following settings can be entered:

Input directory:

Directory on the server where CADDs output files are placed. This directory should be given with a drive letter (K:\...) of a mapped network drive, and **not** with UNC ([\\server\...\...](#)).

Extension:

Extension of the files the server is waiting for (default is .adr).

Maximum waiting time (sec):

The maximum time the program waits for processing of the file. Depending on network load and number of clients, this value might need a higher value than the default 20 seconds.

DXF main directory:

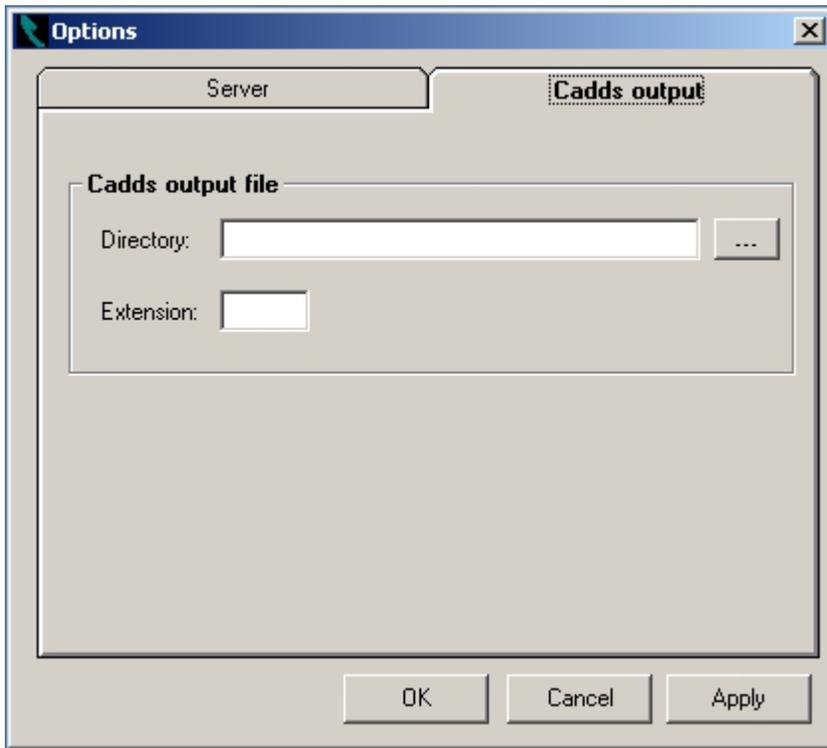
Main directory for the resulting DXF files. The server will place the DXF-files at a subdirectory of this directory with the hostname of the client. So if the main DXF directory is `..\DXF\`, and your hostname is oak, then the DXF files are placed in `..\DXF\oak`. This is to make processing from several clients possible.

Logfile:

The full name of the log file for C2IClient.

6.1.2 CADDs output tab

The CADDs output tab has the following contents:



The following settings can be entered for the CADDs output file:

Directory:

The directory where the cadds2pcf macro puts the output file. This directory should be given with a drive letter (K:\...) of a mapped network drive, and **not** with UNC ([\\server\...\...](#)).

Extension:

Extension of the CADDs output file (default is .adr).

7 C2IClient for Unix

C2IClient for Unix is implemented as a script (cshell) and performs the following actions:

- Adds the name of the client computer to the .adr file
- Sends it through ftp to the Windows server.
- Sends a file with the same name and extension .cmpl to the server. This causes the server to process the .adr file. This is done to make sure that the whole .adr file has arrived at the server before the server starts to read the file.
- Waits a certain time and then checks if there is a file <hostname>.cmpl present at the server output directory. When found, all DXF files of this same directory are moved to the client through ftp. If no result is found within a certain period, a time-out occurs.
- Writes a result file, which is read by the cadds2pcf macro, to inform the user of the result.

8 C2IRun

C2IRun is the server program that does the conversion of the CADDSS output file to a PCF file, which is in turn processed by ISOGEN to a DXF file.

C2IRun has no user interface, and should be running as a service. However, it is not a real service (it is written in VB6). This means it has to be started, but then can run unattended.

It is possible to run the program as a service when using SrvAny, which is part of the NT-Resource Kit.

C2IRun has a number of settings, which can be set through the C2ISetup program (see 9).

The C2IRun program should be running permanently. It will automatically process files when they are placed at the input directory. To make sure that the input files are completely copied to this directory before C2IRun reads them, processing is only done when a file with the same name, but with the extension .cmpl is found in the directory. This file will be deleted automatically to prevent double processing. By default the input file will also be deleted, but one can choose to have the file renamed (e.g. for testing purposes). In that case the extension will be replaced by a timestamp.

To convert the CADDSS information to ISOGEN information the program has to convert nlines to pipes and nfigures to components. However, not every CADDSS user gives the same meaning to the nfigures in CADDSS. Therefore the conversion uses user-definable files to make the mapping between nfigures and components (component mapping file, see 8.1), between properties and attributes (property mapping file, see 8.2) and between CADDSS-end types and ISOGEN-end types (end type mapping file (see 8.3). Which output is required for a specific ISOGEN component is defined in .def files (see 8.4). In this way the system can be extended with new components without changing the program itself. Description of the mapping files is given in the following paragraphs.

Every component in ISOGEN[®] requires a specific lay-out in the PCF-file. However, some components can have more than one lay-out, depending on the values of the SKey or attributes. The layout of each ISOGEN[®] component used in the component mapping file should be defined in a component definition file (see 8.4).

C2IRun removes all cnodes which are not connected to an nline or an nfigure.

Not all cnodes in CADDSS[®] will have a nominal pipe size (NPS) property. However, ISOGEN[®] requires this value at almost all end-point coordinate definitions. In order to provide a value C2IRun will first try to supply each node with an NPS-value. This is done by the following mechanisms:

- If an nfigure has an NPS property, then the property will be copied to all cnodes of the component that do not have an NPS-property yet.
- If not all cnodes of an nfigure have an NPS, but at least one does, then all nodes get the same NPS.

- If only one endpoint of an nline has an NPS, then the other one will be assigned the same NPS.
 - If cnodes coincide, then all these cnodes will get the same NPS.
- These mechanisms are applied repeatedly. If still not all cnodes have an NPS, the program will check the STOCKNO property of the nlines, and will use the last part of this number as the NPS (if numeric).
- Processing is halted when the program cannot give each cnode an NPS.

The CADD[®]S database identifier will be used as the value for the attribute 'UNIQUE-COMPONENT-IDENTIFIER'. This makes it easier to solve problems related to specific nlines and nfigures.

Spools are only created when the property SPOOL is encountered. Spool information is written to the PCF file with each component.

When all data is found to be correct, the program will write a PCF file, which is subsequently processed by ISOGEN[®] to create the dxf-files.

The PCF file created from the example .adr file from 4.2 is as follows:

```

ISOGEN-FILES  ISOGEN.FLS
UNITS-BORE   INCH
UNITS-CO-ORDS MM
UNITS-BOLT-LENGTH MM
UNITS-BOLT-DIA MM
UNITS-WEIGHT KGS
      PIPELINE-REFERENCE  ppibelw1
      DATE-DMY  01/02/01
PIPE
  END-POINT      3730.64      -870.81      0.00  6
  END-POINT      3730.64      -870.81      1142.19  6  BW
  ITEM-CODE  PIPE-BE-40-SA106B-06
  UNIQUE-COMPONENT-IDENTIFIER  4
MESSAGE-POINTED
  CO-ORDS      3730.64      -870.81      571.10
  TEXT  PP-01-6"-1C
  UNDIMENSIONED
PIPE
  END-POINT      3730.64      -870.81      1392.19  6  BW
  END-POINT      3730.64      -870.81      3000.00  6
  ITEM-CODE  PIPE-BE-40-SA106B-06
  UNIQUE-COMPONENT-IDENTIFIER  13
MISC-COMPONENT
  END-POINT      3730.64      -870.81      1142.19  6  BW
  END-POINT      3730.64      -870.81      1392.19  6  BW
  CENTRE-POINT      3730.64      -870.81      1142.19
  SKEY  EXBW
  ITEM-CODE  BELW
  UNIQUE-COMPONENT-IDENTIFIER  8
MESSAGE-ROUND
  TEXT  LGLGK
END-POSITION-NULL
  CO-ORDS      3730.64      -870.81      0.00
END-POSITION-NULL
  CO-ORDS      3730.64      -870.81      3000.00
MATERIALS
ITEM-CODE  PIPE-BE-40-SA106B-06
      DESCRIPTION  "PIPE-BE-40-SA106B-06"
ITEM-CODE  BELW
      DESCRIPTION  "BELW"

```

The creation of the DXF output of the isometric from the PCF input is done by ISOGEN[®] whose customisation is not part of this document.

However, the ISOGEN[®] data directory structure reflects ISOGEN[®] project names and style names. C2IRun uses this information to check if the project and style to use are present.

The name of the project and style to use (determines drawing size and layout) can be set in the c2isettings file, or can be entered by the user.

If FROMNAME is set as the project name, the actual name of the ISOGEN[®] project is determined by looking at the first 3 characters of the CADD[®] part name. The project with the same starting characters will be used. If not found, the default project as set with C2ISetup will be used.

The name of the style to use is taken from the model property ISOGEN-STYLE. The value of this property is requested when creating the .adr file (cvmac dialog) or set as default in c2isettings.

8.1 Component mapping

Component mapping is performed to allow for company specific use of both nfigures in CADD[®] and Components in ISOGEN. On the CADD[®] side there are nfigure names and properties, which should be mapped to Components-SKey combinations in ISOGEN, which then might have attributes which should get their value from CADD[®] properties.

There are some additional entries in the component-mapping file, which are used by C2IRun. Nlines that are not internal, represent pipes. These must be placed in the PCF file, but are not nfigures. To make processing possible, C2IRun adds the property COMPNAME with value NLINE to each nodal line. In this way pipes are generated in the same manner as other components.

The component mapping file is read when C2IRun starts, which means that any changes in this file require a restart of C2IRun.

A default conversion file is delivered with the installation. This file might not contain a correct mapping for your use, and generally needs to be customized.

For each component mentioned in the component mapping file there must be a component definition file (see 8.4). The name of the file is the name of the component in the file followed by the component definition extension (set with C2ISetup).

Each component identifier must be unique. If the ISOGEN[®] name for the component is the same (like BEND for two types of bends) the name of one must be made unique by adding a text after a dot. The actual name of the component will be read from the [component identifier] section from the definition file. In this way it's possible to map different CADD[®] nfigures to the same ISOGEN[®] component, with a different layout (e.g. a bend of 90 degree. has a different layout than a bend of 180 degree.).

The formatting rules for the file are as follows:

- Lines starting with '#' and empty lines are considered comment lines and are not interpreted.
 - Each line starting at position 1 starts a mapping description. This line contains first the ISOGEN® component identification, then the SKEY to be used for this mapping and then an optional description. Both identifier and SKEY must be given, but '?' is allowed for SKEY to indicate a "don't care", or that SKEY is determined in other ways. Alternatively, the SKEY can have two asterisks to indicate that at that position the current end type should be substituted.
 - Each line starting at position 5 indicates a property (optionally with an operator and value) that should be present for the CADDIS entity to map to the Isogen component. The property line has 2 or 4 entries:
 - o the property type :
 - 0 for no value
 - 1 for an integer property value
 - 2 for a double property value
 - 3 for a text property value
 - o the property name : not case-sensitive, no blanks allowed
- The following items are only present when the property type is not zero:
- o the operator: the standard operators: =, <>, <, <=, >, >=
 - o the (limit) value for the property. If it is a text property and the value should contain spaces, then the value must be enclosed by quotes (").
- It is assumed that property names are not case-sensitive.
 - If more than one property is given for a component it means that all property conditions must be met (e.g. AND relation between properties).
 - A property text value can have an asterisk to indicate that any value starting with the preceding letters is acceptable. To set a range, set the property twice, once with the lower limit and once with the upper limit.
 - If no properties are given for an ISOGEN® component, then this component is not activated, and cannot be part of the output. The file as delivered contains entries for all currently known ISOGEN® components, but many are not activated.
 - The first component for which all property descriptions are valid is used as the corresponding component. Depending on the use of properties this may require a change in sequence of the component descriptions!
 - For the pipes (which do not have COMPNAME properties), the property '3 COMPNAME=NLINE' is added by the conversion program, so this can be used to indicate the piping ISOGEN® component.
 - The first successful mapping is used. This means that one has to take care in which sequence the components are placed in the mapping file.
 - EndConnection items: These must be placed at the end of the mapping-file, and END-POSITION-NULL should be the last in the list. The definition of END-POSITION-NULL is used by the program, and should not be modified (see delivered mapping file).

Below are examples of mapping definitions:

```

BEND                                PB** All types. (Fabricated and Fittings)
  3 COMPNAME = BEND
BEND                                BE** All types. (Fabricated and Fittings)
  3 COMPNAME = E5R9
BEND.180                             BUFL (Return) All types. (Fabricated and Fittings)
  3 COMPNAME = E518
BEND-TEED.FABR                       ?    Not activated
BEND-TEED                             ?    Not activated
BOLT                                  ?    Not activated
CAP                                   KABW End Caps all types.
  3 COMPNAME = CAPP
CAP                                   KASC End Caps all types.
  3 COMPNAME = CASC
CAP                                   KASW End Caps all types.
  3 COMPNAME = CASW
CLAMP                                ?    Not activated
CONNECTOR                             ?    Not activated
CONNECTOR.WELDED                       ?    Not activated
COUPLING                              COSW   All straight coupling types.
  3 COMPNAME = CPFW
COUPLING                              COSC   All straight coupling types.
  3 COMPNAME = CPFS
CROSS                                 CRBW All Fitting types.
  3 COMPNAME = CROS
CROSS                                 CRSC All Fitting types.
  3 COMPNAME = CRSC
CROSS                                 CRSW All Fitting types.
  3 COMPNAME = CRSW
CROSS-SET-ON                           ?    Not activated
CROSS-STUB                             ?    Not activated
ELBOW-TEED.FABR                       ETBW (Fabricated) Elbow drilled Branch connection.
  3 COMPNAME = ELR9
  3 CHILDNAME = WELT

```

8.2 Property mapping

CADD5 and ISOGEN[®] do not have the same set of properties, making property mapping necessary. Properties having the same name (and meaning) do not need to be placed in the property-mapping file. All other properties that are actually used should be placed in the file, to ensure proper translation of the values.

If a BOM is to be added to the isometric drawing, ITEM-CODE and DESCRIPTION must be available as ISOGEN[®] attribute. This means that the mapping as shown in the file below should be present.

The property mapping file is read when C2IRun starts, which means that any changes in this file require a restart of C2IRun.

Example of property mapping file:

```

# Filename:  prop2iso.pconv
#
#
# Function: This file contains the mapping of cadd5 property names on
# isogen attribute names. The mapping is required by the C2I program.
#
# For the creation of the BOM, the program will look for isogen attributes
# ITEM-CODE and DESCRIPTION to find the data elements to be used.
#
# The file may contain 2 types of lines:
# - a comment line (any line starting with #)
# - a mapping line, which contains first the name of the cadd5 property,

```

```

#   a (number of) space(s), and then the name of the corresponding isogen attribute.
#
# DO NOT USE TABS IN THIS FILE!!!!!!!!!!!!
#
# Cadds5 property      Isogen attribute
# -----
# ITEMNUMB            NAME
# STOCKNO             ITEM-CODE
# SPOOL               SPOOL-IDENTIFIER
# BRAD                BEND-RADIUS
# BRAT                BEND-RADIUS
# BOLTS               BOLD-QUANTITY
# PIPENAME            MESSAGE-SQUARE
# TRIM                ANGLE
# FABRICATED          FABRICATION-ITEM
# WELDNUMB            WELD-SPEC
# CONTLINE            PIPELINE-REFERENCE
# COMPDESC            DESCRIPTION

```

8.3 End type mapping

CADDS uses end types with up to 4 characters whereas ISOGEN[®] uses end types with 2 characters and so mapping is required. If an end type is not converted, and ISOGEN[®] does not know the end type, an error will occur, or the result becomes unpredictable.

The end type mapping file is read when C2IRun starts, which means that any changes in this file require a restart of C2IRun.

Example of end type mapping file:

```

# Filename:  etype2iso.econv
#
#
# Function: This file contains the mapping of cadds5 end type names on
# isogen end type names. The mapping is required by the C2I program.
# Cadds end types not mentioned in this file will not be translated.
#
# The file may contain 2 types of lines:
# - a comment line (any line starting with #)
# - a mapping line, which contains first the name of the cadds5 end type,
#   a space, and then the name of the corresponding isogen end type.
#   To set the default end type (which is used for isogen when no end type
#   is defined in CADDS) add a line with DEFAULT as the cadds end type.
#
# ATTENTION:  DO NOT USE TABS IN THIS FILE!!!!!!!!
#
# Isogen knows the following end types:
#   BW  Butt weld end
#   CP  Compressed end
#   FL  Flanged end
#   PL  Plain end
#   SC  Screwed end
#   SW  Socket weld end
#   LN  Liner/Nut end
#   MP  Male part end
#   LC  Liner (clamped) end
#   LR  Reducing Liner/Nut end
#
# Cadds5 end type      Isogen end type
# -----
# DEFAULT              BW
# BE                   BW
# BOE                  BW
# PE                   PL
# CWF
# F100                 FL
# F15R                 FL

```

F15S	FL
:	:
:	:
G9RF	FL
GGRJ	FL
SCF	SC
SCM	MP
SO	FL
SWF	SW
TBE	SC
TE	SC
TOE	SC

8.4 Component definition file

The component definition file defines the layout of the component definition in the PCF-file. The contents of the file is documented in the 'PCF Component Reference Guide' and defines which items are obligatory in the output, and which items are optional.

There must be at least two sections in a definition file: [Component Identifier] and [Mandatory Attributes]. The first gives the name of the ISOGEN[®] component, the second the data required to process the component properly. Additional sections define attributes that can be added to the component definition. These are only added to the definition if corresponding properties are given (taking into account the property conversions).

The identifier section can contain only 1 line, the mandatory section not more than 12 lines.

Example component definition file for normal elbow:

```
[Mandatory Attributes]
END-POINT Size End-type
END-POINT Size End-type
CENTRE-POINT
SKEY
[Material Information Attributes]
ADDITIONAL-ITEM
COMPONENT-REMARK-NUMBER data
ERECTION-ITEM
FABRICATION-ITEM
OFFSHORE-ITEM
ITEM-CODE data
REPEAT-PART-NUMBER data
[Connection/Continuation Attributes]
END-CONNECTION
END-POSITION
TAP-CONNECTION
CONTINUATION
[Specification Information Attributes]
INSULATION-ON
INSULATION-SPEC data
MISC-SPEC data
PAINTING-SPEC data
PIPING-SPEC data
TRACING-ON
TRACING-SPEC data
[Supplementary Information Attributes]
ANGLE data
BOP-ELEVATION data
DETAIL-SKETCH-IDENTIFIER data
MESSAGE
NAME data
TAG data
```

```
SPECIAL-STATUS
SPOOL-IDENTIFIER data
UNIQUE-COMPONENT-IDENTIFIER data
WEIGHT data
```

Example component definition file for pipe:

```
[Component Identifier]
PIPE
[Mandatory Attributes]
END-POINT Size End-type
END-POINT Size End-type
[Material Information Attributes]
ADDITIONAL-ITEM
COMPONENT-REMARK-NUMBER data
ERECTION-ITEM
FABRICATION-ITEM
OFFSHORE-ITEM
ITEM-CODE data
REPEAT-PART-NUMBER data
[Connection/Continuation Attributes]
END-CONNECTION
END-POSITION
TAP-CONNECTION
CONTINUATION
[Specification Information Attributes]
INSULATION-ON
INSULATION-SPEC data
MISC-SPEC data
PAINTING-SPEC data
PIPING-SPEC data
TRACING-ON
TRACING-SPEC data
[Supplementary Information Attributes]
MESSAGE
PLANT-AREA data
SPECIAL-STATUS
SPOOL-IDENTIFIER data
UNIQUE-COMPONENT-IDENTIFIER data
WASTE data
WEIGHT data
```

8.5 Log file

When C2IRun is active, it will write messages to a log file, which may be used to find the source of any problems. Each time C2IRun is started a new file will be opened. So if you wish to use the file to debug, then you have to copy the file to another location before restarting C2IRun.

The lines in the log file have the following format:

```
<time stamp> <message code> <message>
```

If the message code is -1, then the message is informational.

The first line is the starting message showing the version of cadds2iso.dll, which is the actual conversion module.

Example of a logfile (one adr file processed):

```
23-1-2003 14:21:21 -1 C2IRun conversion started (cadds2iso.dll 1.5.8)
23-1-2003 14:21:21 -1 Component conversion from D:\ISOGEN\CMI\DataFiles\comp2iso.cconv
23-1-2003 14:21:21 -1 Reading .adr-files from dir D:\ISOGEN\CMI\InputDir\
23-1-2003 14:21:21 -1 -----
23-1-2003 14:21:21 -1 Starting with D:\ISOGEN\CMI\InputDir\BERTA1_ppibelw1.adr
23-1-2003 14:21:21 -1 number CNodes found = 4
23-1-2003 14:21:21 -1 number NLines found = 3
23-1-2003 14:21:21 -1 number NFigures found = 1
23-1-2003 14:21:21 -1 model to D:\ISOGEN\CMI\PCFDir\ppibelw2.pcf
23-1-2003 14:21:21 -1 PCF data written to D:\ISOGEN\CMI\PCFDir\ppibelw1.pcf
23-1-2003 14:21:21 -1 using project Severeysa and style FINAL-FAB
23-1-2003 14:21:24 -1 .DXF file(s) made
23-1-2003 14:21:24 -1 moving files to D:\ISOGEN\CMI\DXFDir\BERTA1\
23-1-2003 14:21:24 -1 File renamed to D:\ISOGEN\CMI\InputDir\BERTA1_ppibelw1.030123-142124
23-1-2003 14:21:39 -1 C2IRun stopped because stop-file found
23-1-2003 14:21:39 -1 D:\ISOGEN\CMI\InputDir\stop.adr deleted
```

9 C2ISetup

C2ISetup is the program used to modify settings for C2IRun. It can also be used to start and stop C2IRun, or to view the log file.

When started the main form will show:



The form has buttons to start the set options dialog, to start C2IRun (not active when running), to stop C2IRun, to view the log file and to terminate the setup. It also shows the status of C2IRun (not active or starting time), and the current date and time. When the set options button is clicked, the options dialog is displayed. In the following paragraphs describe the tabs of this dialog.

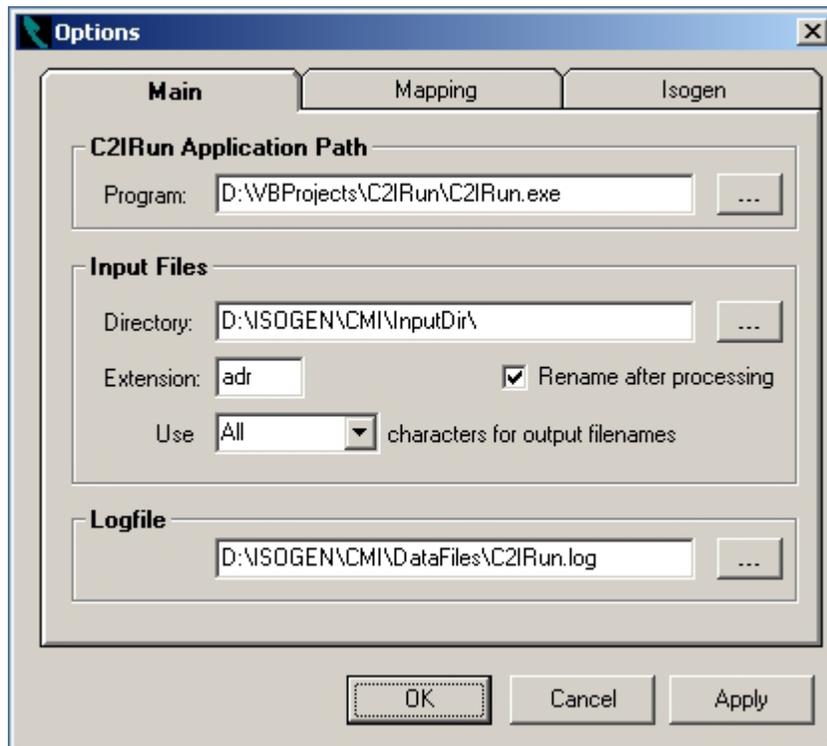
When a select button is shown behind an input box , input can also be done through standard windows file or folder selection dialogs.

To store the settings click 'OK' or 'Apply', to leave the dialog without saving values click 'Cancel'.

All setting information will be stored in the registry in the current user section. This means that each user can have individual settings.

9.1 Main

This tab contains the main processing options.



C2IRun Application Path

Program:

The full file name for C2IRun.exe. This path is used to start C2IRun.

Input Files

Directory:

Name of the directory where the input files are read from. The program will process any file with the extension given by 'Extension', when a .cmpl file with the same name is present at this directory.

Extension:

Extension of the files to be processed.

Rename after processing:

If this box is checked, input files are not deleted after processing, but renamed (extension is replaced by timestamp). This can be useful when testing, but may cause disk space problems.

Use ... characters for output file names

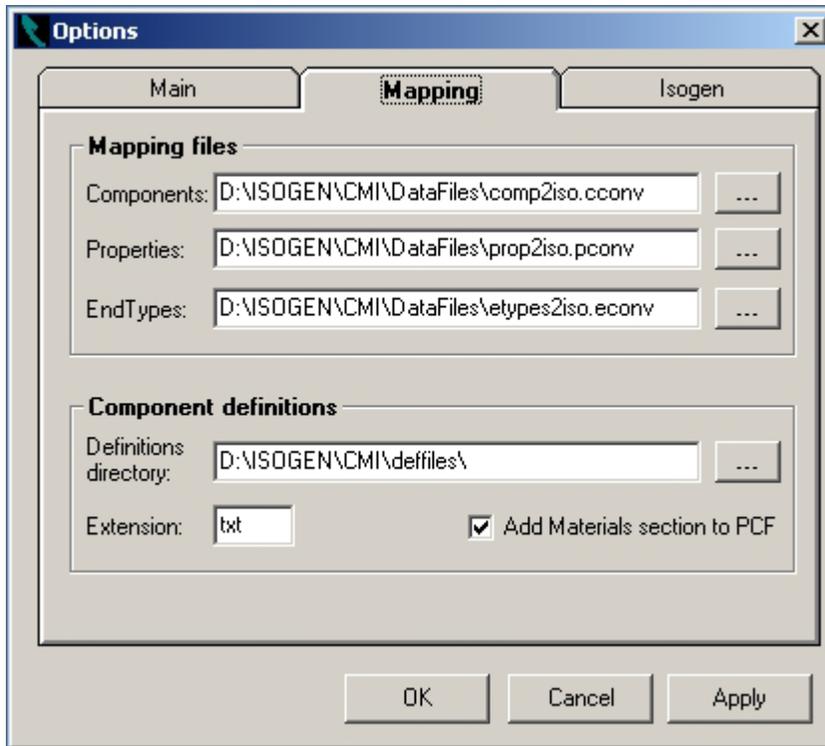
Limits the number of characters used for output file names.

Logfile

Path to the C2IRun logfile.

9.2 Mapping

In this tab one can set the names of the mapping files, and the directory for the component definition files.



Mapping files

Components

Path to the component mapping file.

Properties

Path to the property mapping file.

End Types

Path to the end type mapping file.

Component definitions

Definitions directory

Path to the directory where the component definition files are placed.

Extension

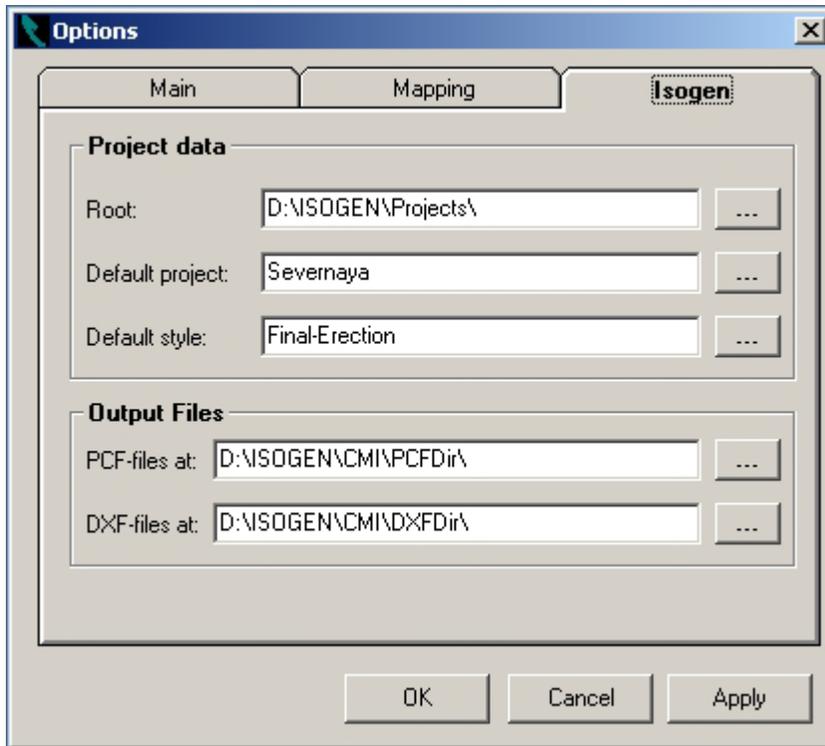
Extension of the component definition files.

Add materials section to PCF

If checked, a materials section will be added to the PCF, which can be used by ISOGEN[®] to generate a BOM.

9.3 Isogen

The settings in this tab determine the default behaviour for the type and style of drawing Isogen produces and where resulting DXF file is placed.



Project data

Root

The project root of ISOGEN[®]. This directory determines which projects and styles are available for isometric creation.

Default project

This is the name of the project that will be used when all other means to determine a project fail.

Default style

This is the name of the style that will be used when all other means to determine a style fail.

Output Files

PCF-files at:

Directory where PCF-files generated by C2Irun are placed. ISOGEN[®] will copy the input PCF files from this directory to the Project\Style\INPUTS directory before processing them.

DXF-files at:

Directory where the resulting DXF-files are placed. C2IRun will copy all DXF-files to a subdirectory of this directory, with the hostname of the client that requested the conversion. C2IRun will also place a hostname.cmpl file, which signals to C2IClient that processing is completed.

Aknowledgements

ISOGEN is the registered trademark of ICI plc.

CADDS is the registered trademark of PTC (Parametric Technology Corporation)

References

The following Alias documents are used preparing the software

PCF Format Guide

PCF Component Reference Guide

ISOGEN SKEY Definitions

Alias Limited Industrial Software Products

Stuart Road, Manor Park

Runcorn, Cheshire WA7 1TS

United Kingdom

Email: info@alias.ltd.uk