08:36:30 OCA PAD AMENDMENT - PROJECT HEADER INFORMATION 07/06/94

Active

Project #: B-10-F61 Cost share #: Rev #: 2
Center # : 10/24-6-R8042-0A0 Center shr #: OCA file #:
Work type : RES
Contract#: 0012-0003-AA Mod #: 02 Document : SUBCONT
Prime #: F09603-93-G-0012-0003 Contract entity: GTRC

Subprojects ? : N CFDA: N/A
Main project #: PE #: N/A

Project unit: OIP Unit code: 03.010.200
Project director(s):
GOURISANKAR V ENGR COLL (404)894-7772

Sponsor/division names: MERCER UNIVERSITY / ATLANTA, GA
Sponsor/division codes: 400 / 123

Award period: 940124 to 940930 (performance) 940930 (reports)

Sponsor amount	New this change	Total to date
Contract value	0.00	98,888.56
Funded	38,368.56	98,888.56
Cost sharing amount		0.00

Does subcontracting plan apply ?: N

Title: DEVELOPMENT OF CATIA CONVERSION UTILITIES.

PROJECT ADMINISTRATION DATA

OCA contact: Ina R. Lashley 894-4820

Sponsor technical contact

MR. ALAN FROST
(912)929-6400

MERCER ENGINEERING RESEARCH CENTER
1861 WATSON BLVD.
WARNER ROBINS, GA 31093-3631

Sponsor issuing office

MS. ANDREA D. LAWTON
(919)929-6400

MERCER ENGINEERING RESEARCH CENTER
1861 WATSON BOULEVARD
WARNER ROBINS, GA 31093-3631

Security class (U,C,S,TS) : U ONR resident rep. is ACO (Y/N): N
Defense priority rating : N/A N/A supplemental sheet
Equipment title vests with: Sponsor X GIT
NONE PROPOSED.
Administrative comments -
MOD 02 EXTENDS PERIOD OF PERFORMANCE TO 9/30/94 AND FUNDS "OPTION 2" FOR
$38,368.56. THIS PROJECT IS NOW FULLY FUNDED.

GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION

NOTICE OF PROJECT CLOSEOUT

Closeout Notice Date 09/30/94

Project No. B-10-F61 Center No. 10/24-6-R8042-0A0

Project Director GOURISANKAR V School/Lab OIP

Sponsor MERCER UNIVERSITY/ATLANTA, GA

Contract/Grant No. 0012-0003-AA Contract Entity GTRC

Prime Contract No. F09603-93-G-0012-0003

Title DEVELOPMENT OF CATIA CONVERSION UTILITIES.

Effective Completion Date 940930 (Performance) 940930 (Reports)

Closeout Actions Required:	Y/N	Date Submitted
Final Invoice or Copy of Final Invoice	Y	
Final Report of Inventions and/or Subcontracts	Y	
Government Property Inventory & Related Certificate	Y	
Classified Material Certificate	N	
Release and Assignment	N	
Other	N	

Comments

Subproject Under Main Project No.

Continues Project No.

Distribution Required:

Project Director	Y
Administrative Network Representative	Y
GTRI Accounting/Grants and Contracts	Y
Procurement/Supply Services	Y
Research Property Managment	Y
Research Security Services	N
Reports Coordinator (OCA)	Y
GTRC	Y
Project File	Y
Other	N
	N

NOTE: Final Patent Questionnaire sent to PDPI.

Georgia Tech CALS Technology Center
Georgia Institute of Technology
Atlanta, Georgia 30332-0140
(404) 894-7772 phone
(404) 894-9812 fax

Technology Development *Education* *Technology Transfer*

MARCH 28th, 1994.

Alan Frost / Adrea Lawton
Mercer Engineering Research Center
1861 Watson Boulevard
Warner Robins, GA 31093

RE: CDRL A0001 for subcontract #0012-0003-AA

Alan Frost / Andrea Lawton:

Please find, enclosed, the final copy of the Engineering Report that details the results of the investigation into the CATIA IGES Processor. The suggestions resulting from your review of the first draft have been incorporated in the final document.

We have enclosed an electronic copy of the report as per the contractual requirement.

The base portion of the subcontract has been successfully completed, the results of which are documented in the aforementioned report.

We request that the first option, totaling $20,965.00 be exercised. Please do not hesitate to contact me if you have any questions.

Sincerely,

V. Gourisankar, Asst Director
CALS Technology Center

cc: R.E. Fulton, Director

DEVELOPMENT OF CATIA CONVERSION UTILITIES

Sub contract # 0012-0003-AA

CDRL A0001

Engineering Report
Results of CATIA IGES Processor Investigation

March 28th, 1994

Prepared by

CALS Technology Center

Georgia Institute of Technology

CATIA VERSION 3.0 RELEASE 2.5

INTRODUCTION

CATIA stands for Computer Aided Three-dimensional Interactive Application. CATIA is a CAD/CAM system which operates in two different environments:

* IBM System/370 computers with MVS and VM operating systems.
* IBM RISC System/6000 computers with AIX operating system.

The WR-ALC Product Data Conversion Facility hosts CATIA on an IBM RISC System/6000 computer.

COMPOSITION

CATIA is made up of numerous products. The products can be grouped into seven major families based on the intended use as shown below:

FAMILY	PRODUCTS
CATIA CORE	Base Library
DATA EXCHANGE	Interface Data Management Access
GEOMETRIC MODELING	Drafting 3D Design Advanced Surface Solids Geometry
ANALYSIS	Kinematics Image Design Finite Element Modeler
CAM	Numerical Control - Lathe Numerical Control - Mill
CAE	Piping & Tubing Building Des. & Facil. Layout Structural Design & Steelwork Schematics
PROGRAMMING	Graphics Interactive Interface Geometry Interface (CATGEO) Math Subrout. Package (CATMSP) Interactive User Access (IUA)

Geometry creation in CATIA is supported through the Base, Drafting, 3D Design, Advanced Surfaces, and Solids Geometry

products. The IGES based transfer of geometric elements created with all of the aforementioned products, with the exception of the Drafting product, has been investigated and the results documented in this report. We did not conduct any tests on the Drafting product.

GEOMETRIC MODELING IN CATIA

Geometric modeling in CATIA is facilitated through the creation and management of the following elements/concepts:

* Geometric Elements (Wireframes, Surfaces, Solids)
* Dress-up Elements (Patterns, Dimensions, Texts & Notes, Fonts)
* Data Structure (Sets, Layers, Details & Dittos)
* Models (Named Collection of Geometric Elements)

CATIA supports the traditional bottoms-up 3-D modeling. The user starts with the wire frame representation of an object progressing to surface and then solid representations. 3-D geometry can be defined precisely (canonical or polynomial) or approximated (faceted polyhedra). In addition to the traditional bottoms-up approach, CATIA supports Constructive Solids Geometry also.

The user can create geometry in two different modes inside CATIA: SPACE mode and DRAW mode. In the SPACE mode, 3-D geometric entities that make up product geometry are created directly in 3-D space. In the DRAW mode, the projections and mappings of product geometry onto background planes are created, annotated and laid out. The IGES based transfer of SPACE elements have been investigated and the results documented in this report. We did not conduct any tests on the DRAW elements.

We did not conduct any tests on the IGES based behavior of geometric models that have un-resolved external references such as details & dittos and the use of part libraries.

CATIA USE

The interactive user of CATIA creates geometry through the use of interactive functions and interactive menus. When the function is selected, the corresponding menu is displayed. Selection of a menu item results in an ordered sequence of interactions.

CATIA IGES PROCESSOR

The CATIA-IGES interface that is used to convert CATIA-coded data into IGES coded data and vice versa has two phases, namely, a data editing phase and a data exchange phase. CATIA uses a format known as IBM IGES Format (IIF) as an intermediate format for conversion between CATIA and IGES.

IIF consists of both a file structure as well as a library of routines used for reading and writing the IIF file. The physical format of an IIF file is different from that of an IGES file, but both files are logically equivalent. The IIF file is a binary file designed for efficient access to data and contains the same information as the IGES file. In an IGES file, the DE records and the PD records are physically distant from each other, while they are kept close to each other in an IIF file, thereby reducing input/output thrashing.

The data exchange phase of the CATIA-IGES interface deals with the conversion between a CATIA model and an IIF file. The data editing phase is concerned with the editing and reformatting of an IIF file into an IGES file or vice versa.

The data editing phase of this exchange can be handled using either a command line interface utility, igesp, or a graphical interface utility, igesm. Both versions allow the user to translate a file from IGES to IIF or IIF to IGES. In addition, a user-oriented printout of an IGES file or IIF file, which makes reading the contents of an IGES or IIF file easier for the user, can also be produced. When igesm is used, panels are presented where the user can select the options and carry out the translation. The command line version is invoked as

```
igesp -flag [-f optionsfile] infile [outfile]
```

where
- flag is
  - -i for translation from IGES to IIF
  - -c for translation from IIF to IGES
  - -p for user-oriented printout from IIF
  - (or IGES, if IIF file is absent)
- optionsfile is
  - the name of the file from which the translation parameters are to be read. (defaults to igesinp.data).
- infile is
  - the name of the input file (IIF or IGES).
- outfile is
  - the name of the output file (IIF or IGES).
  - (by default the infile name is used to create the outfile name)

There are two separate utilities for the data exchange, depending on the direction of conversion. For CATIA to IIF, the utility to be used is named, catige. For IIF to CATIA translation, the utility to use is called igecat.

Catige can be invoked from the command line as:

```
catutil -l catige -x catige.opt -o logfile
```

where catige.opt is the name of a file containing the options to

be used for the CATIA to IIF conversion. If this is left out, the panels for choosing these options is displayed from which the options can be selected.

If the logfile name is not specified using the -o option, the default logfile name catige.out is used. This file lists the no. of CATIA elements and the no. of IGES elements created. It also identifies the CATIA elements that were not successfully translated into IGES.

Igecat can be invoked from the command line as:

```
catutil -l igecat -x igecat.opt -o logfile
```

where igecat.opt is the name of a file containing the options to be used for the IIF to CATIA conversion. If this is left out, the panels for choosing these options is displayed from which the options can be selected. If the logfile name is not specified using the -o option, the default logfile name igecat.out is used. This file lists the no. of IGES elements and the no. of CATIA elements created. It also identifies the IGES elements that were not successfully translated into CATIA.

ELEMENT BY ELEMENT INVESTIGATION

The results of the CATIA-IGES conversion element by element investigation are presented in Appendix A. Some general comments about the results are presented here.

A common feature observed in the CATIA to IGES conversion is that CATIA always outputs an IGES drawing element, #404, an IGES view element, #410, an IGES planar associativity element, #402 form 16, an IGES group element, #402 form 7, and the associated property elements, #406. However, the CATIA geometry elements are not part of (dependent on) the #404 drawing element. This cannot be avoided.

Another feature that is common to CATIA generated IGES files is the presence of implementer-defined entity types for property elements, #406. These have form numbers above 5000. It is possible to drop these implementer-defined from the output IGES file during the translation from IIF to IGES by setting the option, IBMDE to NO. This results in an IGES file with null entities taking the place of these implementer-defined entities.

CATIA IGES processor does not translate CATIA CSG elements or polyhedral volume elements. These elements, if present, are always dropped during the CATIA to IIF translation phase. In order to transfer the geometry of CSG elements to IGES, it is necessary to first convert the solid elements into their corresponding surface representation inside CATIA. The surface representation can then be transferred to IGES.

LOOP TESTING

Single element IGES files and the simple and complex model IGES files were read back into CATIA. This testing is a preliminary examination of the import capability of the CATIA IGES translator. All single wireframe elements were imported without error. All single surface elements were imported without error with the exception of the skin and net elements. These elements were dropped in CATIA to IGES transfer and could not be loop tested. All single solid elements were converted to surface representations before transfer from CATIA to IGES. The resulting IGES files were imported to CATIA without error.

CATIA GEOMETRY MIGRATION

CATIA classifies geometric elements as one of three different types: wireframes, surfaces, and solids. A primitive solid can be broken into its constituent surfaces and wireframe boundaries by the use of the SOLID2 / SURFACE command. The resulting surface still retains volume information via the polyhedral volume element (CATIA element # 17). This polyhedral volume is not output to the IGES file. If this operation is performed on a non-primitive solid, the resulting surfaces will be the surface representations of the underlying solid primitives used to construct the complex solid. Another method of breaking solids into their surface representations is the use of the CATSOL utility. This utility will break a complex solid into its surface representation without reverting to the surface representations of the underlying primitive solids. To obtain the wireframe representation of a solid, the CATMOD utility SOLID / POLEDGE function may be used. It is also possible to migrate in the opposite direction. Wireframe elements can be used to create surfaces. Surfaces can then be used to create solids. The user must identify the faces that will form the closed volume by executing the LIMIT2 / VOLUME command. Then, the user will use the SOLID / CREATE / COMPLEX / VOLUME to create a solid out of the specified volume. It is not possible to go from wireframe to solid in one single command. It is not possible to migrate from one geometric form to another via the IGES translator.

EXCHANGE WITH I/EMS & CADDS 4X

A tabulation of the results of the study of data exchange with I/EMS and CADDS 4X can be found in Appendix B.

CATIA IGES Export to Other CAD Systems

Fifteen models from CATIA have been exported to I/EMS and CADDS 4X to determine peculiarities in the CATIA IGES export capability that were not evident from loop testing. Most of these test files consisted of surface elements and solid elements that had been modified and converted to their surface representations.

All IGES transfers described are direct translations. No previously developed PDCSF flavorings were applied unless specifically stated.

Translation from CATIA to IGES was accomplished without geometry loss. When a solid element has been created in CATIA, even after conversion to a surface representation, CATIA volume elements will exist in the CATIA model file. These CATIA volume elements (CATIA element # 7 & 17) are not supported by the CATIA IGES translator. This does not result in geometry loss.

CATIA to I/EMS

When these files were read into I/EMS, one error was encountered.

The error encountered was specific to one case. The test file in question is a solid sphere that was converted to its surface representation inside CATIA. The error issued by the I/EMS IGES processor claimed that it could not fix inconsistencies in this entity and that it could not convert it to b-spline. This is most likely due to the difficulties in modeling spherical elements (closed in both parametric directions.)

CATIA to CADDS 4X

The CATIA IGES file was validated before import to CADDS 4X. Import of this IGES file into CADDS 4X resulted in one warning and one error.

The warning issued is in reference to trimmed surfaces. The translator issues an alert if the trimming boundary is the same as the natural boundary of the base surface. This does not result in loss.

The error issued is the same error issued in I/EMS for the spherical surface.

CATIA IGES Import from Other CAD Systems

Several models from Intergraph / Engineering Modelling System (I/EMS) and ComputerVision / Computer Aided Design and Drafting (CADDS 4X) have been imported to CATIA to determine peculiarities in the CATIA IGES import capability that were not evident from loop testing. No previously developed PDCSF flavorings were applied unless specifically stated.

I/EMS to CATIA

Eight I/EMS models were converted to IGES for testing. All of these models are from the C-130 project. Translation from I/EMS to IGES was accomplished without geometry loss for all models.

When these files were read into CATIA, there were two warnings and five errors.

The CATIA IGES processor issued a warning that the IGES version number given was not supported by the translator. The translator also said that the IGES maximum line thickness was out of CATIA range. Neither of these two warnings should result in losses or affect the processing of the IGES file.

The first error was that the list of elements given for the composite curve was invalid. These composite curves are all bounding edges for cylindrical surfaces.

The second error claimed that a curve software routine resulted in a non-zero return code - degenerate element / impossible geometry. Further investigation of these elements shows that the I/EMS IGES translator has written a 126 element (B-spline curve) out for a point. This has been determined to be a random error.

The third error was that one or more constituent elements of a composite curve were in error. This is a consequence of the second error.

The fourth error involved IGES element 212 (General Note). The I/EMS IGES translator wrote out some null 212 elements in the Parameter Data section of one of the IGES files. This error is eliminated in version 2.2 of the I/EMS IGES translator.

The fifth error is an insufficient number of points for a copious data element. This is most likely a random error, since it does not appear in any other file and it only occurs once in that particular file.

CADDS 4X to CATIA

Five CADDS 4X files were used for testing.

Translation from CADDS 4X to IGES was accomplished without geometry loss.

Translation from IGES to CATIA was accomplished without geometry loss. The CATIA IGES processor issued a warning that the IGES version number given was not supported by the translator. The translator also said that the IGES maximum line thickness was out of CATIA range. Neither of these two errors resulted in losses or affected the processing of the IGES file.

GENERAL CATIA PROGRAMMING

Customized features can be programmed into the CATIA system through the use of the following products: IUA, GII, CATGEO, and CATMSP.

IUA provides a programming language and a mechanism to develop and debug IUA procedures. IUA also provides a series of commands (built in procedures) with each CATIA products. The IUA procedures and commands can be used from within the CATIA interactive environment. It should be noted that the scope of IUA commands is limited. Each menu item in a CATIA function is not supported by an IUA command. This makes the automation of an interactive sequence, within CATIA, cumbersome.

Custom IUA procedures can be developed that will facilitate the automation of an interactive sequence. However, custom IUA procedures - especially those that manipulate the geometric DB - require the use of the CATGEO product. Using the CATGEO product involves the development of structured FORTRAN programs that use the library of subroutines provided with it.

An IUA procedure or command can be executed from within the CATIA interactive environment by typing in the name at the command line (appropriate syntax is needed).

The GIG product enables the sophisticated user to develop a customized CATIA function. The use of GII involves three steps:

Design of the function (menu & dialog)
Develop the task (FORTRAN routines)
Link & Integrate into CATIA.

The quickest way to program CATIA is to use IUA product. The GIG product enables sophisticated applications, but is more involved and time consuming.

CATIA can be used in the batch mode through the CATGEO product.

PDCSF CATIA PROGRAMMING

Unlike I/EMS (part programming language) or CADDS (CV MAC), CATIA does not provide for an automation script. The IUA, though a formal language, is limited in scope as a script for automating an interactive sequence of operations.

The PDCSF assisted conversion does require the automation of interactive sequences. It is our intent to structure CATIA assisted conversion similar to that of other CAD systems in PDCSF. We will attempt to use the IUA facility to the extent possible to develop the assisted conversion. The use of CATGEO will be minimized to keep the level of effort within the scope of the contract. We believe that the use of the GII product to incorporate assisted conversion as a CATIA function is well outside the scope of the current contract and, therefore, will not be attempted.

In summation, the assisted conversion from CATIA will be

attempted in this manner:

a) Use AIX scripting for CATIA start_up.
b) Use IUA feature as much as possible.
c) Use instructions & user interactions for situations where IUA is not sufficient.
d) Use AIX scripts for starting IGES/Works.
e) Use X-Windows for concurrent displays.
e) Avoid the use of CATGEO to the fullest possible extent.
CATGEO will be the last resort.
f) Do not attempt to use CATGII.

The PDCSF direct conversion does require the use of CATIA product(s) in the batch mode. The data exchange product can be used in the batch mode by automating the catige and igecat utilities from within an AIX script.

PDCSF/CATIA FILE MANAGEMENT REQUIREMENTS

In this version of CATIA, hosted in the AIX O.S., models are stored on the hard drive as explained below:

A CATIA 'model' is a named collection of CATIA geometry entities.

The CATIA system allows for what are known as 'model files'. A model file can contain one or more models (members). A CATIA user can select a 'model file' and one of the model members via the FILE function in CATIA.

Within AIX, a CATIA 'model file' corresponds to a directory. All CATIA models that are members of this model file are stored under this directory. This directory has to be identified to the CATIA system as a valid storage area ('model file') through an entry in CATIA set up files.

One CATIA model does not correspond to one AIX file. One CATIA model corresponds to one AIX file and entries in another AIX file that serves as an index. The name of the index file is "map." Other files are usually named fic0, fic1, fic2, etc. Therefore, under an AIX directory that corresponds to a CATIA model file, there are one or more fic* files and one map file. The map file is not fully human readable. The human readable portion of the map file can be displayed with the AIX strings command.

A CATIA model can be copied from one 'model file' to another 'model file' using the FILE function within the CATIA application. Outside of the CATIA application, the same copy can be done via the CATIA utility. Never attempt to use the AIX cp command to perform the copying on the hard drive.

HOW TO PUT A CATIA MODEL INTO PDCSF ?

Assume that a CATIA user (at WR-ALC) has a CATIA model, stored under a CATIA 'model file,' that needs to be PUT in to the PDCSF repository. It is common practice among the WR-ALC to use the part number to name the CAD model of a part.

Inside the CATIA system:

Name of the CATIA 'model file':	models_1
Name of the CATIA model:	x9282830

Inside the AIX O.S.:

Name of the directory:	$HOME/models_1
Files in the models_1 dir:	fic0
	fic1
	fic2
	.....
	map

One fic* file along with some entries in the map file constitute the CATIA model x9282830. The map file links the fic* file with the CATIA model name x9282830.

STEP 1: Create a directory $HOME/x9282830.

STEP 2: Identify $HOME/x9282830 as a valid 'model file' to the CATIA application.

STEP 3: The CATIA model x9282830, resident in models_1, is copied to the pre-defined 'model file' x9282830. This is done using the CATIA utility outside of the CATIA application.

NOTE: One or more CATIA models can be copied this way.

STEP 4: Now, on the hard drive, we have:

$HOME/x9282830/fic0 and
$HOME/x9282830/map

The PDCSF PUT function is invoked. The part number, the type of data, and tool type are assigned. The directory x9282830 and all its contents are then transferred to the PDCSF repository and cataloged against the part number. From this point onwards, it is possible to browse the PDCSF repository and look for all CATIA models cataloged against that particular part number. In this case, PDCSF repository will list x9282830 as an available CATIA model.

In other words, in the PDCSF repository, a named CATIA model

appears as a directory of the same name with fic0 and map under it. There can be as many CATIA models as needed stored against a part number.

HOW TO GET A CATIA MODEL FROM PDCSF ?

We have established that inside the PDCSF repository a named CATIA model appears as a directory of the same name with fic0 and map under it.

STEP 1: The part number tree is browsed and the appropriate part number selected.

STEP 2: The GET function is invoked. The named CATIA data set is selected (x9282830). GET function is executed.

STEP 3: A directory (x9282830) is created on the local file system and the fic0, map files are deposited under it.

Please note that before the CATIA model named x9282830 can be used by the CATIA application one of two things have to happen:

a) The directory x9282830 is validated as a 'model file' and then read by the CATIA application.
b) The CATIA model x9282830 is copied to another 'model file' and used by the CATIA application.

ASSISTED CONVERSION REQUIREMENTS

Let us assume that a named CATIA model is available on the transparent file system. (Through a GET or CATIA utility or CATIA FILE function.)

CATIA model name:	x9282830
Transparent FS:	$HOME/pdcsf/convert/catia/x9282830
	$HOME/pdcsf/convert/catia/x9282830/fic0
	$HOME/pdcsf/convert/catia/x9282830/map

STEP 1: Run the "org_file x9282830" command from the $HOME/pdcsf/convert/catia directory.

The $HOME/pdcsf/convert/catia/x9282830 directory will be identified to the CATIA application as a valid 'model file.'

STEP 2: Run the "start_catia x9282830" command from the $HOME/pdcsf/convert/catia directory.
The following sub-steps are accomplished.

The CATIA application is launched. The user is prompted to read the 'appropriate model file' and the model. Other steps in assisted conversion are pursued.

NOTE 1: The CATIA 'model file' x9282830 remains a valid storage area until it is removed explicitly as part of the cleanup utility.

NOTE 2: The start_catia utility would also have the third parameter that specifies the phase of conversion.

DIRECT CONVERSION REQUIREMENTS

Let us assume that several CATIA models need to be converted to IGES files. Assume that:

The $HOME/convert/catia/batchwrite directory has been created. Batchwrite directory has already been validated as a CATIA model file. Both batchwrite & batchread/native are permanently configured as valid CATIA model files.

Convert CATIA to IGES

STEP 1: Copy as many CATIA models as needed into batchwrite.
This has to be done via the CATIA utility.
The batchwrite directory will contain fic* files and the map file.

STEP 2: Execute the conv_to_iges utility.

Convert IGES to CATIA

STEP 1: Copy as many IGES files as needed into batchread.
The batchread directory will contain only IGES files.

STEP 2: Execute the conv_to_native utility.
The batchread/native directory will contain the fic* files and the map file resulting from the conversion to the native format.

SUMMARY

The Georgia Tech project team has conducted a detailed investigation of the behavior of the CATIA CAD system and the CATIA IGES translator as specified in the Statement of Work. Specifically, the following tasks were performed. An element by element analysis was done by creating every possible element by every possible creation method. A study of the effects of migration between wireframe, surface, and solid geometric representations was conducted, also. Several simple and complex models were created and exported to IGES. The single element IGES files and simple and complex model IGES files were then read back into CATIA as a preliminary test of the CATIA IGES import capability. Detailed results of the element by element analysis and loop testing can be found in Appendix A.

Several IGES files were selected from Intergraph I/EMS and ComputerVision CADDS 4X for import to CATIA. Several CATIA IGES files were selected for export to I/EMS and CADDS 4X. The results of the exchange of data between these three CAD systems can be found in Appendix B. Finally, the preferred internal mathematical representation of CATIA was determined.

In the opinion of the Georgia Tech project team, CATIA's compliance with the IGES standard is as follows. CATIA does support all wireframe and surface elements in the standard. Solid elements are not supported. When a solid element is written to an IGES file, all geometry is lost. In the experience of this team, virtually no CAD system can outpot the precise surfaces and topological information (B-rep elements) required to describe a solid in IGES. The flavoring of the mathematical representation of an IGES file has no effect on the IGES import capability. It is possible, inside CATIA, to change from a solid representation to a surface representation.

There are two points of concern raised by the investigation. The tolerance values specified in the IGES translator options files have a significant effect on whether geometry can be accepted into CATIA or whether the geometry will be acceptable to other CAD systems upon export from CATIA. The Georgia Tech project team will determine the optimum values for these tolerances for our conversion utilities. Another point is the change in units from metric to fps units. CATIA uses metric units as a default. All other CAD systems in the PDCSF area use fps units. It has been decided that CATIA model files will be in inches. CATIA files will be converted to inches before export to IGES and incoming IGES files will be read in in inches. The Georgia Tech project team will monitor the utilities to be certain that the unit changes are accomplished accurately.

Data exchange with other CAD systems proceeds without error for a majority of cases. Some errors have been detected, mostly I/EMS to CATIA transfers. The majority of these errors have been determined to be random errors. In the event that a predictable error is found, every attempt will be made to automatically compensate for loss.

Appendix A.

Behavior of the CATIA IGES Export Translator

CATIA Wireframe Geometry Entities

IGES output using STANDARD option for conics

	Operation	Native Elements	IGES Elements	Loop Test
Arc	ARC/CREATE	1-CST 1-CRV	1-106F3 1-126F0	Yes
Point	POINT	1-PT	1-116	Yes
Circle	LINE	1-LN	1-110	Yes
Circle	CURVE2/CIRCLE	1-CRV	1-100	Yes
Ellipse	CURVE2/ELLIPSE	1-CRV	1-104F1	Yes
Conic	CURVE2/CONIC	1-CRV	1-104F1	Yes
Spline	SPLINE/COMPUTE	1-CST 1-CRV	1-106F2 1-126F0	Yes

Behavior of the CATIA IGES Export Translator

CATIA Wireframe Geometry Entities

IGES output using B-SPLINE option for conics

	Operation	Native Elements	IGES Elements	Loop Test
Arc	ARC/CREATE	1-CST 1-CRV	1-106F3 1-126F0	Yes
Point	POINT	1-PT	1-116	Yes
Circle	LINE	1-LN	1-110	Yes
Circle	CURVE2/CIRCLE	1-CRV	1-100	Yes
Ellipse	CURVE2/ELLIPSE	1-CRV	1-126F3	Yes
Conic	CURVE2/CONIC	1-CRV	1-126F3	Yes
Spline	SPLINE/COMPUTE	1-CST 1-CRV	1-106F2 1-126F0	Yes

Behavior of the CATIA IGES Export Translator

CATIA Wireframe Geometry Entities

	Operation	Native Elements	IGES Elements	Loop Test
Curve	CURVE1/INTERSEC	1-CRV	1-126F0	Yes
Curve	CURVE1/ISOPARAM	1-CRV	1-126F0	Yes
Curve	CURVE1/BOUNDARY	1-CRV	1-126F0	Yes
Curve	CURVE1/PROJECT	1-CRV	1-126F0	Yes
Curve	CURVE1/PT-PT	1-CRV	1-126F0	Yes
Curve	CURVE1/OFFSET	1-CRV	1-126F0	Yes

Behavior of the CATIA IGES Export Translator

CATIA Wireframe Geometry Entities

	Operation	Native Elements	IGES Elements	Loop Test
Curve	LIMIT1/RELIMIT	1-CRV	1-126F0	Yes
Curve	LIMIT1/CORNER	1-CRV	1-100	Yes
Curve	LIMIT1/MACHINE	1-CRV	1-110	Yes
Curve	LIMIT1/BREAK	2-CRV	2-126F0	Yes
Curve	LIMIT1/CONCATEN	1-CRV	1-126F0	Yes
Curve	LIMIT1/EXTRAPOL	1-CRV	1-126F0	Yes

Behavior of the CATIA IGES Export Translator

CATIA Surface Geometry Entities

IGES export using Standard option for canonical surfaces

	Creation Method	Native Elements	IGES Elements	Loop Test
Plane	Plane	1-PLN	1-108 Plane	Yes
Cone	Surf1/Revolutn/Cone	1-SUR	2-110 Line 1-120 Surf of rev	Yes
Cylinder	Surf1/Revolutn/Cyl	1-SUR	2-110 Line 1-120 Surf of rev	Yes
Sphere	Surf1/Revolutn/Sph	1-SUR	1-100 Circular arc 1-110 Line 1-120 Surf of rev	Yes
Torus	Surf1/Revolutn/Torus	1-SUR	1-100 Circular arc 1-110 Line 1-120 Surf of rev	Yes
Ruled Surface	Surf1/Cylinder	1-SUR	1-128 F 7 B-Spline surf	Yes
Offset Surface	Surf1/Cylinder	1-SUR	1-128 F 7 B-Spline surf	Yes
Pipe Surface	Surf1/Pipe	1-SUR	1-128 F 7 B-Spline surf	Yes

Behavior of the CATIA IGES Export Translator

CATIA Surface Geometry Entities

IGES output using B-Spline option for canonical surfaces

	Creation Method	Native Elements	IGES Elements	Loop Test
Plane	Plane	1-PLN	1-108 Plane	Yes
Cone	Surf1/Revolutn/Cone	1-SUR	1-128 F3 B-Spline surf	Yes
Cylinder	Surf1/Revolutn/Cyl	1-SUR	1-128 F6 B-Spline surf	Yes
Sphere	Surf1/Revolutn/Sph	1-SUR	1-128 F4 B-Spline surf	Yes
Torus	Surf1/Revolutn/Torus	1-SUR	1-128 F5 B-Spline surf	Yes
Ruled Surface	Surf1/Cylinder	1-SUR	1-128 Form 7	Yes
Offset Surface	Surf1/Cylinder	1-SUR	1-128 Form 7	Yes

Behavior of the CATIA IGES Export Translator

CATIA Surface Geometry Entities

	Operation	Native Elements	IGES Elements	Loop Test
Face -Planar	LIMIT2/FACE/ CREATE/PLANE	1-SUR 1-FAC	2-102 8-126 1-128F7 1-142 1-144 1-212	Yes
Face -Planar	LIMIT2/FACE/ BREAK	2-SUR 2-FAC	4-102 16-126 2-128F7 2-142 2-144 2-212	Yes
Face -Planar	LIMIT2/FACE/ UNION	1-SUR 1-FAC	2-102 12-126 1-128F7 1-142 1-144 2-212	Yes

Behavior of the CATIA IGES Export Translator

CATIA Surface Geometry Entities

	Operation	Native Elements	IGES Elements	Loop Test
Face -Non-Planar	LIMIT2/FACE/ CREATE/SURFACE	1-SUR 1-FAC 2-LN 2-CRV	2-102 4-110 9-126 1-128F7 1-142 1-144 1-212	Yes
Face -Non-Planar	LIMIT2/FACE/ BREAK	1-SUR 2-FAC 5-LN 3-CRV	4-102 7-110 17-126 1-128F7 2-142 2-144 2-212	Yes
Face -Non-Planar	LIMIT2/FACE/ UNION	1-SUR 1-FAC 5-LN 18-CRV	2-102 12-110 23-126 1-128F7 1-142 1-144 1-212	Yes

Behavior of the CATIA IGES Export Translator

CATIA Surface Geometry Entities

	Operation	Native Elements	IGES Elements	Loop Test
Face -Non-Planar	LIMIT2/FACE/ INTERSEC	1-SUR 1-FAC 3-LN 9-CRV	1-102 7-110 9-126 1-128F7 1-142 1-144 1-212	Yes
Face -Non-Planar	LIMIT2/FACE/ SUBTRACT	1-SUR 2-FAC 1-LN 5-CRV	4-102 3-110 21-126 1-128F7 3-142 2-144 2-212	Yes
Face -Non-Planar	LIMIT2/FACE/ DIVIDE	1-SUR 3-FAC 1-LN 6-CRV	1-102 1-110 22-126 1-128F7 3-142 3-144 3-212	Yes

Behavior of the CATIA IGES Export Translator

CATIA Surface Geometry Entities

	Operation	Native Elements	IGES Elements	Loop Test
Surface	LIMIT2/SURFACE/ BREAK	1-LN 2-SUR	1-110 2-128F7	Yes
Surface	LIMIT2/SURFACE/ CONCATEN	1-LN 1-SUR	1-110 1-128F7	Yes
Surface	LIMIT2/SURFACE/ EXTRAPOL	1-LN 1-SUR	1-110 1-128F7	Yes

Behavior of the CATIA IGES Export Translator

CATIA Surface Geometry Entities

	Operation	Native Elements	IGES Elements	Loop Test
Skin	LIMIT2/SKIN/	5-LN	4-102	No
	CREATE	3-CRV	7-110	
		1-SUR	17-126	
		2-FAC	1-128F7	
		1-SKIN	2-142	
			2-144	
			2-212	
Skin	LIMIT2/SKIN/	5-LN	4-102	No
	OFFSET	3-CRV	7-110	
		1-SUR	17-126	
		2-FAC	1-128F7	
		1-SKIN	2-142	
			2-144	
			2-212	
Skin	LIMIT2/SKIN/	5-LN	2-102	No
	MODIFY	3-CRV	6-110	
		1-SUR	10-126	
		1-FAC	1-128F7	
		1-SKIN	1-142	
			1-144	
			1-212	

Note: All skin type CATIA elements are dropped in CATIA to IGES transfer.

CATIA Surface Geometry Entities

	Operation	Native Elements	IGES Elements	Loop Test
Net	NET1/CREATE	1-LN 2-CRV 1-NET	1-110 2-126	No
Net	NET1/DEFORM NODE	1-LN 2-CRV 1-NET	1-110 2-126	No
Net	NET1/DEFORM ISOPAR	1-LN 2-CRV 1-NET	1-110 2-126	No
Net	NET1/SMOOTH NET	1-LN 2-CRV 1-NET	1-110 2-126	No
Net	NET1/APPROXIM	1-LN 2-CRV 1-NET	1-110 2-126	No
Net	NET1/SURFACE	1-LN 2-CRV 1-SUR 1-NET	1-110 2-126 2-128F0	No

Note: All CATIA Net type elements are dropped in CATIA to IGES transfer.

Behavior of the CATIA IGES Export Translator

CATIA Solid Geometry Entities

IGES output after conversion to surfaces using CATSOL

	Creation Method	Native Elements	IGES Elements	Loop Test
Cone	Solid/Cone	3-LN 4-SUR 4-CRV 4-FAC	6-100 8-102 7-110 21-126 2-128F3 2-128F7 4-142 4-144 4-212	Yes
Cuboid	Solid/Cuboid	6-SUR 6-FAC 14-LN	24-126 14-110 10-102 6-128F7 24-142 6-144 6-212	Yes

Note: In all these cases a polyhedral volume element in CATIA is dropped.

Behavior of the CATIA IGES Export Translator

CATIA Solid Geometry Entities

IGES output after conversion to surfaces using CATSOL

	Creation Method	Native Elements	IGES Elements	Loop Test
Cylinder	Solid/Cylinder	2-LN 4-SUR 4-FAC 4-CRV	6-100 8-102 4-110 20-126 2-128F2 2-128F7 4-144 4-142 4-212	Yes
Pyramid	Solid/Pyramid	9-LN 5-SUR 5-FAC	10-102 9-110 32-126 5-142 4-144 5-128F7 4-212	Yes

Note: In all these cases a polyhedral volume element in CATIA is dropped.

Behavior of the CATIA IGES Export Translator

CATIA Solid Geometry Entities

IGES output after conversion to surfaces using CATSOL

	Creation Method	Native Elements	IGES Elements	Loop Test
Sphere	Solid/Sphere	1-LN 4-SUR 4-FAC 4-CRV	12-100 8-102 1-110 8-126 4-142 4-144 4-128F4 4-212	Yes
Torus	Solid/Torus	1-LN 4-SUR 4-FAC 8-CRV	15-100 8-102 1-110 25-126 4-128 F4 4-144 4-142, 4-212	Yes

Note: In all these cases a polyhedral volume element in CATIA is dropped.

CATIA Solid Geometry Entities

	Operation	Native Elements	IGES Elements	Loop Test
Solid	SOLID/OPERATN/ UNION	20-LN 13-SUR 19-FAC 1-VOL 16-CRV	22-110 20-100 48-102 9-104F2 129-126 2-128F3 11-128F7 24-142 19-144 19-212	Yes
Solid	SOLID/OPERATN/ INTERSCT	7-LN 8-SUR 8-FAC 1-VOL 12-CRV	8-110 12-100 16-102 9-104F2 62-126 2-128F3 6-128F7 8-142 8-144 8-212	Yes

Note: Solid elements do not transfer into IGES. Solids are first converted to surfaces before output to IGES.
The volume element is dropped in each case.

Behavior of the CATIA IGES Export Translator

CATIA Solid Geometry Entities

	Operation	Native Elements	IGES Elements	Loop Test
Solid	SOLID/OPERATN/ SPLIT	3-LN 4-SUR 4-FAC 1-VOL 4-CRV	4-110 8-100 8-102 9-104F2 19-126 2-128F3 2-128F7 4-142 4-144 4-212	Yes
Solid	SOLID/OPERATN/ SUBTRACT	14-LN 7-SUR 7-FAC 1-VOL 2-CRV	16-110 3-100 14-102 57-126 1-128F3 6-128F7 7-142 7-144 7-212	Yes

Note: Solid elements do not transfer into IGES. Solids are first converted to surfaces before output to IGES.
The volume element is dropped in each case.

Appendix B.

CATIA import to I/EMS

Filename	Errors:	Loss- Yes/No	Rand / Pred
cone2surf	none	No	
cuboid2surf	none	N/A	
cyl2surf	none	No	
depthsurf	none	N/A	
facsurunion	none	No	
offsetsurf1	none	No	
prism2surf	none	No	
pyr2surf	none	No	
ruledsurf1	none	No	
solintrsct	none	No	
solsplit	none	No	
solsub	none	No	
solunion	none	No	
sphr2surf [a]	Unable to fix incosistencies in this entity. Cannot convert entity to B-spline.	Not according to log file.	
torus2surf	none	No	

[a] It is likely that this element is encountering error due to the difficulties in modeling spherical elements (closed in two directions).

CATIA import to CADDS

Filename	Errors:	Loss- Yes/No	Rand / Pred
cone2surf	Trim surface encountered where outer boundary is the natural boundary of the surface. No trimming will be done.	No	
cuboid2surf	none	N/A	
cyl2surf	Trim surface encountered where outer boundary is the natural boundary of the surface. No trimming will be done.	No	
depthsurf	none	N/A	
facsurunion	none	No	
offsetsurf1	none	No	
prism2surf	none	No	
pyr2surf	none	No	
ruledsurf1	none	No	
solintrsct	none	No	
solsplit	none	No	
solsub	none	No	
solunion	none	No	
sphr2surf ▪	An error has occurred trying to trim a surface.	Not according to log file.	
torus2surf	none	No	

▪ It is likely that this element is encountering error due to the difficulties in modeling spherical elements (closed in two directions).

I/EMS import to CATIA

Filename	Errors:	Loss- Yes/No	Rand / Pred
samplec °	Invalid list of elements for the composite curve.	Yes - trimmed surfaces	Pred - 23 of 119 total was lost. - 19.3%
sampled °	A curve software routine resulted in a non-zero return code. Degenerate element.	Yes	Rand - 2 of 364 total was lost. - 0.5%
	One or more costituent elements of composite curve in error: constituent entities are individually processed but no CATIA composite curve is created.	Yes - trimmed surfaces	Rand - 2 of 119 total was lost. - 1.7%
	Invalid list of elements for the composite curve.	Yes - trimmed surfaces	Pred - 23 of 119 total was lost. - 19.3%
samplee °	A Surface software routine resulted in a non-zero return code. Impossible Geometry	Yes	Rand - 4 of 119 total was lost. - 3.4%
	Invalid list of elements for the composite curve.	Yes - trimmed surfaces	Pred - 23 of 119 total was lost. - 19.3%
x9282821 ª	Length of line is zero. Point is created in blink mode.	No	
	Width, height, slant or character number of all text strings of general note with DE # is zero. General note is not processed. - Entity 210	Yes	Rand - 5 of 5 total was lost. - 100%
	Width, height, slant or character number of all text strings of general note with DE # is zero. General note is not processed. - Entity 212	Yes	Rand - 6 of 872 total was lost. - 0.7%
x9282822	Element type change resulted by projecting a "curve" onto a "line".	No	
x9282824	none	No	
x9282832	none	No	
x9282834	Insufficient number of points in copious data element. Minimum: 2. Actual: 1.	Yes	Rand - 1 of 37 total was lost. - 2.7%
	Distance between point at address: 842 and point at address: 843 is too small. Minimum valid value is: .4000E-02.	Yes	Rand - 6 of 182 total was lost. - 3.3%

° It is likely that the "Invalid list of elements" errors are caused by 102's that describe a cylindrical surface. This is currently under investigation.

ª These errors are caused by the I/EMS IGES output processor and not the IGES import function of CATIA. It is under investigation with Intergraph.

CADDS import to CATIA

Filename	Errors:	Loss- Yes/No	Rand / Pred
3w	none	N/A	
425	none	N/A	
68a	none	No	
7150	none	N/A	
part1	Length of line is zero. Point is created in blink mode.	No	

DEVELOPMENT OF CATIA CONVERSION UTILITIES

Sub contract # 0012-0003-AA

CDRL A0002

Software Requirements Specification for the CATIA Conversion Utilities

May 26, 1994

Prepared by

CALS Technology Center

Georgia Institute of Technology

Software Requirements Specification for the CATIA Conversion Utilities

Table Of Contents

List of Figures

List of Tables

1 PDCSF HISTORY

The Product Data Conversion & Storage Facility (PDCSF) at Warner Robins Air Logistics Center (WR-ALC) has been developed to facilitate error free exchange of 3-D geometric representations between the major CAD systems in use at WR-ALC. Development of the Facility began with an extensive requirements gathering process. These requirements were then verified within the WR-ALC user community, specifically the engineering and manufacturing functions. A list of design specifications was developed and implemented. At this point, the facility was turned over to the full-time staff at WR-ALC for daily use.

It is important to note that many of the design criteria for the PDCSF have not changed with the addition of a new CAD system (CATIA). The specifications for the original design can be found in the documents titled as follows:

Preliminary Design Specifications for the PDCSF Conversion Application - CALS Technology Center, 3/25/93

System Design Specification for the PDCSF (Conversion Application) - Georgia Tech Research Corporation, 5/25/93

Detail Design Specification for the PDCSF (File and Catalog Data Management) - Intergraph Corporation, 7/15/93

Detailed Design Specification for the PDCSF Conversion Application - CALS Technology Center, 7/29/93

The aforementioned documents can be found in the PDCSF area at WR-ALC or at the CALS Technology Center at Georgia Tech.

2. SCOPE

2.1 Identification

The scope of the CATIA Conversion Utilities is to facilitate the conversion of 3-D geometric models between CATIA and the five CAD systems in place in the PDCSF Conversion Application at WR-ALC. The five CAD systems are:

a) I/EMS from Intergraph Corporation
b) CADDS 4X from Prime ComputerVision
c) CADKEY from CADKEY Incorporated
d) BRAVO3 from Schlumberger
e) I-DEAS from SDRC

The CATIA Conversion utilities will allow for the conversion between CATIA and the other five CAD systems via the MIL-STD-28000, the Initial Graphics Exchange Specification (IGES).

2.2 System Overview

The CATIA Conversion Utilities will provide a flexible environment for the conversion of 3-D geometric data. These utilities reflect a strategy similar to that of the PDCSF Conversion Application.

A detailed description of the functionality and behavior of the CATIA IGES processor can be found in the document "DEVELOPMENT OF CATIA CONVERSION UTILITIES: Engineering Report Results of CATIA IGES Processor Investigation."

Data conversion via IGES is accomplished through two phases. The first phase is native to IGES conversion and the second phase is the subsequent IGES to native conversion. During these two phases two different types of information losses can occur: random and predictable. The CATIA Conversion Utilities, in conjunction with the established software utilities in the PDCSF Conversion Application, will facilitate the identification, isolation and compensation for both types of information loss. Phase one utilities will be developed within the CATIA programming environment. Phase two utilities will be adapted or developed within the IGES/Works programming environment.

2.3 Conversion Process

The CATIA Conversion Utilities, along with the established software utilities in the PDCSF Conversion Application, will offer two methods for data transfer: direct and

assisted.

Direct conversion of one or more native CATIA models to IGES files or conversion of one or more IGES files to CATIA native files can be accomplished in an automated fashion. The user will type one command line in the operating system and the conversion and necessary flavorings will be done automatically. Direct conversion does not allow for compensation for information loss.

Assisted conversion of one CATIA native model to one of the other native formats is also possible. This type of conversion is highly interactive and will fully compensate for information losses that occur in either phase one or two.

3 CONVERSION ARCHITECTURE

The architecture of the PDCSF has not changed significantly with the addition of CATIA. The same tightly integrated environment is still necessary for data conversion.

3.1 Dedicated Conversion Accounts

The established, dedicated PDCSF accounts will remain unchanged. The hardware hosting CATIA will have a user account dedicated to conversion activities. Direct and assisted conversion to and from CATIA will be accomplished from this account. The name of the account is PDCSF. The password will be PDCSF/PDCSF1 (passwords are subject to change.)

The '.profile' on the dedicated CATIA machine will be changed as necessary for proper operation of the conversion software.

This account is meant to be used only for the following PDCSF activities:

a) Perform Direct Conversion
b) Perform Assisted Conversion
c) Analyze, Flavor An IGES File

3.2 Transparent File System

The transparent file system established for PDCSF will remain unchanged except for the addition of a new directory for CATIA conversions.

The $HOME of each PDCSF account will have the directory structure shown in the upper, boxed portion of the Figure 3.1 - TRANSPARENT FILE SYSTEM FOR CONVERSION.

In this figure, the capital letters denote directories and small letters denote files. The CATIA directory will contain the phase one utilities for assisted conversion from CATIA. The intermediate files created during assisted conversion from CATIA will also be stored under this directory.

The intermediate files generated during the assisted conversion are stored according to the same directory structure used previously. This directory structure is shown in the area enclosed by the dotted rectangle in Figure 3.1 - TRANSPARENT FILE SYSTEM FOR CONVERSION.

The following table lists the NFS mounts involved:

MACHINE	MOUNT	STATUS
pdcsf3	$HOME/convert	remote (client) - YES
ids500	$HOME/convert	remote (client) - YES
idssun	$HOME/convert	remote (client) - YES
eccf3	ECCF3$DUB1:[PDCSF.CONVERT]	remote (client) - NO
ids486	c:\convert	remote (client) - NO
ids6000	$HOME/convert	local (server) - YES

Table 3.1 - NFS Mounts for Transparent File System

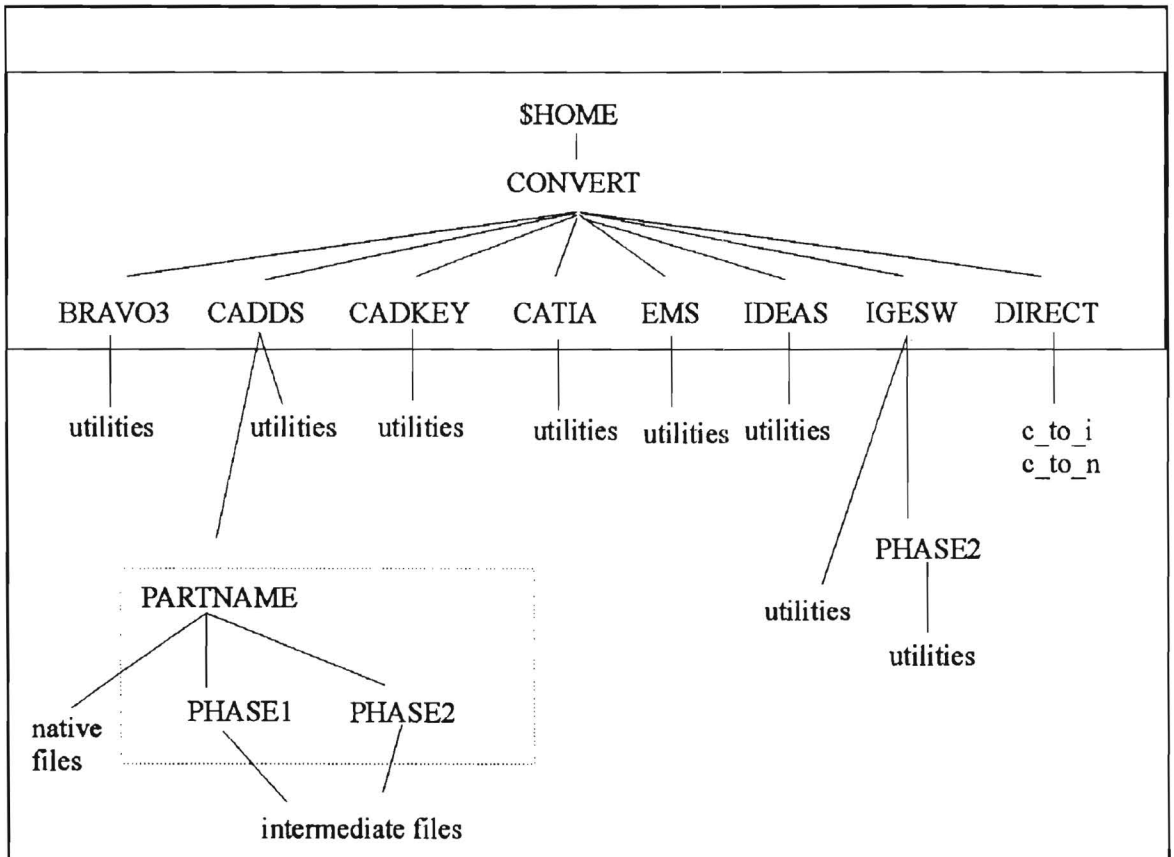


Figure 3.1 - TRANSPARENT FILE SYSTEM FOR CONVERSION

At the present time, all hardware components have NFS capability except eccf3.

3.3 Concurrent Sessions

The necessity for concurrent sessions for visual comparison is unchanged. From our investigation, we have established that CATIA can sustain concurrent sessions with IGES/Works. The same limitations upon resolution and internal configurations exist as in the previous documentation.

3.4 Remote Executions

The capability for remote execution between the first five CAD systems remains unchanged. The following table show the remote execution capabilities of the PDCSF hardware components:

NAME	REMOTE EXECUTION CAPABILITY
pdcsf2	bidirectional remote execution is possible
idssun	bidirectional remote execution is possible
ids500	bidirectional remote execution is possible
ids486	remote execution to other computers is possible (to pc is not)
eccf3	remote execution not possible
ids6000	bidirectional remote execution is possible

Table 3.2 - Remote Execution Capability

It is not possible to remotely submit a command for execution on ids486 or eccf3.

3.5 Launching Applications

The necessity for launching different graphics applications during assisted conversion remains unchanged.

The table below lists the utilities that have been designed and developed for this purpose:

NAME	LOCATION	DESCRIPTION
start_ems UNIX script	$HOME/convert/ems	Starts EMS 2.1 on pdcsf3 Will only run on Intergraph machines (pdcsf*) All the necessary paths are set
start_cadds UNIX script	$HOME/convert/cadds	Starts CADDS 4X on idssun Sunview must be started manually Will only run on idssun All the necessary paths are set
start_ck DOS .BAT file	C:\CONVERT\CADKEY NFS is available	Starts CADKEY on ids486 All the necessary paths are set
start_bravo VMS .COM file	ECCF3$DUB1:[PDCSF.CONVERT] NFS not available	Starts BRAVO3 on ECCF3 Display is automatically redirected to idsvt1 All the necessary paths are set
start_ideas UNIX script	$HOME/convert/ideas	Starts I-DEAS on ids500 X-Windows is started automatically Will only run on ids500 All the necessary paths are set
start_catia UNIX script	$HOME/convert/catia	Starts CATIA on ids6000 Will only run on ids6000 All the necessary paths are set
start_iw UNIX script	$HOME/convert/igesw	Starts IGES/Works on idssun, pdcsf3, ids500 and ids6000 The X-Window environment must be started manually before execution Uses rsh, rcmd capabilities Can be expanded to include any UNIX platform

Table 3.3 - Commands for Launching Applications

3.6 Executing Application Specific Utilities

The CATIA specific utilities are stored under the CATIA directory within the transparent file system as described in Section 3.2 "Transparent File System."

An overview document and step-by-step conversion instructions are provided for CATIA in Section 6 "APPENDIX - CATIA User's Guides."

4 SYSTEM DESIGN

4.1 Hardware Configuration Item Identification

The hardware components of the PDCSF are as follows:

HOSTNAME	IP ADDRESS	MODEL / TYPE
pdcsf3	137.244.160.184	INTERPRO 6545
ids486	137.244.160.185	486 PC
ids500	137.244.160.186	DEC 5000
eccf3	Not Available	VAX
idssun	137.244.160.190	SPARC2
ids6000	137.244.160.219	PowerStation 6000/530H

Table 4.1 - Hardware Components

The applications hosted on or intended functions of these machines are as follows:

HOSTNAME	FUNCTION	OPERATING SYSTEM
pdcsf3	I/EMS host	CLIPPER OS
ids486	CADKEY host	DOS
ids500	I-DEAS host	ULTRIX OS
eccf3	BRAVO3 host	VMS OS
idssun	CADDS 4X, IGES/Works host	SUN OS
ids6000	CATIA host	AIX OS

Table 4.2 - Functions of PDCSF Hardware

All these machines have TCP/IP based Local Area Networking between them, with the exception of eccf3. This network is shown in Figure 4.1 - CONVERSION ARCHITECTURE.

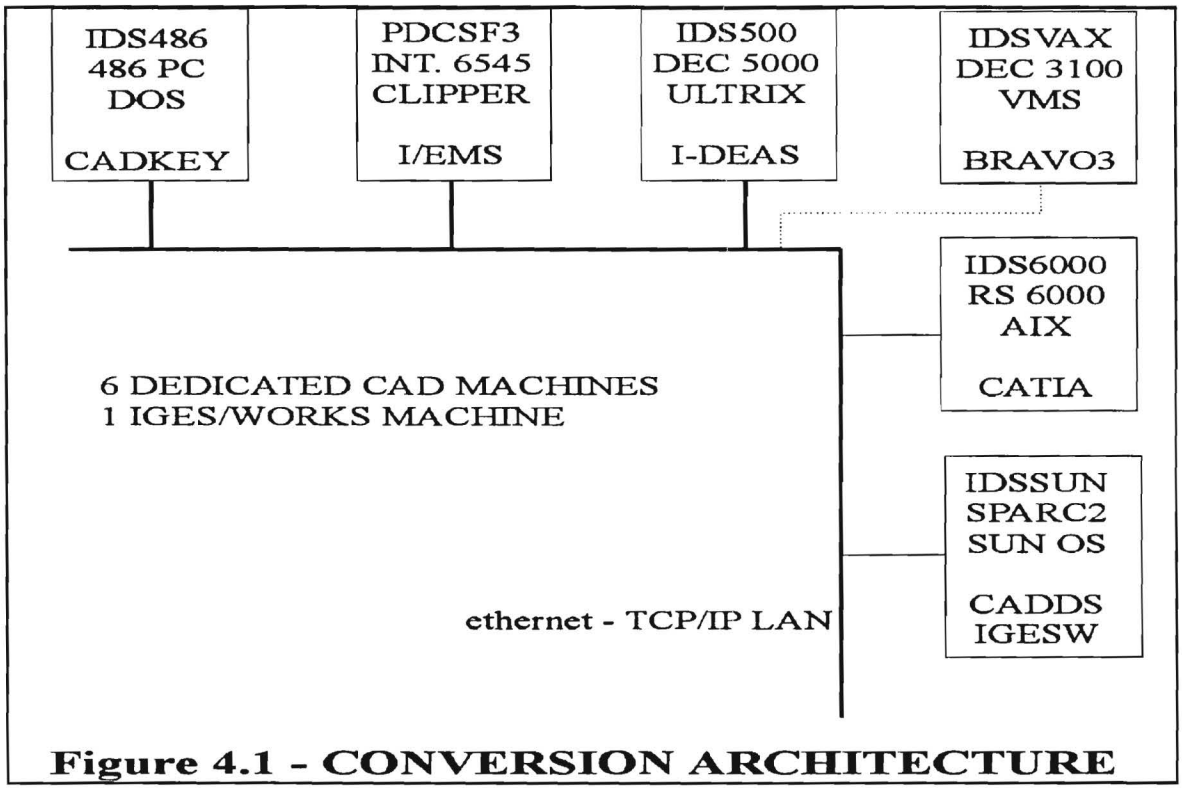


Figure 4.1 - CONVERSION ARCHITECTURE

4.2 Software Configuration Item Identification

The Software Components (SC) for the CATIA utilities are as follows:

Assisted Conversion SC
- CATIA Phase I SC
- Phase II SC

Direct Conversion SC
- Convert_to_Native SC
- Convert_to_IGES SC

The hierarchical decomposition is given in Figure 4.2 - SOFTWARE COMPONENTS - CONVERSION SC. This figure also contains pairs of bars that indicate the current level of completion for the software development and systems integration for each SC.

Design specifications for each of the aforementioned SC are detailed in the following sections.

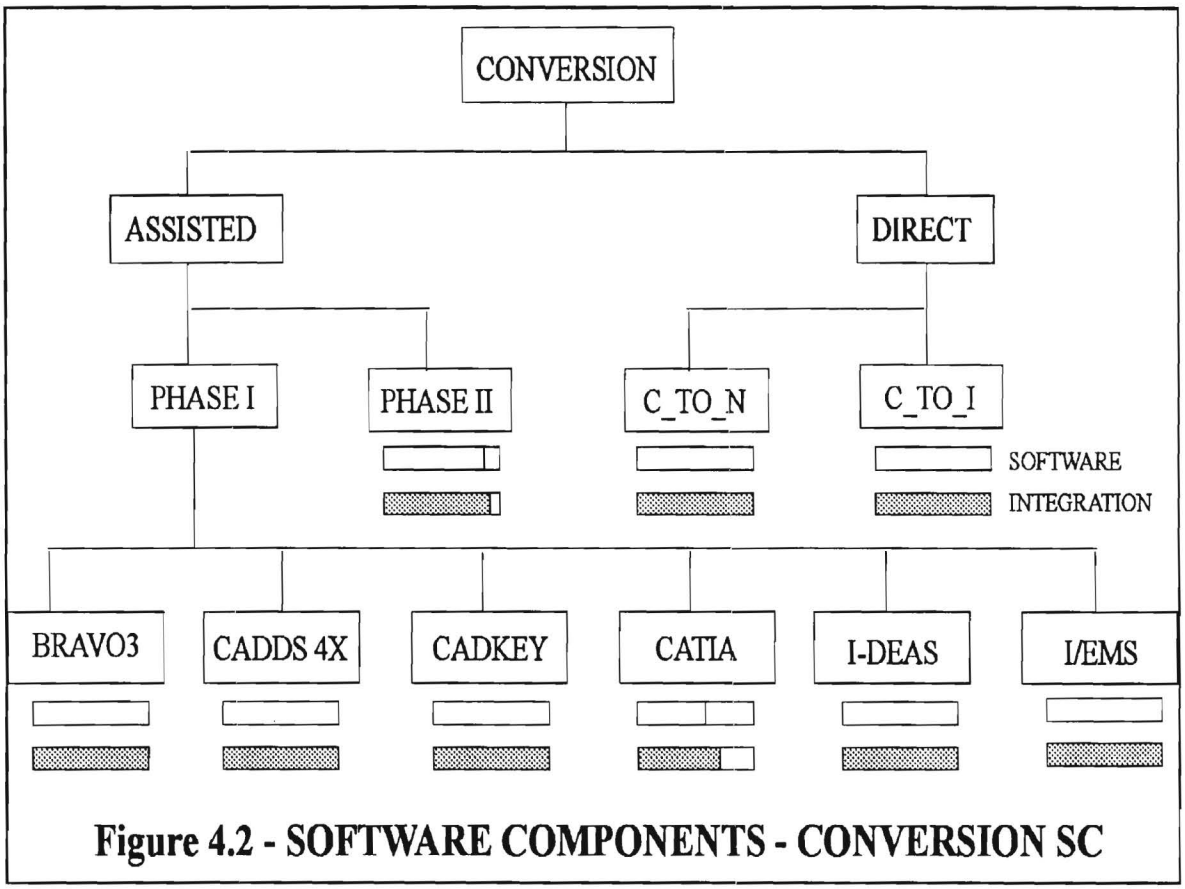


Figure 4.2 - SOFTWARE COMPONENTS - CONVERSION SC

4.2.1 Assisted Conversion SC

The assisted conversion software can process one CATIA file at a time. Compensation for information losses is possible during assisted conversion. Assisted conversion is highly interactive. The various SC that compose the assisted conversion process are detailed in the following sections.

4.2.1.1 Phase I SC

This SC consists of numerous utilities written in the CATIA macro programming language. These utilities take full advantage of the geometric modeling capabilities within the CATIA environment. Consequently, they are also subject to any limitations within CATIA.

The software utilities that are a part of this SC will require updating as the CATIA software is updated.

A complete listing of the CATIA Phase I SC and a brief description of each utility is as follows:

CLEANUP
Deletes intermediate files created during the conversion process.
cleanup

COMPARE
Asks user to identify dropped surfaces and isolates them to si file.
Select "COMPARE" from phase1 function.

DISPLAY
Fits all graphics to isometric view.
Select "DISPLAY" from phase1 function.

EXTRACT_SI
Creates bounding curves and iso-pars for each problem surface.
Select "EXTRACT_SI" from phase1 function.

EXTRACT_SI2
Creates bounding curves and iso-pars for each problem surface.
Select "EXTRACT_SI2" from phase1 function.

START_CATIA
Starts the CATIA CAD system and opens the model filename.
start_catia filename phasenum

WRITE_IGES
Writes an IGES file for the named file.
write_iges filename

WRITE_OK
Writes an IGES file for the ok file.
write_ok filename

WRITE_SI
Writes an IGES file for the si file.
write_si filename

WRITE_SI2
Writes an IGES file for the si2 file.
write_si2 filename

4.2.1.2 Phase II SC

The Phase II SC consists of numerous IGES/Works utilities and UNIX scripts. These utilities remain unchanged with the exception of the following SC:

ISOLATE PRED
Separates the elements causing predictable errors from ok.igs.
This is invoked from the menu button, "Isolate Pred."

PARSE
Sets up data files for Isolate Pred.
This is invoked from the menu button, "Parse."

SELECT PART
Sets up files for starting the phase two conversion process.
This is invoked from the menu button, "Select Part."

TO NATIVE
Reads the input IGES file and converts to the native CAD format.
This is invoked from the menu button, "To Native."

Please consult documents outlined in Section 1 "PDCSF HISTORY" for a complete listing of the Phase II SC.

The Phase II utilities will require updating if the behavior of the CATIA IGES processor changes drastically.

4.2.2 Direct Conversion SC

The direct conversion can convert one or many IGES files to CATIA native models or one or many CATIA native models to IGES files. No compensation for information loss is possible with direct conversion. The direct conversion is completely automatic for CATIA.

The CATIA specific software utilities have been designed to run on the transparent file system, described in Section 3.2 "Transparent File System." These utilities are designed to work within the hardware configuration described in Section 4.1 "Hardware Configuration Item Identification."

The direct conversion SC consists of two SC: Convert_to_Native and Convert_to_IGES.

4.2.2.1 Convert_to_Native SC

Convert_to_Native enables the conversion of one or many IGES files to CATIA native models.

Operation of this command is unchanged with the exception of the addition of CATIA (cat) as an acceptable target name. For more detailed information on this SC, please consult documentation described in Section 1 "PDCSF HISTORY."

4.2.2.2 Convert_to_IGES SC

Convert_to_IGES enables the conversion of one or many CATIA native models to IGES files.

Operation of this command is unchanged with the exception of the addition of CATIA (cat) as an acceptable source name. For more detailed information on this SC, please consult documentation described in Section 1 "PDCSF HISTORY."

DEVELOPMENT OF CATIA CONVERSION UTILITIES

Sub contract # 0012-0003-AA

CDRL A0003

Software Design Document for the CATIA Conversion Utilities

June 2, 1994

Prepared by

CALS Technology Center

Georgia Institute of Technology

Software Design Document for the CATIA Conversion Utilities

Table Of Contents

CATIA CONVERSION UTILITIES - Software Design Document

1 PDCSF HISTORY

The Product Data Conversion & Storage Facility (PDCSF) at Warner Robins Air Logistics Center (WR-ALC) has been developed to facilitate error free exchange of 3-D geometric representations between the major CAD systems in use at WR-ALC. Development of the Facility began with an extensive requirements gathering process. These requirements were then verified within the WR-ALC user community, specifically the engineering and manufacturing functions. A list of design specifications was developed and implemented. At this point, the facility was turned over to the full-time staff at WR-ALC for daily use.

It is important to note that many of the design criteria for the PDCSF have not changed with the addition of a new CAD system (CATIA). The specifications for the original design can be found in the documents titled as follows:

Preliminary Design Specifications for the PDCSF Conversion Application - CALS Technology Center, 3/25/93

System Design Specification for the PDCSF (Conversion Application) - Georgia Tech Research Corporation, 5/25/93

Detail Design Specification for the PDCSF (File and Catalog Data Management) - Intergraph Corporation, 7/15/93

Detailed Design Specification for the PDCSF Conversion Application - CALS Technology Center, 7/29/93

The aforementioned documents can be found in the PDCSF area at WR-ALC or at the CALS Technology Center at Georgia Tech.

2 SYSTEM OVERVIEW

The CATIA Conversion Utilities will provide a flexible environment for the conversion of 3-D geometric data. These utilities reflect a strategy similar to that of the PDCSF Conversion Application.

The philosophy of the CATIA Conversion Utilities has been described in detail in the Software Requirements Specification.

A detailed description of the functionality and behavior of the CATIA IGES processor can be found in the document "DEVELOPMENT OF CATIA CONVERSION UTILITIES: Engineering Report Results of CATIA IGES Processor Investigation."

Data conversion via IGES is accomplished through two phases. The first phase is native to IGES conversion and the second phase is the subsequent IGES to native conversion. During these two phases two different types of information losses can occur: random and predictable. The CATIA Conversion Utilities, in conjunction with the established software utilities in the PDCSF Conversion Application, will facilitate the identification, isolation and compensation for both types of information loss. Phase one utilities will be developed within the CATIA programming environment. Phase two utilities will be adapted or developed within the IGES/Works programming environment.

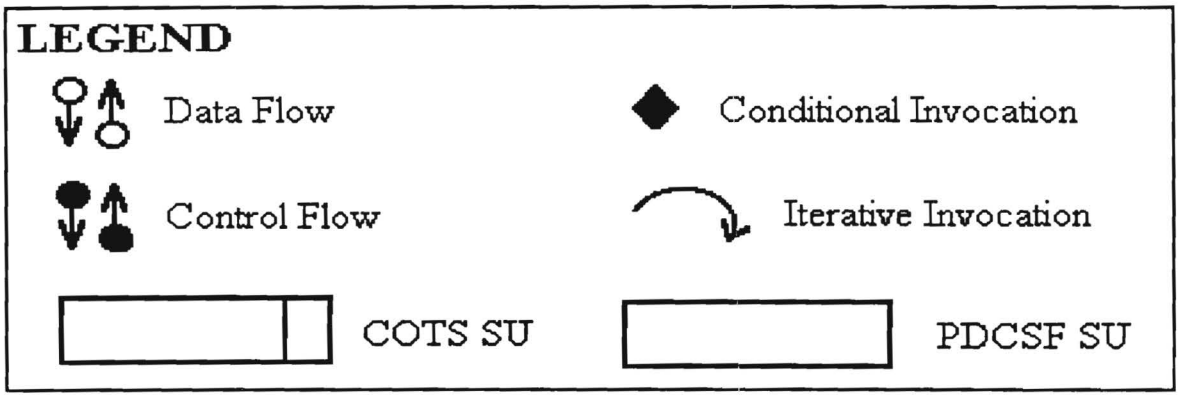
3 SOFTWARE SPECIFICATIONS

The following Sections contain structure charts and the related Software Unit (SU) specifications organized in alphabetical order. This document shows the detail design for the CATIA Conversion Utilities, including CATIA macros, CATIA-specific OS utilities, CATIA-specific IGES/Works utilities and previously developed utilties that were affected by the addition of CATIA to the PDCSF. Details on the established PDCSF software can be found in the following documents:

Detail Design Specification for the PDCSF (File and Catalog Data Management) - Intergraph Corporation, 7/15/93
Detailed Design Specification for the PDCSF Conversion Application - CALS Technology Center, 7/29/93

The aforementioned documents can be found in the PDCSF area at WR-ALC or at the CALS Technology Center at Georgia Tech.

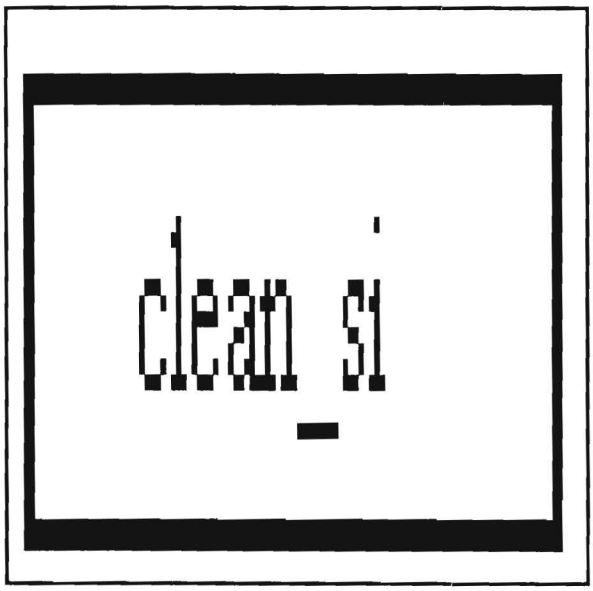
4 ASSISTED CONVERSION SC



4.1 Phase I SC

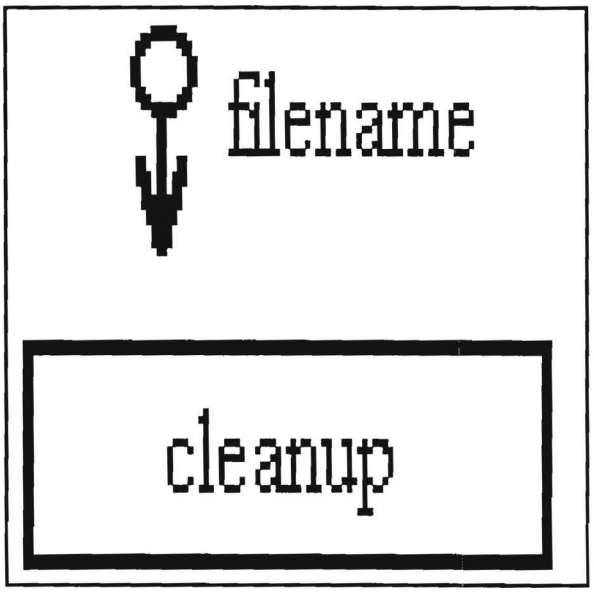
CLEAN_SI

* CLEAN_SI :Deletes original problem surfs from si model.
*----------



CLEANUP

```
# Name       : cleanup
# Originator: Gayle G. Casey
#           : CALS Technology Center, Georgia Tech
#           : For support:
#           :   Gayle G. Casey
#           :   (404) 894-9811
#           :   casey@cad.gatech.edu
# Latest     : Thursday, June 2, 1994
# Product    : CATIA
# Command    : cleanup filename
# Comments   : Removes all of the files associated with the filename CATIA
#     : part.  Used at the end of conversion after the part has been
#     : returned to the central repository.
#
```



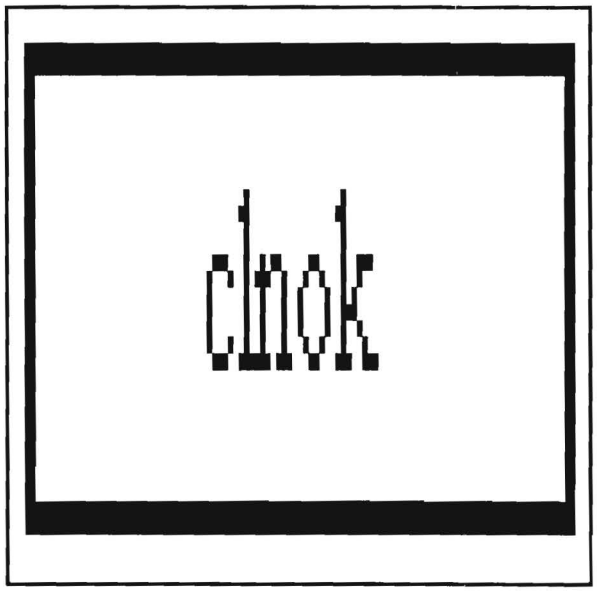
CLNOK

C This routine is to be called from the GII function phase1
C from the sub menu item, ISOLATE
C It removes all the picked faces and surfaces from the OK model after
c the COMPARE operation.

c Find the DDNAME (as FILE) from DSNAME

c Select only no-pick faces and surfaces and erase them

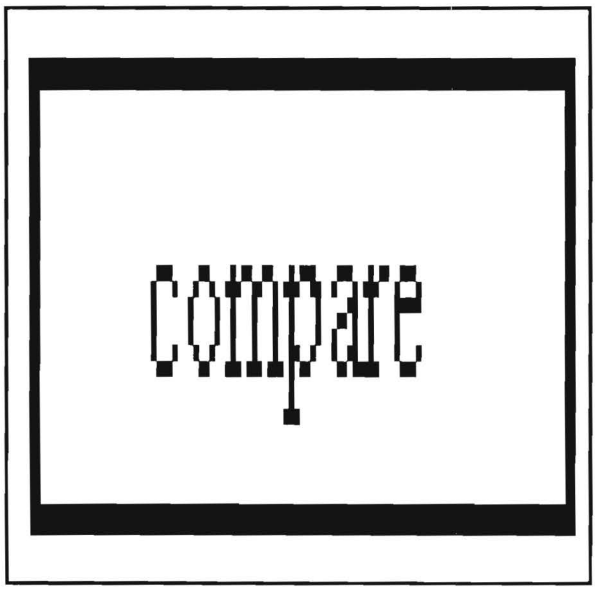
c Save current model (OK)



COMPARE

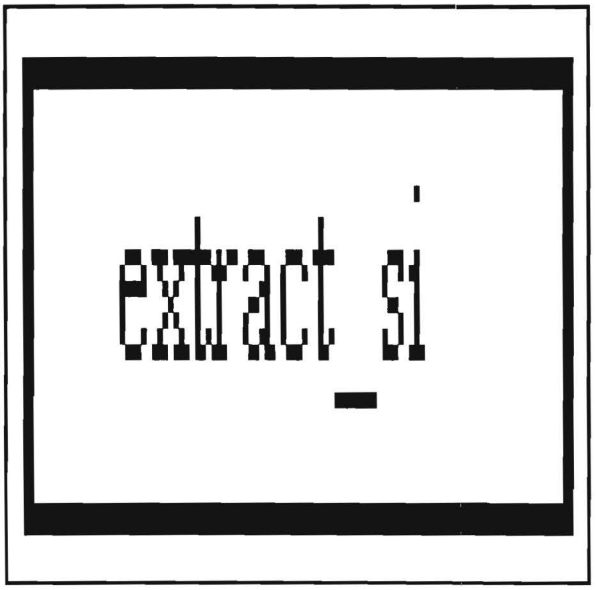
```
* COMPARE : Loops through all faces & surfaces in model highlighting each
*          prompting the user to pick the surfaces & faces that are lost
*          The picked faces & surfaces are moved to the model SI
*-----------------------------------------------------------------------
*
* Element types SUR (5) and FAC (6) are selected
*
*  CALL CATGEO
*
*  Select the next face or surface

*  If JADP is not the last element

*    Find its graphic display attributes

*    Change the display attributes to make the boundary visible

*    Remove highlighting from previous one and highlight the current one.
*

*    Make selected ones unpickable and the others pickable

*    Remove highlighting from the previous one

*  Ask if user wishes to quit

* Accept the changes to the model so far

* Save current model (OK)

* Select all the faces and surfaces that are pickable into a stack
* and erase them.

* Empty the stack

* Select all the elements other than faces and surfaces and erase them.



* Save current model as a new model 'SI'
```

* Make sure that the IUA does not end



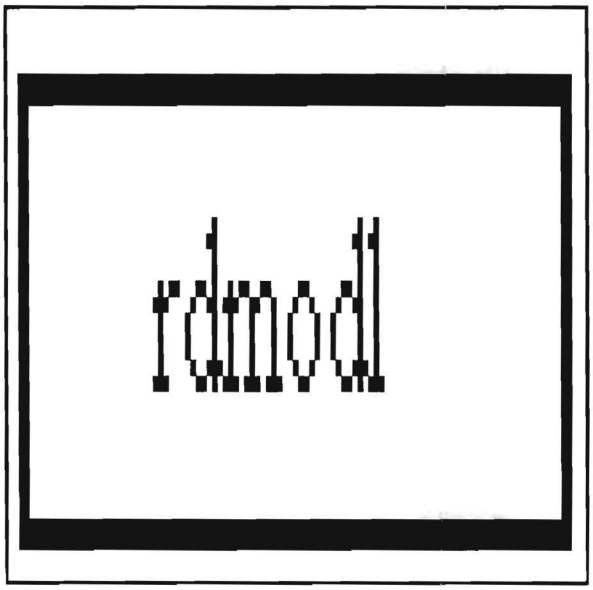
EXTRACT_SI

* EXTRCSI : Instructs user on extraction of isopars and bounding curves.



RDMODL

```
C       rdmodl.f

C       Called by IUA procedure rdmodl to setup display.
```



START_CATIA

```
#               Unix Script start_catia
#  Author: V. Kaladi
#          CALS Technology Center, Georgia Tech
#          (404)-894-7777
#          e-mail: vkaladi@cad.gatech.edu
#
#  This script starts up CATIA for assisted conversion with the model given
#  by argument 1.

# Define the full pathname of the home directory

# Define the directory for CATIA pdcsf scripts and CATIA pdcsf configuration
# files.

#check arguments

# directory to hold logfiles.

# Select CATIA model to use based on phasenum ($2)

# $HOME/CATIA/assisted is defined to be a CATIA model file in the SYSDYN
# file, $ctacfgdir/sysdyn. Set up a symbolic link to point
# $HOME/CATIA/assisted to $MODELDIR to define $MODELDIR as a CATIA
# model file.
# This is to avoid problems with long pathnames in SYSDYN file.

# edit sysdyn file to define MODELDIR as a CATIA model file, with DSNAME as
# MODELFILE

# For phase1 do the preprocessing to create the OK model in inches and
# with solids converted to surfaces.

  # Create a summary output of the original model as catsumr.out

  # Copy original model to OK. Do further processing on OK.

  # Use the model OK for further processing

  # Convert all solids in OK model to surfaces and wireframes
```

Convert units in the model OK to inches

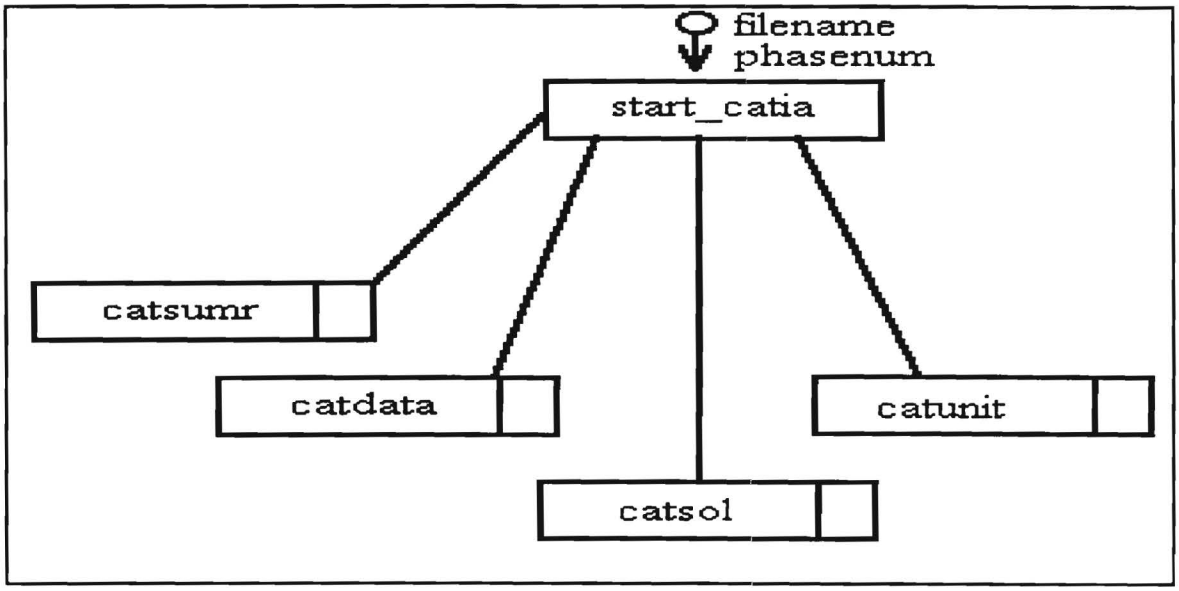
create input file for FORTRAN utility to read from.

#start CATIA for assisted conversion

erase input file for IUA utility.

#edit sysdyn file to remove $MODELFILE as a CATIA model file

Remove the symbolic link from $HOME/CATIA/assisted to $MODELDIR



WRITE_IGES

```
#               Unix Script writeiges
#  Author: V. Kaladi
#          CALS Technology Center, Georgia Tech
#          (404)-894-7777
#          e-mail: vkaladi@cad.gatech.edu
#
# This utility converts the CATIA model in the model file,
# $HOME/convert/catia/file_name to IGES during phase1 of assisted
# conversion from CATIA.
# The CATIA model that is actually converted is the model OK at
# the beginning of assisted conversion, which is produced during
# the preprocessing stage of start_catia where the units are converted
# to inches and solids are converted to surfaces from the original model.
# Usage:
#       writeiges file_name
#

# Define the home directory

# Define the directory for CATIA pdcsf scripts and CATIA pdcsf configuration
# files.

#Assign names to the arguments

#directory to hold the log files

# Work from the convert/catia directory

# ASSISTED is already defined as a CATIA model file by the utility
# start_catia before this utility (writeiges) is run.

# Write IIF file from $MODELTITL

# Convert IIF to IGES file
```


filename
writeiges

WRITE_OK

```
#               Unix Script writeok
#  Author: V. Kaladi
#          CALS Technology Center, Georgia Tech
#          (404)-894-7777
#          e-mail: vkaladi@cad.gatech.edu
#
# This utility converts the CATIA model OK, in the model file,
# $HOME/convert/catia/file_name to IGES during phase1 of assisted
# conversion from CATIA.
# Usage:
#       writeok file_name
#

# Define the home directory

# Define the directory for CATIA pdcsf scripts and CATIA pdcsf configuration
# files.

#Assign names to the arguments

#directory to hold the log files

# Work from the convert/catia directory

# ASSISTED is already defined as a CATIA model file by the utility
# start_catia before this utility (writeiges) is run.

# Write IIF file from $MODELTITL

# Convert IIF to IGES file
```


filename
writeok

WRITE_SI

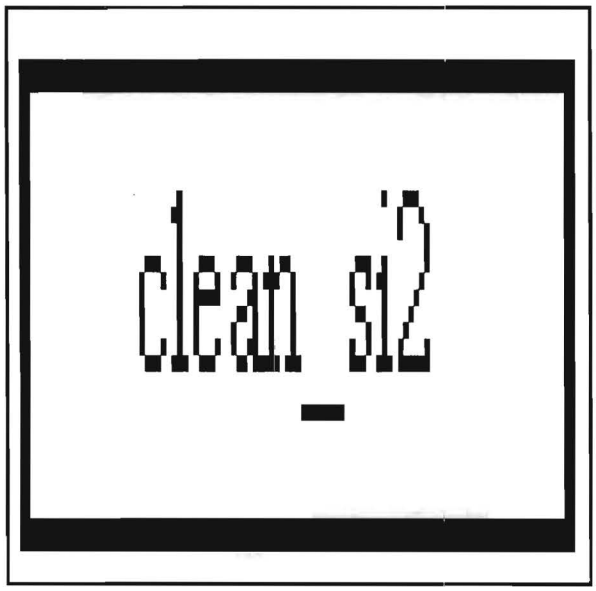
```
#               Unix Script writesi
#  Author: V. Kaladi
#          CALS Technology Center, Georgia Tech
#          (404)-894-7777
#          e-mail: vkaladi@cad.gatech.edu
#
# This utility converts the CATIA model, SI in the model file,
# $HOME/convert/catia/file_name, to IGES during phase1 of assisted
# conversion from CATIA.
# Usage:
#       writesi file_name
#

# Define the home directory

# Define the directory for CATIA pdcsf scripts and CATIA pdcsf configuration
# files.

#Assign names to the arguments

#directory to hold the log files

# Work from the convert/catia directory

# ASSISTED is already defined as a CATIA model file by the utility
# start_catia before this utility (writeiges) is run.

# Write IIF file from $MODELTITL

# Convert IIF to IGES file
```


filename
writesi

4.2 Phase II SC

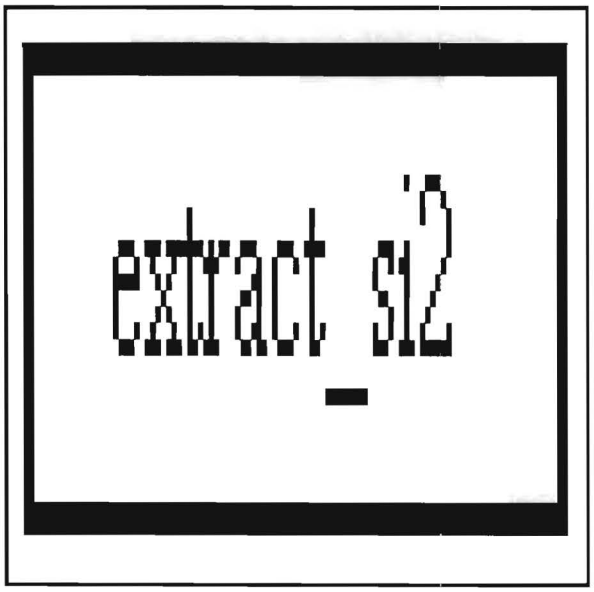
CLEAN_SI2

* CLEAN_SI2 :Deletes original problem surfs from si2 model.
*----------



EXTRACT_SI2

* EXTRCSI2 : Instructs user on extraction of isopars and bounding curves.



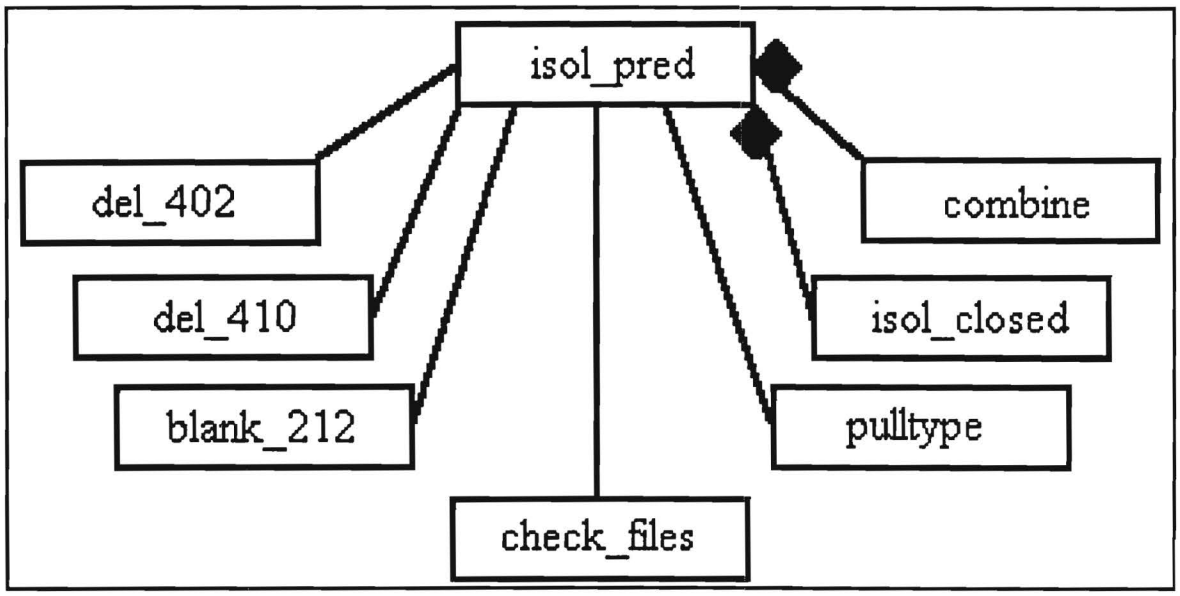
ISOLATE PRED

```
#               IGES/Works Script: isol_pred.scr
#
#  Author:  V. M. Kaladi
#           CAE/CAD Lab, Georgia Tech
#           PDCSF Project
#
#  Modified May 26, 1994 for CATIA by Gayle G. Casey.
#
#  Script to isolate all predictable problem entities from the file
#  phase2/ok.igs into a new file
#  $in_dir/phase2/pred.igs and then to overwrite the file
#  $in_dir/phase2/ok.igs with the remaining elements.
#  The predictable problem entities are selected based on the flags
#  contained in the data file $pred_dat

#  Other scripts called: pulltype.scr
#                        iso_closed.scr
#                        combine.scr
#                          blank_212.scr
#
#  Note: uses the global variable, partdir, targetsys, defined in .igwin_pdcsf

#  define all the variables that are needed

# set the name of the home directory in $HOME

# set the directory name for the direct conversion utilities and the
# corresponding temporary data files in $direct_dir.

#  $in_dir is the full pathname of the directory for the part

# store the name of the phase2 IGES/Works utilities directory in $phase2dir

#  assign values to the file names
####################
# Isolate all common predictable errors.
# This includes views and groups

################
# Delete groups
```

```
  # open the data file for the script del_402.scr as $del_402_dat
  # and write the name of the file ok.igs into it.

#######################
# Delete views

  # open the data file for the script del_410.scr as $del_410_dat
  # and write the name of the file ok.igs into it.

#######################
# Blank 212's

  # open the data file for the script blank_212.scr as $blank_212_dat
  # and write the name of the file ok.igs into it.

#######################

#  use the Unix script check_files to see that $pred_dat is
#  non-empty and return the result in the data file check_files.dat

#  open the data file check_files.dat and read the result of checking
#  for the presence of file $pred_dat. $yes=1, if $pred_dat is nonempty.

#  read the first record; if none, $eof=0
#  and get the value of $yes from the buffer

#  close the data file check_files.dat

#  erase the data file check_files.dat

#  allocate a model named testpred

#  open the input IGES file and create a model

  # store all the independent elements in selection all_ent
  # initially set selection list ok to be all of all_ent

    # $pred_dat is not empty

    #  open the data file $pred_dat and read the flags

    #  read the first record; if none, $eof=0
```

```
# until the end of file on $pred_dat

  # get the value of $flag

    # use pulltype.scr to get a selection list named surf144
    # consisting of all entities of type IGES #144

    # save as selection list ok all entities in the previously saved
    # selection ok which are not in surf144

  # Ensure ok is now the current selection list

    # use pulltype.scr to get a selection list named surf143 consisting of
    # all entities of type IGES #143

    # save as selection list ok all entities in the previously saved
    # selection ok which are not in surf143

  # read the next record; if none, $eof=0

# Close the data file $pred_dat whose ID is $fid

# End of if $pred_dat is not empty

# allocate a new model named test2

# now restore the saved selection list ok as the current
# selection list but keep the saved selection list ok

# copy all the entities in the current selection list to model test2
# These are all the ok elements now

# Write the selection list ok to the file $in_iges (overwrite)
# and free model test2

# allocate a new model named test2

# go back to the original model test

# select entities which are in all_ent that are not in ok
```

```
    # These are all the predictable problem elements now
    # copy them to the model test2 and output to the file $pred_iges

# If the target CAD system is CADDS, remove closed #144's
# Use iso_closed.scr to extract all closed surfaces into closed.igs
# and then use combine.scr to combine closed.igs and pred.igs into pred.igs.

  # open the data file for the script iso_closed.scr as $iso_closed_dat
  # and write the names of the files ok.igs and closed.igs  into it.

  # open the data file for the script combine.scr as $combine_dat
  # and write the names of the files pred.igs and closed.igs  into it.
  # pred.igs and closed.igs are combined into pred.igs

# If the target CAD system is CATIA, remove closed #144's
# Use iso_closed.scr to extract all closed surfaces into closed.igs
# and then use combine.scr to combine closed.igs and pred.igs into pred.igs.

  # open the data file for the script iso_closed.scr as $iso_closed_dat
  # and write the names of the files ok.igs and closed.igs  into it.

  # open the data file for the script combine.scr as $combine_dat
  # and write the names of the files pred.igs and closed.igs  into it.
  # pred.igs and closed.igs are combined into pred.igs

  # End of If target specific actions
```



PARSE

```
/*                PROGRAM PARSE.C                                    */
/*                                                                   */
/*  Author: V.M. Kaladi                                              */
/*          CAE/CAD Lab, Georgia Tech                                */
/*          PDCSF Project                                            */
/*                                                                   */
/*  Changed May 27, 1994 for CATIA by Gayle G. Casey                 */
/*                                                                   */
/*  PARSES THE FILE                                                  */
/*  This program parses the IGES file and generates a summary of     */
/*  its contents - a list of element types and number of occurences  */
/*  are printed                                                      */

/*  ADVISES ON CONVERSION */
/*  Based on the results of parsing and the intended target          */
/*  a list of potential problems and suggested solutions are         */
/*  produced. A file pred.dat is created in the current directory    */
/*  containing the entity type nos. of all predictable problem       */
/*  present.                                                         */

/* INVOCATION                                                        */
/* "parse filename wfm/sur/sld/all/phase2/autoconv target"           */
/*   filename --> the name of the iges file                          */
/*   wfm --> produces summary of wireframe elements                  */
/*   sur --> produces summary of surface elements                    */
/*   sld --> produces summary of solid elements                      */
/*   str --> produces summary of structural elements                 */
/*   all --> produces summary of ALL elements                        */
/*   phase2 --> no summary, used for phase2 conversion advice        */
/*   autoconv --> no summary, used for automatic conversion advice   */
/*   target --> the name of the target CAD system for conversion advice
                 if "none", no conversion advice is provided
*/

/* ERROR CODES                                                       */
/*   -1 --> Insufficient Arguments                                   */
/*   -2 --> Problem Opening File                                     */
```

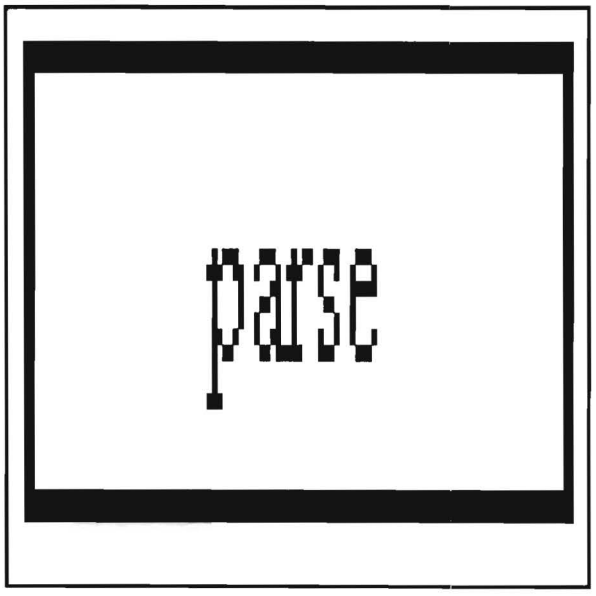
```
    /*Flags for advise section*/

/* keeps count of each IGES entity */
/* keeps count of entities needing attention */
/* Total Number of Elements */

      /*Main program begins now*/

  /* IGES entity      */
  /* flags for entities needing attention      */
  /* start of sections         */
  /* current position in file       */
  /* 1st and 2nd lines of    */
  /* directory entry         */

       /* Check Command Line Arguments                              */

       /* Open the File to be Parsed                          */

  /*argv[1] is the name of IGES file to parse  */

       /* initialize counts to zero                          */

       /* Get the start locations of each section                */

  /* go to start of DE section */

  /* Now in the Directory Section */
    /*read the second line for this entry*/
    /* store position in file before function call */
    /* go back to where left off */
    /*read the first line for next entry*/
  /* end of while loop */

       /* Directory Entry Section completed                     */
       /* Start of Element Statistics Section                 */

       /* Total up count for all entities and print             */

  /* avoid display of source name - Vasu Kaladi, Aug 30, '93  */
  /* display_source_name(fp, Orig_glob); */
```

```
      /* print wireframe summary */

      /* print surface summary */

      /* print solid summary */

      /* print structure summary */

   /* create autoconv_parse.dat for both automatic and phase2 conversion */

      /*provide target specific conversion advice */

        /* End of Element Statistics Section                    */

       /* FUNCTION get_origin */
/* start of sections    */

       /* FUNCTION display_source_name */


/* start of global section    */

   /* source system name string */
   /* no of characters in source system name */

   /* go to beginning of global section */

      /* Read and discard all the previous strings and then read the  */
      /* no. of characters in the source system name string          */
      /* Each is a Hollerith string with a comma at the end          */
      /* Note: it is assumed that the name does not spill onto  the  */
      /*       2nd line of the Global section                        */


       /* FUNCTION entity_to_string */

            /* end of switch */

       /* FUNCTION entity_label */
          /* This outputs the string corresponding to the entity */
          /* for printout of summary information */
```

```
            /* end of switch */

        /* FUNCTION parse_dir_entry */

/* start of sections          */
/*pointer to IGES entity       */
  /* IGES entity       */

     /* get the string equivalent of entity */
     /* if 1st 8 characters of line1 & str
     are equal                          */
     /* entity found                    */
  /* end of for loop                  */

  /* Note: entity_to_string outputs str[]="     106" for both            */
  /*       forms 11 and 12. The form no. is line2 at column 33          */
  /*       of field width=8                                             */

     /* 102 entity found   */
     /* 106 entity found   */
     /* form no=11   */
       /* form no=12   */

    /* Set flags for entities needing attention  */
    /* Note: count_flag is a global variable     */
       /* independent */

  /* to return entity to calling program */

       /* FUNCTION check_102                                            */

/* To check if the composite curve entity (#102) just found points to     */
/* either a conic arc (#104) or a #106 Form 12 entity                     */

  /* parameter entry no. as per line1 */
     /* No. of constituent curves in the composite  */
     /* Directory entry no. of constituent curve */
     /*   dummy integer                 */

  /* pointer to parameter entry      */
  /* starts at line1[8]             */
  /*offset to parameter entry    */
```

```
/* go to the parameter entry   */
/*read the no. of sub curves      */

/* Note: count_flag is a global variable */

/*check each sub curve              */
  /*get the DE# of this sub curve    */
  /*keep the position in the
  parameter entry of the composite curve  */
  /*offset to directory entry */
  /*go to the directory entry of the sub curve */
  /*read this directory entry   */
  /* if entity type is 106 */
    /*read 2nd line of directory entry*/
    /*form no starts col.33*/
      /*#102 pointing to #106 form 12 found*/
  /* if entity type is 106 */
    /*#102 pointing to #104 found*/
  /* return to the parameter entry of #102   */
/*end of for loop */

     /* FUNCTION WFM_Summary() */

    /* e100 to e130 are wireframe entities        */
      /* Output count and percentage of each entity */

     /* get the output string */

     /*FUNCTION SUR_Summary()*/

    /* Entities 118 to 143 are surface entities   */
      /* Output count and percentage of each entity */

     /*FUNCTION SLD_Summary()*/

    /* Entities 150 to 430 are solid entities     */
      /* Output count and percentage of each entity */

     /* FUNCTION STR_Summary() */

    /* Entities 302 to 422 are structural entities  */
      /* Output count and percentage of each entity   */
```

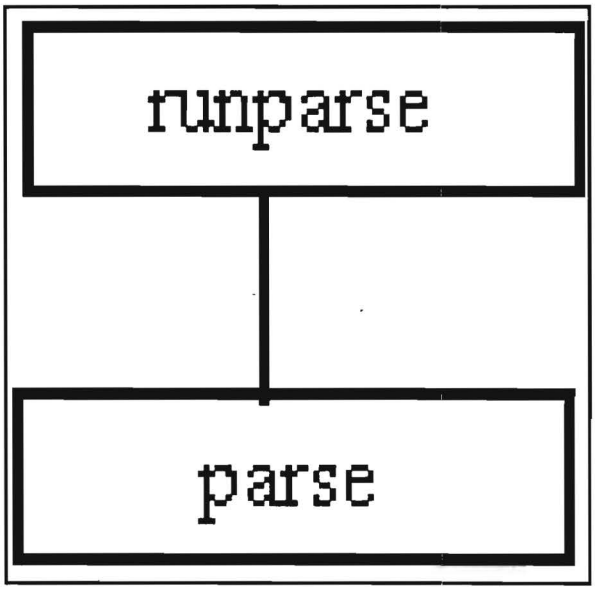
```
   /* FUNCTION advise() */

   /* FUNCTION bravo3_advise() */
/*  create the file pred.dat  */

   /* FUNCTION cadds_advise() */

   /* FUNCTION ems_advise() */

   /* FUNCTION ideas_advise() */

   /* FUNCTION cadkey_advise() */

/*  create the file pred.dat  */

  /* FUNCTION catia_advise() */

   /* FUNCTION designer() */

   /* FUNCTION autoconv() */
```



RUNPARSE

```
#               IGES/Works Script: runparse.scr
#
#  Author:  V. M. Kaladi
#           CAE/CAD Lab, Georgia Tech
#           PDCSF Project
#
#  Modified May 26, 1994 for CATIA by Gayle G. Casey.
#
#  Note run_parse is not called  (Vasu Kaladi, Aug 3, '93)
#  Script to call the Unix script, run_parse that calls the parse program
#  (in C) with the appropriate command line arguments.
#  Note: uses the global variable, partdir, defined in .igwin_pdcsf

#  define all the variables that are needed

  #  get the name of the input directory in $in_dir first and then
  #  assign full pathname of file phase2/ok.igs to $in_iges

    # prompt the user with the menu to select the target system name

    #  read user input and assign system name to $sysname

  # Continue only if menu option chosen is not "Quit"
    #  set $command to contain the command to invoke the parse program
    #  with the options specified above

    # move pred.dat to the phase2 directory

    # Assign the name of the target CAD system to the global variable targetsys

    #  allocate a model

    #  read the iges file $in_iges

    #  Search For Drawing Entities
      #  if drawings are present display message and quit phase2 conversion.

    # free the model

    #  delete autoconv_parse.dat
```

```
  # end of if sysname ne "quit"

# end of if partdir=""
```

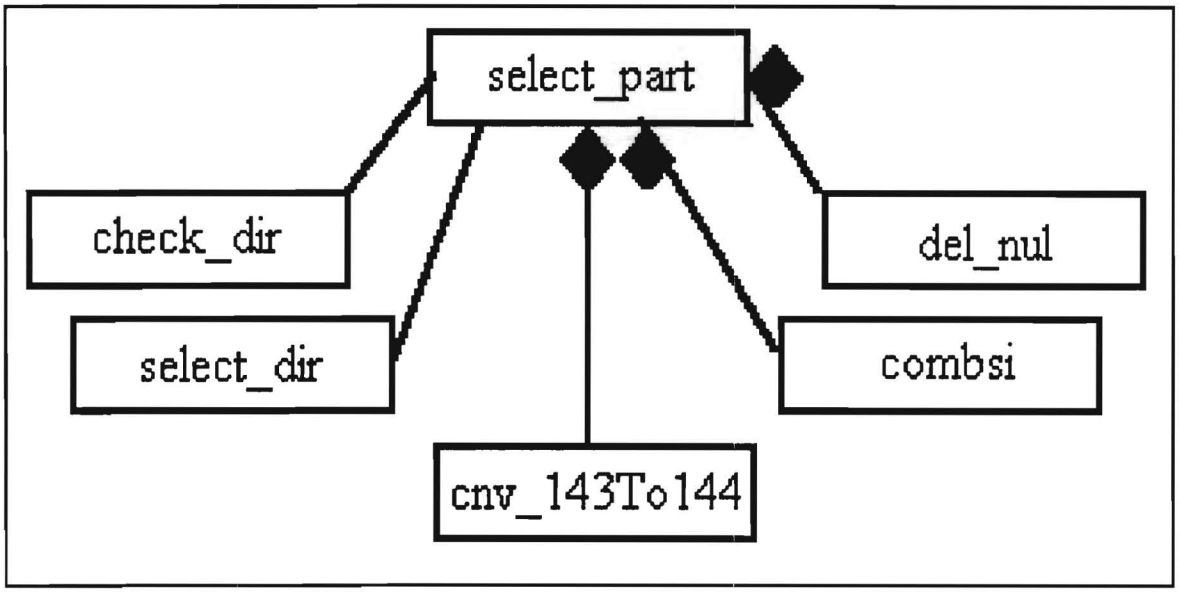


SELECT PART

```
#               IGES/Works Script: select_part.scr
#
#  Author:  V. M. Kaladi
#           CAE/CAD Lab, Georgia Tech
#           PDCSF Project
#
#  Modified May 26,1994 for CATIA by Gayle G. Casey
#
#  Script to choose the directory containing the IGES file to work on.
#  use the sh script select_dir.
#  Note: This script is to be called while in the directory
#         $HOME/convert/igesw
#  Note: uses the global variables, partdir, sourcesys defined in .igwin_pdcsf

#  Other scripts called: select_dir (Unix scripts)

#  define all the variables that are needed

#Edit the next line to assign the value of $HOME

# set the directory name for the direct conversion utilities and the
# corresponding temporary data files in $direct_dir.
# set the directory name for the ideas conversion utilities
########

 # prompt the user with the menu to select the source system directory

 #  read user input and assign system name to $sysname

  # break out of the while ($selectpart, eq, continue) loop

  # go to the end of the while ($selectpart, eq, continue) loop

 #  store the source system name in global variable sourcesys

 #  inform the user the name of the system selected and
 #  provide the list of part names available to select from

 #  output the system directory name to the (temporary) file
$HOME/convert/igesw/workdir.dat
```

```
# provide the user with the list of directories in the directory
# $HOME/convert/$sysname.

# prompt the user to select part name and echo selection to user

# construct the pathname to the selected part

#Assign this value to the global variable partdir

# Use the Unix script check_dir to find out if phase2 directory is empty
# If it is not, prompt to the user to decide whether old files are
# to be erased or left alone.

# open the data file check_dir.dat and read the result of checking
# if $phase2dir is empty

# read the first record; if none, $eof=0
# and get the value of $emptyph2 from the buffer

# close the data file check_dir.dat

# erase the data file check_dir.dat
 # evidence of previous phase2 conversion on this part.

# use the Unix script select_dir to check if the required files
# and directories are present.
# Also, clean up the phase2 directory depending on the argument $emptyph2

# open the data file select_part.dat and read the result of selecting
# part (success or fail)

# read the first record; if none, $eof=0

# close the data file select_part.dat

# erase the data file select_part.dat.

# get the value of $selectpart from the buffer
 #De-assign the value of the global variable partdir

 # make user decide whether to quit phase2 conversion or try
```

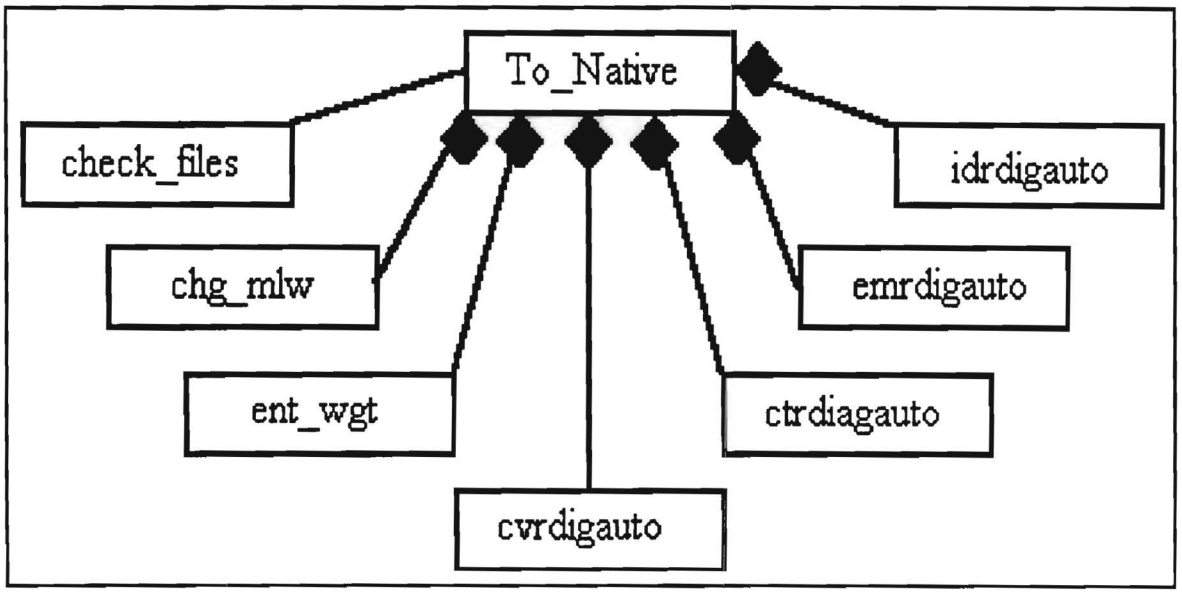
```
  # selecting part again.

  # read user input and assign value to $selectpart
#while $selectpart = continue

    # If source system is IDEAS, convert any IGES #143's in ok.igs to #144's

    # create data file for cnv_143To144.scr

    # If source system is IDEAS, combine phase1 supporting information
    # component files into a single file phase1/si.igs and copy it to
    # phase2/si.igs

    # If source system is CATIA, delete null elements from ok and
    # si files.

    # create data file for processing ok file through del_nul.scr

    # create data file for processing si file through del_nul.scr

  # end of if sourcesys=ideas or catia

# end of if (emptyph2, ne, no)

# end of if (selectpart, ne, quit)
```



TO NATIVE

```
#               IGES/Works Script: To_Native.scr
#
#  Author:  V. M. Kaladi
#           CAE/CAD Lab, Georgia Tech
#           PDCSF Project
#
#  Changed May 27, 1994 for CATIA by Gayle G. Casey.
#
#  Script to convert from IGES to target CAD format

#  define all the variables that are needed
#  Note: uses the global variable, partdir, defined in .igwin_pdcsf

# Edit the next lines to assign the name of the dedicated CAD machines

# store the name of the phase2 IGES/Works utilities directory in $phase2dir


#  get the name of the input directory in $in_dirb first and then
#  assign the name of phase2 directory for this in $in_dir

 # prompt the user with the menu to select the input IGES file

 #  read user input and assign the name of input file to $in_iges

   # get out of the menu prompt loop

 #  use the Unix script check_files to see that the $in_iges file is
 #  non-empty and return the result in the data file check_files.dat

  #  open the data file check_files.dat and read the result of checking
  #  for the presence of file $in_iges

  #  read the first record; if none, $eof=0
  #  and get the value of $yes from the buffer

  #  close the data file check_files.dat

  #  erase the data file check_files.dat
```

```
# end of while ($selectfiles, eq, "none")

# prompt the user with the menu to select the target system name

#  read user input and assign system name to $sysname

  #  Change the maximum line weight to an acceptable value for CATIA.
  # open the data file for the script chg_mlw.scr as $chg_mlw_dat
  # and write the name of the selected file into it.

  #  Change the entity weights to acceptable values for I/EMS.
  # open the data file for the script ent_wgt.scr as $ent_wgt_dat
  # and write the name of the selected file into it.
```



WRITE_SI2

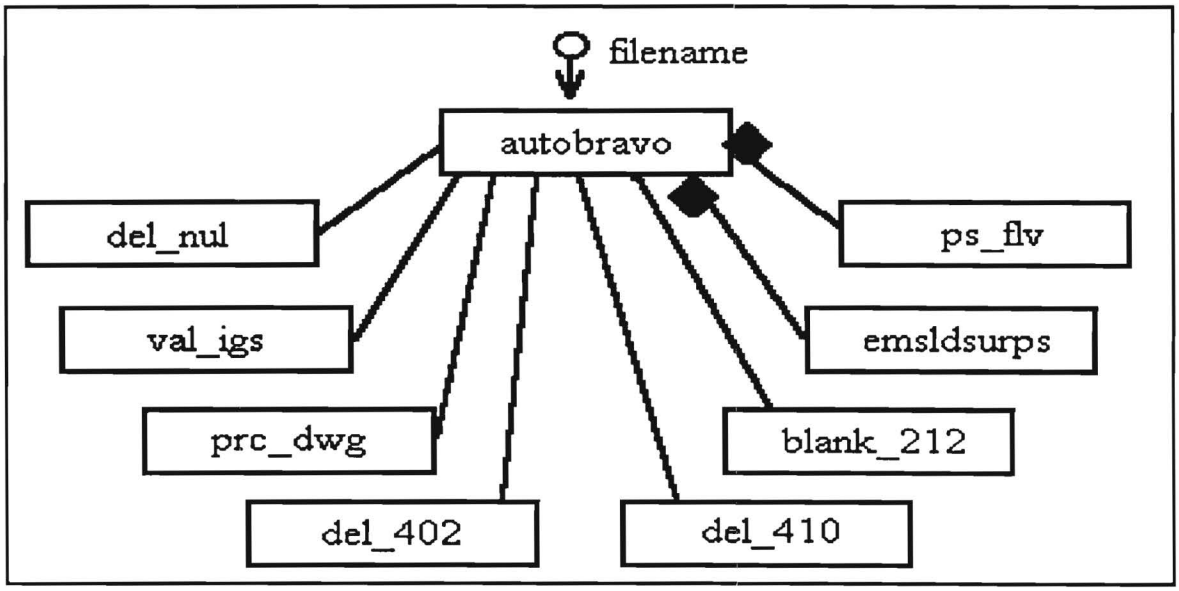
```
#               Unix Script writesi2
#  Author: V. Kaladi
#          CALS Technology Center, Georgia Tech
#          (404)-894-7777
#          e-mail: vkaladi@cad.gatech.edu
#
# This utility converts the CATIA model, SI2 in the model file,
# $HOME/convert/catia/file_name, to IGES during phase1 of assisted
# conversion from CATIA.
# Usage:
#       writesi2 file_name
#

# Define the home directory

# Define the directory for CATIA pdcsf scripts and CATIA pdcsf configuration
# files.

#Assign names to the arguments

#directory to hold the log files

# Work from the convert/catia directory

# ASSISTED is already defined as a CATIA model file by the utility
# start_catia before this utility (writeiges) is run.

# Write IIF file from $MODELTITL

# Convert IIF to IGES file
```


filename
writesi2

5 DIRECT CONVERSION SC

AUTOBRAVO

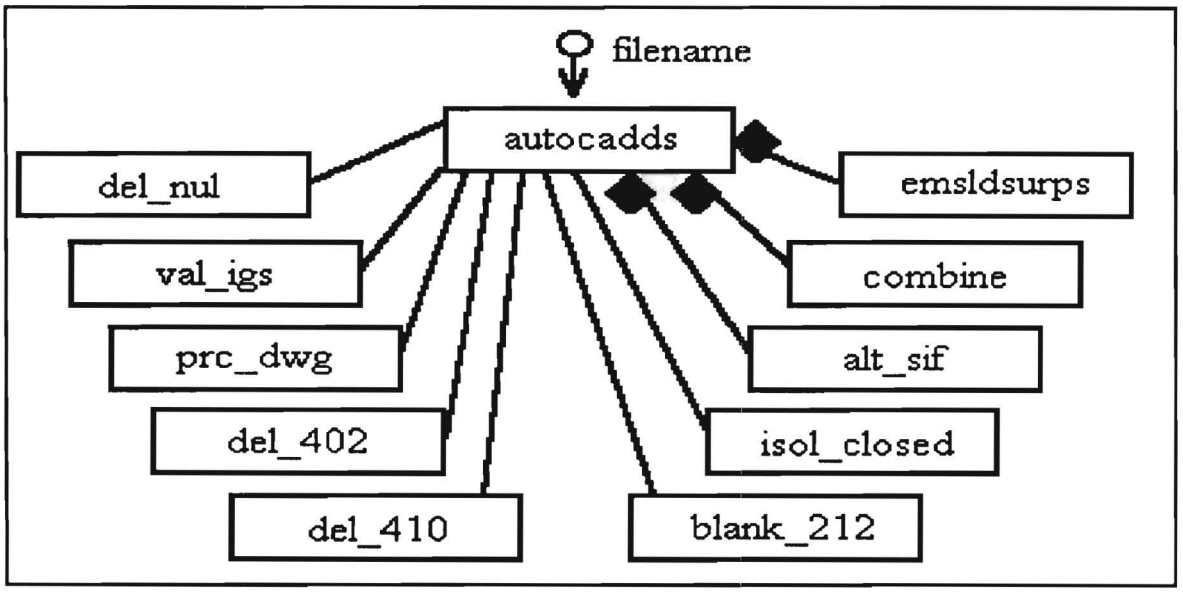
```
#               Unix Script: autobravo
#
#  Author:  V. M. Kaladi
#           CAE/CAD Lab, Georgia Tech
#           PDCSF Project
#
#  Modified April 27, 1994 by Gayle Casey to include script to delete null
#  elements from IGES file.
#
#  Unix Script to carry out the automatic conversion of one
#  IGES file to a BRAVO3 native file.
#
#  Usage:  autobravo filename
#          where
#            filename= name of the IGES file to process

# Modify the next two lines to set the names of the dedicated IGES/Works and EMS
# machines respectively.
##############################################

#Use the script del_nul.scr to delete null elements from IGES file

#Use the script val_igs.scr to validate and fix the input IGES file
#using IGES/Works capabilities.

# First test if drawings are present and process accordingly.

#Use the script del_402.scr to remove all groups

#Use the script del_410.scr to remove all independent views

# CATIA outputs all 212 elements with nonsensical values
# for the box width and height of the text.  The PDCSF
# team is trying to find a way to replace or scale these
# values so that the text is an appropriate size inside
# the model.  For the time being, the text will be blanked
# for display purposes.  If it is necessary to see the text
# it can be un-blanked within the CAD system.

#Use the script blank_212.scr to blank all 212 elements in IGES file
```


filename
autobravo
del_nul
val_igs
prc_dwg
del_402
del_410
blank_212
emsldsurps
ps_flv

AUTOCADDS

```
#               Unix Script: autocadds
#
# Author:  V. M. Kaladi
#          CAE/CAD Lab, Georgia Tech
#          PDCSF Project
#
# Modified April 27, 1994 by Gayle G Casey to include script to delete
# null elements.
#
# Unix Script to carry out the flavorings for automatic conversion of one
# IGES file to a CADDS native file.
#
# Usage:  autocadds filename
#         where
#           filename= name of the IGES file to process


# Modify the next two lines to set the names of the dedicated IGES/Works and
# EMS machines respectively.

#Use the script del_nul.scr to delete null elements from IGES file

#Use the script val_igs.scr to validate and fix the input IGES file
#using IGES/Works capabilities.

# First test if drawings are present and process accordingly.

#Use the script del_402.scr to remove all groups

#Use the script del_410.scr to remove all independent views

# CATIA outputs all 212 elements with nonsensical values
# for the box width and height of the text.  The PDCSF
# team is trying to find a way to replace or scale these
# values so that the text is an appropriate size inside
# the model.  For the time being, the text will be blanked
# for display purposes.  If it is necessary to see the text
# it can be un-blanked within the CAD system.

#Use the script blank_212.scr to blank all 212 elements in IGES file
```

```
# Added capability to handle closed trimmed surfaces, May 3, '93 Vasu Kaladi.
# For CADDS, isolate the closed trimmed surfaces and produce the alternate
# supporting information. The flavored file will now contain the
# original surface and the trimming curves in place of the trimmed
# closed surface.

# Use isol_closed.scr to pullout all the closed trimmed surfaces into the
# file closed$$.igs.

# Use alt_sif.scr to create an IGES file with the supporting information for
# the closed surfaces, if there are any.

# Use combine.scr to combine the two files and create the single file
# to replace the original IGES file.

# Erase all the temporary files.
```



AUTOCADKEY

```
#               Unix Script: autocadkey
#
#  Author:  V. M. Kaladi
#           CAE/CAD Lab, Georgia Tech
#           PDCSF Project
#
#  Modified April 27, 1994 by Gayle G Casey to include script to delete
#  null elements.
#
#  Unix Script to carry out the automatic conversion of one
#  IGES file to CADKEY
#  Note: Only a set of flavored IGES files are produced in this
#        case of direct conversion to CADKEY. These have to be
#        read manually into CADKEY
#
#  Usage:  autocadkey filename
#          where
#            filename= name of the IGES file to process

# Note: Variables HOME, HOSTNAME, rem_cmd, igesw_mc, ems_mc, ideas_mc,
# cadds_mc, bravo_mc., cadkey_mc, igwdir, scriptdir  are defined and
# exported from the script conv_to_native.

# Modify the next two lines to set the names of the dedicated IGES/Works and EMS
# machines respectively.

#Use the script del_nul.scr to delete null elements from IGES file

#Use the script val_igs.scr to validate and fix the input IGES file
#using IGES/Works capabilities.

# First test if drawings are present and process accordingly.

#Use the script del_402.scr to remove all groups

#Use the script del_410.scr to remove all independent views

#Use the script iso_ind.scr to isolate indirect surfaces into the file
#named "indirect.igs"
```

Use the script cnv_pcs.scr to convert all ok elements into PCS
if ok elements exist.

#Use the script alt_sif.scr to extract supporting information for
#indirect surfaces (contained in the file "indirect.igs")

Use the script cnv_pcs.scr to convert all alternate
supporting information for indirect surfaces into PCS

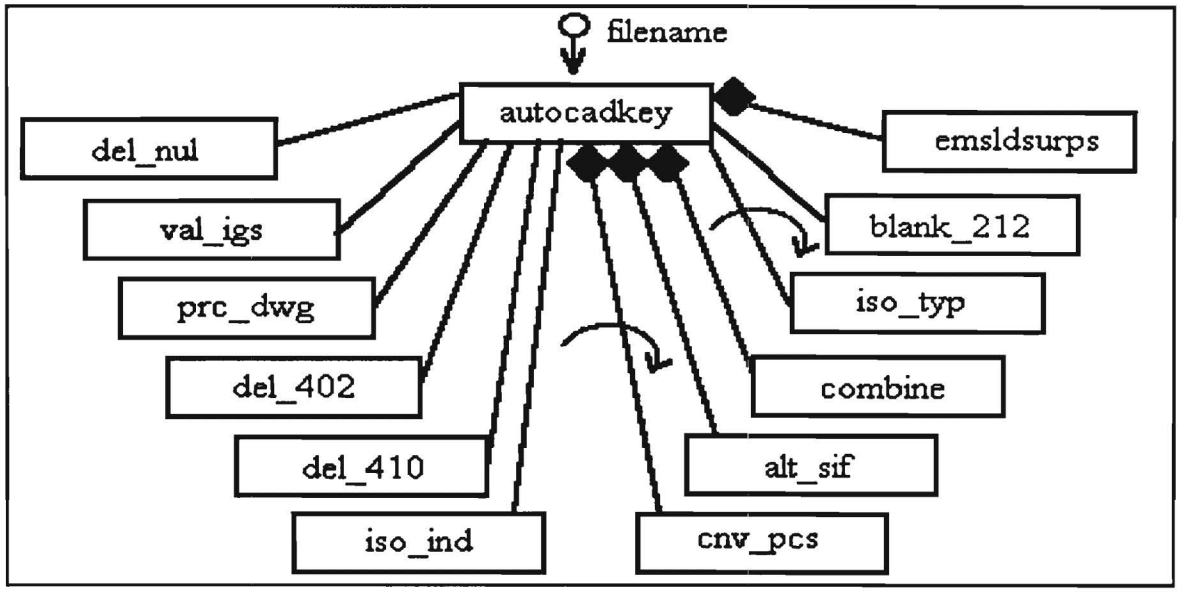
#Use the script combine.scr to combine the alternative supporting
#information for indirect surfaces back into the original file.

#Use the script iso_typ.scr to remove all 128's

#Use the script iso_typ.scr to remove all 126's

CATIA outputs all 212 elements with nonsensical values
for the box width and height of the text. The PDCSF
team is trying to find a way to replace or scale these
values so that the text is an appropriate size inside
the model. For the time being, the text will be blanked
for display purposes. If it is necessary to see the text
it can be un-blanked within the CAD system.

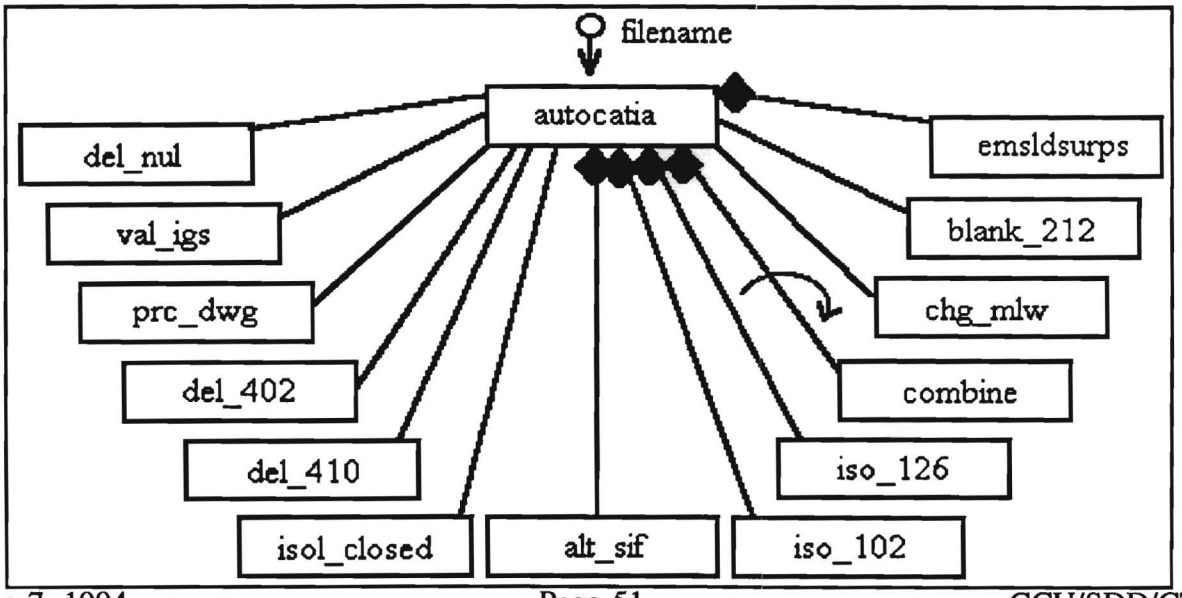
#Use the script blank_212.scr to blank all 212 elements in IGES file



AUTOCATIA

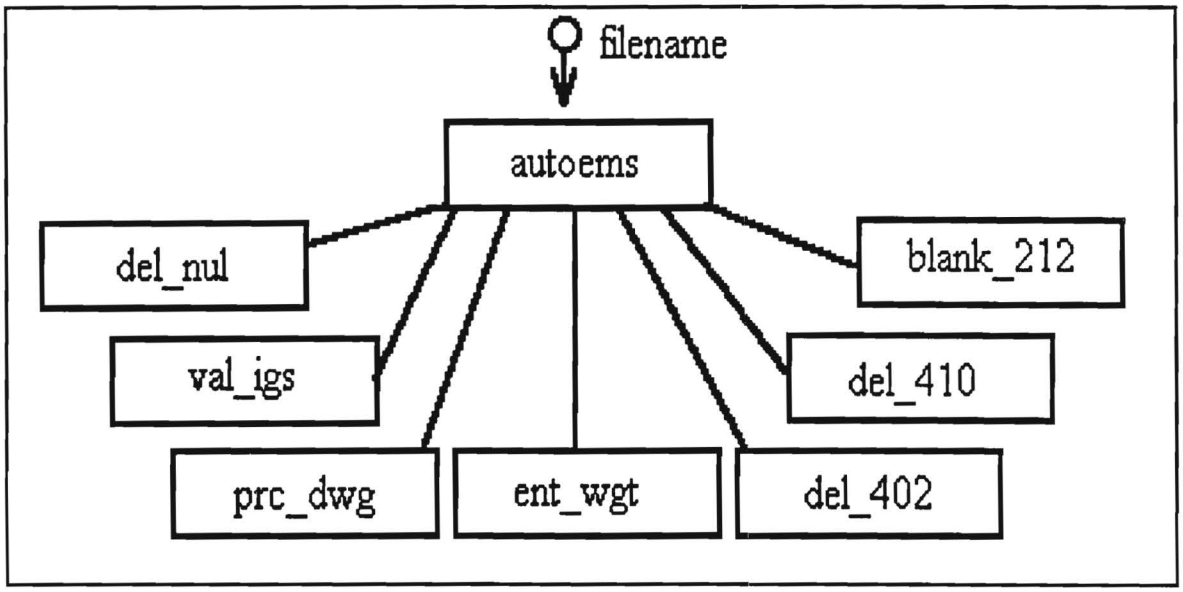
```
#~Name       : autocatia
#~Originator: Gayle G. Casey and Vasu Kaladi
#            : CALS Technology Center, Georgia Tech
#            : For support:
#            :  Gayle G. Casey
#            :  (404) 894-9811
#            :  casey@cad.gatech.edu
#~Latest     : Wednesday, April 27, 1994
#~Product    : CATIA VERSION 3  RELEASE 2.5
#~Command    : This utility is called only by another utility.
#~Comments   : The autocatia script is a UNIX Script to carry out the
#            : flavorings for automatic conversion of one IGES file to
#            : a CATIA native file.

# Modify the next two lines to set the names of the dedicated IGES/Works and
# CATIA machines respectively.

#Use the script del_nul.scr to delete null elements from IGES file

#Use the script val_igs.scr to validate and fix the input IGES file
#using IGES/Works capabilities.

# First test if drawings are present and process accordingly.

#Use the script del_402.scr to remove all groups

#Use the script del_410.scr to remove all independent views

# Isolate the closed trimmed surfaces and produce the alternate
# supporting information. The flavored file will now contain the
# original surface and the trimming curves in place of the trimmed
# closed surface. The trimming curves will be broken down further
# from a 102, Composite Curve to the underlying curves, generally
# all 126, B-Spline Curve.

# Use isol_closed.scr to pullout all the closed trimmed surfaces into the
# file closed$$.igs.

# Use alt_sif.scr to create an IGES file with the supporting information for
# the closed surfaces, if there are any.
```

```
# Use the iso_102.scr to isolate all 102 elements from $closed_iges
# into type_102.igs.

# Use the iso_126.scr to isolate all 126 elements from type_102.igs
# into type_126.igs.

# Use combine.scr to combine the two files and create the single file
# to replace the original IGES file.

# Use combine.scr to combine the two files and create the single file
# to replace the original IGES file.

# Use chg_mlw.scr to change the value of the maximum line width in
# the global section.

# Erase all the temporary files.

# CATIA outputs all 212 elements with nonsensical values
# for the box width and height of the text. The PDCSF
# team is trying to find a way to replace or scale these
# values so that the text is an appropriate size inside
# the model. For the time being, the text will be blanked
# for display purposes. If it is necessary to see the text
# it can be un-blanked within the CAD system.

#Use the script blank_212.scr to blank all 212 elements in IGES file
```



AUTOEMS

```
#               Unix Script: autoems
#
# Author:  V. M. Kaladi
#          CAE/CAD Lab, Georgia Tech
#          PDCSF Project
#
# Modified April 27, 1994 by Gayle G. Casey to include script to delete
# null elements and script to change entity weights.
#
# Unix Script to carry out the automatic conversion of one
# IGES file to an EMS native file.
#
# Usage:  autoems filename
#         where
#           filename= name of the IGES file to process

#Use the script del_nul.scr to delete null elements from IGES file

#Use the script val_igs.scr to validate and fix the input IGES file
#using IGES/Works capabilities.

# First test if drawings are present and process accordingly.

# Use the script ent_wgt.scr to change entity weights
# if IGES file is from CATIA.

#Use the script del_402.scr to remove all groups

#Use the script del_410.scr to remove all independent views

# CATIA outputs all 212 elements with nonsensical values
# for the box width and height of the text.  The PDCSF
# team is trying to find a way to replace or scale these
# values so that the text is an appropriate size inside
# the model.  For the time being, the text will be blanked
# for display purposes.  If it is necessary to see the text
# it can be un-blanked within the CAD system.

#Use the script blank_212.scr to blank all 212 elements in IGES file
```

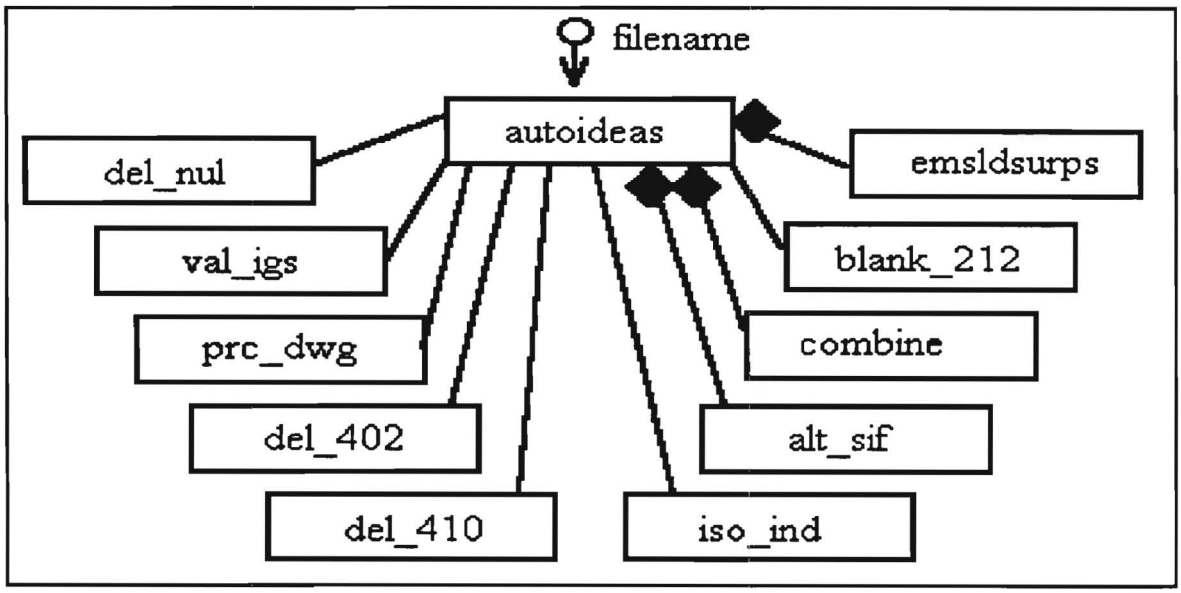

filename
autoems
del_nul
blank_212
val_igs
del_410
prc_dwg
ent_wgt
del_402

AUTOIDEAS

```
#               Unix Script: autoideas
#
#  Author:  V. M. Kaladi
#           CAE/CAD Lab, Georgia Tech
#           PDCSF Project
#
#  Modified April 27, 1994 by Gayle G. Casey to include script to delete
#  null elements.
#
#  Unix Script to carry out the automatic conversion of one
#  IGES file to an IDEAS native file.
#
#  Usage:  autoideas filename
#          where
#            filename= name of the IGES file to process

# Note: Variables HOME, HOSTNAME, rem_cmd, igesw_mc, ems_mc, ideas_mc,
# cadds_mc, bravo_mc., cadkey_mc, igwdir, scriptdir  are defined and
# exported from the script conv_to_native.

#Use the script del_nul.scr to delete null elements from IGES file

#Use the script val_igs.scr to validate and fix the input IGES file
#using IGES/Works capabilities.

# First test if drawings are present and process accordingly.

#Use the script del_402.scr to remove all groups

#Use the script del_410.scr to remove all independent views

#Use the script iso_ind.scr to isolate indirect surfaces into the file
#named "indirect.igs"

#Use the script alt_sif.scr to extract supporting information for
#indirect surfaces (contained in the file "indirect.igs")

#Use the script combine.scr to combine the alternative supporting
#information for indirect surfaces back into the original file.
```

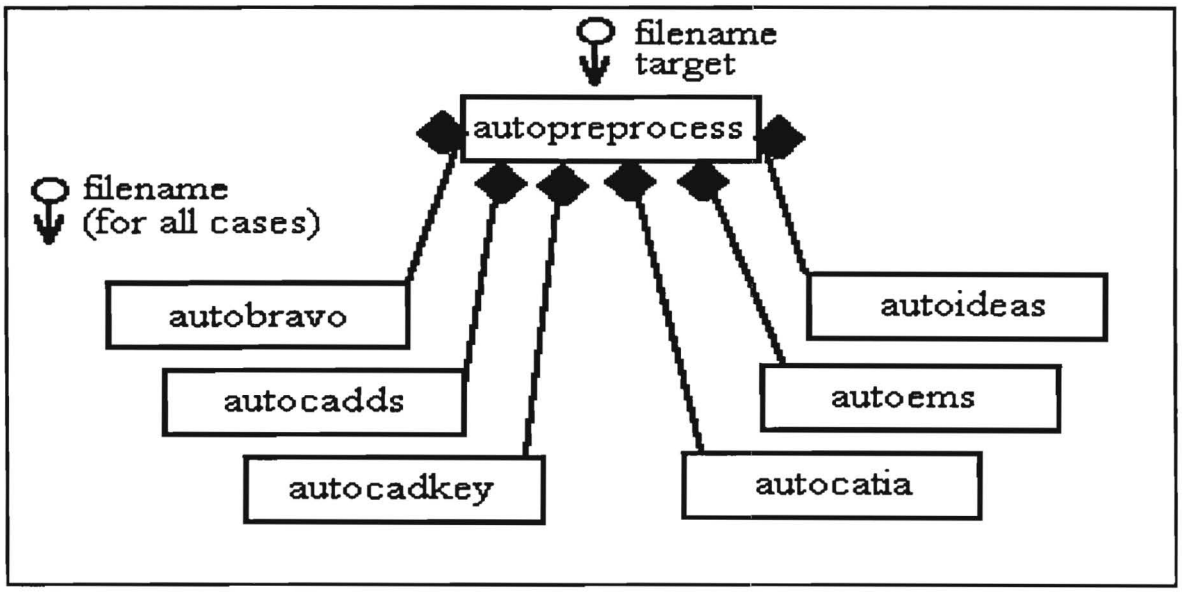
CATIA outputs all 212 elements with nonsensical values
for the box width and height of the text. The PDCSF
team is trying to find a way to replace or scale these
values so that the text is an appropriate size inside
the model. For the time being, the text will be blanked
for display purposes. If it is necessary to see the text
it can be un-blanked within the CAD system.

#Use the script blank_212.scr to blank all 212 elements in IGES file



AUTOPREPROCESS

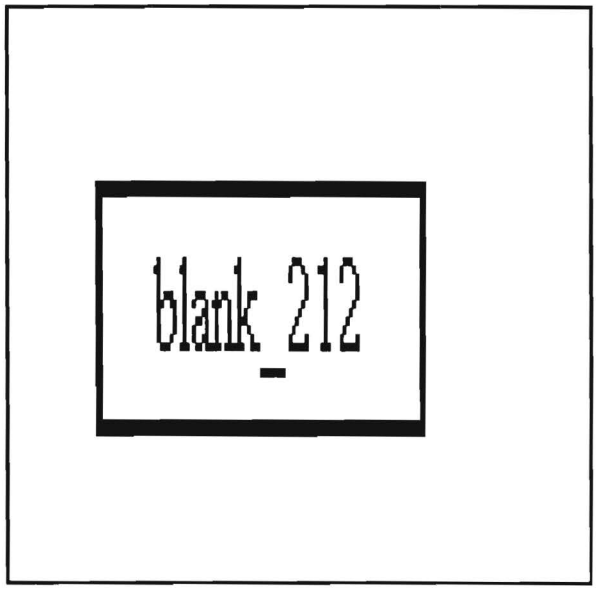
```
#               Unix Script: autopreprocess
#
# Author:  V. M. Kaladi
#          CAE/CAD Lab, Georgia Tech
#          PDCSF Project
#
# Modified April 11, 1994 by Gayle Casey to include CATIA.
#
# Unix Script to carry out the automatic conversion of one
# IGES file to a native file on a specified target CAD system.
#
# Usage:  autopreprocess filename target
#         where
#           filename= name of the IGES file to process
#           target= name of the target CAD system
#                   brv for BRAVO3
#                   cdd for CADDS
#                   ems for EMS
#                   ide for IDEAS
#                   cdk for CADKEY
#                 cat for CATIA
#
# set the name of the machine on which "parse.c" is compiled and the
# location of this executable

#copy the IGES file, $1, to the subdirectory "flavor"

#change to the directory holding the flavored IGES files

# since rsh puts autoconv_parse.dat in the home directory, move it back here.
```


filename
target
autopreprocess
filename
(for all cases)
autobravo
autocadds
autocadkey
autoideas
autoems
autocatia

BLANK_212.SCR

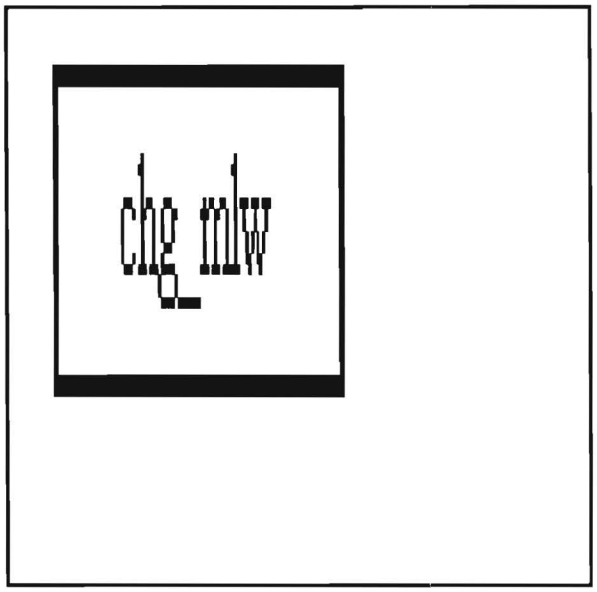
```
#~Name       : blank_212.scr
#~Originator: Gayle G. Casey
#            : CALS Technology Center, Georgia Tech
#            : For support:
#            :  Gayle G. Casey
#            :  (404) 894-9811
#            :  casey@cad.gatech.edu
#~Latest     : Thursday, May 5, 1994
#~Product    : IGES/Works 1.4
#~Command    : This utility is called only by another utility.
#~Comments   : CATIA outputs all 212 elements with nonsensical values
#            : for the box width and height of the text.  The PDCSF
#            : team is trying to find a way to replace or scale these
#            : values so that the text is an appropriate size inside
#            : the model.  For the time being, the text will be blanked
#            : for display purposes.  If it is necessary to see the text
#            : it can be un-blanked within the CAD system.
#
#  The name of the IGES file is passed via file named blank_212.dat
#
#  Files created:  original iges file with all 212 entities blanked
#
# define all the variables that are needed !

# variables needed for incorporation into direct conversion

# set the home directory name

# set the directory name for the direct conversion utilities and the
# corresponding temporary data files in $direct_dir.

# open the file $blank_212_dat file
#
# read the first record - the name of the iges file
#
# parse the record and assign the names of original IGES file
#
#  close the data file $blank_212_dat

# allocate a model named test - temporary
```

```
#
# read the iges file in to this model
#
# check to see if file is from CATIA
#
#
# If file is from CATIA, select all 212 entities and blank them.
#
# If the file is not from CATIA, say so and end program.
#
# free all the allocated models
```



CHG_MLW.SCR

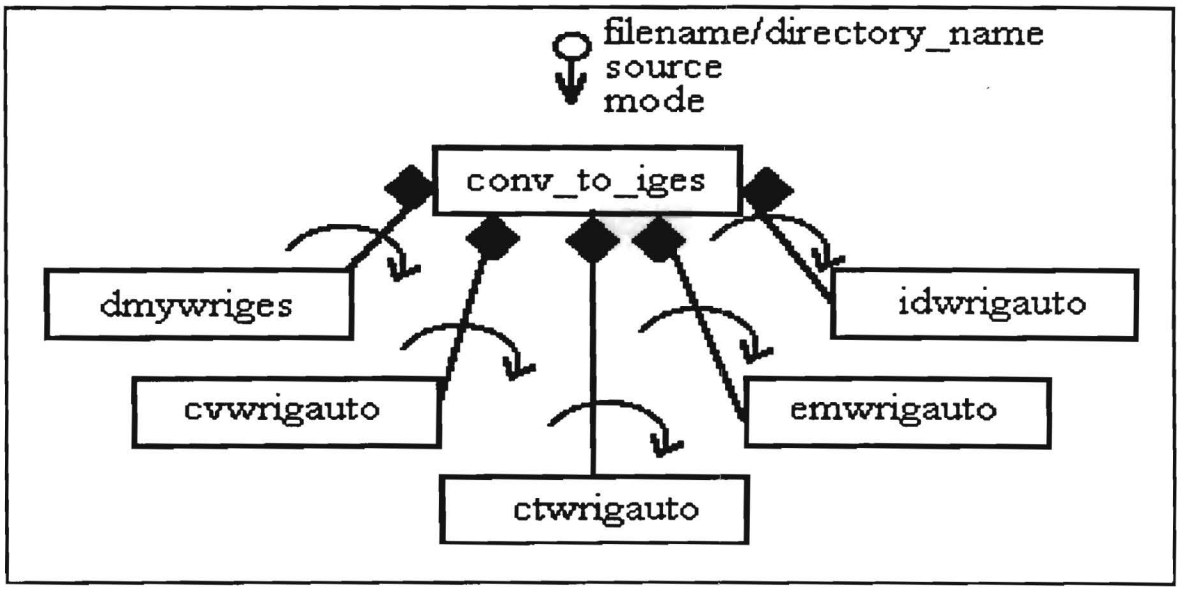
```
#~Name      : chg_mlw.scr
#~Originator: Gayle G. Casey
#           : CALS Technology Center, Georgia Tech
#           : For support:
#           :  Gayle G. Casey
#           :  (404) 894-9811
#           :  casey@cad.gatech.edu
#~Latest    : Thursday, May 5, 1994
#~Product   : IGES/Works 1.4
#~Command   : This utility is called only by another utility.
#~Comments  : The chg_mlw script is an IGES/Works Script to change the
#    : maximum line width in the global section of an IGES file.
#
# define all the variables that are needed !
#
# variables needed for incorporation into direct conversion

# set the directory name for the direct conversion utilities and the
# corresponding temporary data files in $direct_dir.

# open the file $chg_mlw_dat file
#
# read the first record - the name of the iges file
#
# parse the record and assign the file_name
#
#  close the data file $val_igs_dat
#
# allocate a model named test - temporary
#
# read the iges file into this model
#
# Issue the change line weight command
#
# overwrite the old iges file with the results of conversion
#
# free all the models
```



CONV_TO_IGES

```
#               Unix Script: conv_to_iges
#
#  Author:  V. M. Kaladi
#           CAE/CAD Lab, Georgia Tech
#           PDCSF Project
#
#  Modified May 3, 1994 for CATIA.
#
#  Unix Script to carry out the automatic conversion of one or more
#  Native files on a specified source CAD system into a set of IGES files .
#
#  Usage:  conv_to_iges filename/directory_name source mode
#          where
#            filename= name of the Native file to process, if mode=single
#                      This file is to be placed in the directory,
#                      $HOME/convert/[bravo|cadds|ems|ideas|cadkey|catia]
#            directory_name = name of the directory holding the native files
#                             to be converted, if mode=multiple. This directory
#                             is to placed in the directory
#                             $HOME/convert/[bravo|cadds|ems|ideas|cadkey|catia]
#            source= name of the source CAD system
#                      brv for BRAVO3
#                      cdd for CADDS
#                      ems for EMS
#                      ide for IDEAS
#                      cdk for CADKEY
#                      cat for CATIA
#            mode= single, multiple or restart

# pdcsf1, pdcsf2, pdcsf3 use "rcmd" instead of "rsh"

# edit the names of the dedicated CAD machines below:

#give names to the command line arguments

# Select the directory name based on source system

# setup directories based on mode of operation

#create a temporary directory to hold a copy of the
```

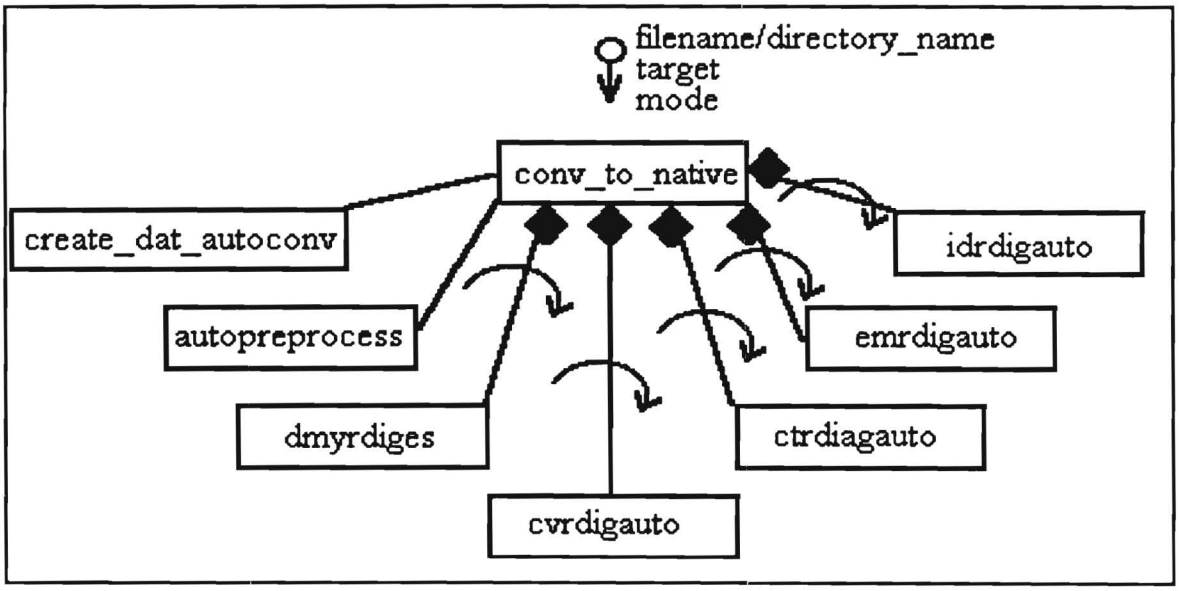
```
#single native file and proceed just as for multiple
#mode of operation.
#
# The restart mode is an unpublished feature designed for the PDCSF
# project.  This feature is not for use with CATIA.  For more information
# please call support personnel.
#
#use restart file from previous run

#obtain the listoffiles as a file with only one entry per source CAD file.
#This is achieved by cutting out the extensions in the filenames and then
#picking out only the unique names left

#create the restart file by first making a copy of $listoffiles

# For CATIA, $listoffiles is not used

#run the native system write_IGES utility

#Process for multiple mode

#run the native system write_IGES utility

#edit restart file to remove this IGES file from the list of files

#delete the restart file if it is empty

#delete any old directory by this name

#move the IGES files directory from the temporary directory to this
#directory. Append .iges to the input filename to get the name of this
#directory.
```


filename/directory_name
source
mode
conv_to_iges
dmywriges
idwrigauto
cvwrigauto
emwrigauto
ctwrigauto

CONV_TO_NATIVE

```
#               Unix Script : conv_to_native
#
# Author:  V. M. Kaladi
#          CAE/CAD Lab, Georgia Tech
#          PDCSF Project
#
# Modified by Gayle Casey, Tuesday, April 26, 1994 to include error checking.
# Modified by Gayle Casey, Thursday, April 21, 1994 to include CATIA.
#
# Unix Script to carry out the automatic conversion of one or more
# IGES files to a set of native files on a specified target CAD system.
#
# Usage:  conv_to_native filename/directory_name target mode
#         where
#           filename= name of the IGES file to process, if mode=single
#                     This file is to be placed in the directory,
#                     $HOME/convert/[bravo|cadds|ems|ideas|cadkey]
#           directory_name = name of the directory holding the IGES files
#                            to be converted, if mode=multiple. This directory
#                            is to placed in the directory
#                            $HOME/convert/[bravo|cadds|ems|ideas|cadkey]
#           target= name of the target CAD system
#                     brv for BRAVO3
#                     cdd for CADDS
#                     ems for EMS
#                     ide for IDEAS
#                     cdk for CADKEY
#                     cat for CATIA
#           mode= single, multiple or restart

# pdcsf1, pdcsf2, pdcsf3 use "rcmd" instead of "rsh"

# edit the names of the dedicated CAD machines below:

#give names to the command line arguments

# Select the directory name based on target system

# setup directories based on mode of operation
```

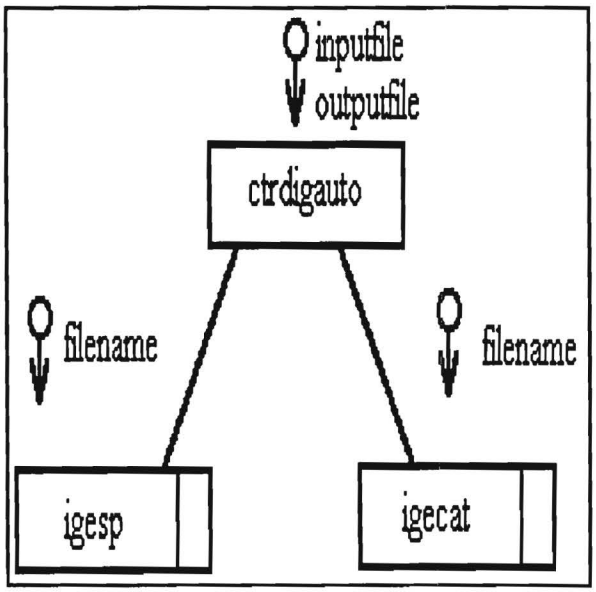
```
#create a temporary directory to hold a copy of the single
#IGES file and proceed just as for multiple mode of operation

# create report file

#create data file for automatic conversion as $listoffiles

#create the restart file by first making a copy of $listoffiles

#Process for multiple mode

#append the logfile produced during the preprocessing stage to
#the report file

#read the flavored IGES file into the target CAD system
# if running on pdcsf1, pdcsf2, pdcsf3 use rcmd instead of rsh

# erase the data file for processing engineering drawings

#edit restart file to remove this IGES file from the list of files

#delete the restart file if it is empty

#delete any old directory by this name

#move the native files directory from the temporary directory to this
#directory. Append .native to the input filename to get the name of this
#directory.

#save the flavored IGES file, if target is CADKEY
#save the flavored IGES file, if target is BRAVO also (Vasu Kaladi, Jul 22, '93)
```


filename/directory_name
target
mode
conv_to_native
create_dat_autoconv
idrdigauto
autopreprocess
emrdigauto
dmyrdiges
ctrdiagauto
cvrdigauto

CTRDIGAUTO

```
#               Unix Script ctrdigauto
#  Author: V. Kaladi
#          CALS Technology Center, Georgia Tech
#          (404)-894-7777
#          e-mail: vkaladi@cad.gatech.edu
#
# This script reads one IGES file into CATIA. The 1st argument is the
# full path name to the IGES file (without the extension). The 2nd argument
# is the name of the directory to hold the CATIA model followed by the
# name of the CATIA model.
#

# Work from the convert/catia directory

# Define the directory for CATIA pdcsf scripts and CATIA pdcsf configuration
# files.

# Assign names to the arguments

# $HOME/CATIA/batchread is defined to be a CATIA model file in the SYSDYN
# file, $ctacfgdir/sysdyn. Set up a symbolic link to point
# $HOME/CATIA/batchread to $MODELDIR to define $MODELDIR as a CATIA
# model file.
# This is to avoid problems with long pathnames in SYSDYN file.

# Define MODELDIR as a CATIA model file with the DSNAME, BATCHREAD

# Convert the IGES file into an IIF file. The IIF file is left
# in the same directory as the input IGES file.

# move the .iif file to the current directory with the name $$.iif since
# file names longer than 44 characters are not acceptable in igecat.opt

# make an igecat option file for this.

# run igecat  and place the log file in $MODELDIR

# Remove the symbolic link from $HOME/CATIA/batchread to $MODELDIR

# edit sysdyn file to remove MODELFILE as a CATIA model file
```

erase intermediate files



CTWRIGAUTO

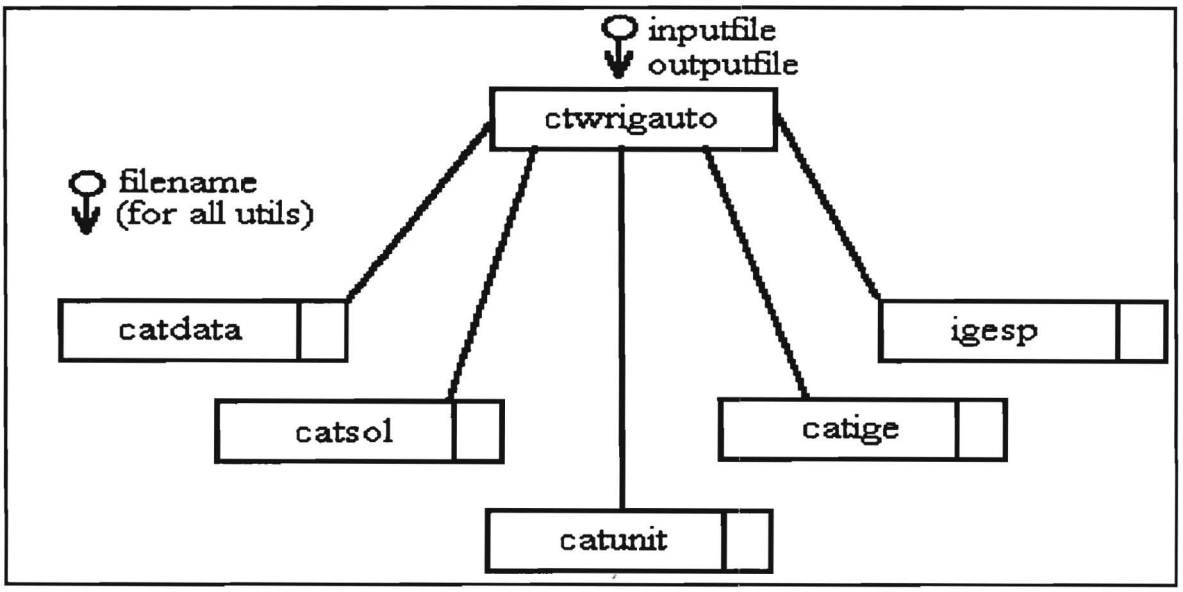
```
#               Unix Script ctwrigauto
#  Author: V. Kaladi
#          CALS Technology Center, Georgia Tech
#          (404)-894-7777
#          e-mail: vkaladi@cad.gatech.edu
#
# This utility converts all the CATIA models present in the directory
# given as the 1st argument to IGES files in the directory given as
# the second argument.
# 1st argument = name of directory holding CATIA native files
# 2nd argument = name of directory to hold the output IGES files
#

# Define the directory for CATIA pdcsf scripts and CATIA pdcsf configuration
# files.

#Assign names to the arguments
#directory to hold the log files

# Work from the convert/catia directory

# $HOME/CATIA/batchwrite is defined to be a CATIA model file in the SYSDYN
# file, $ctacfgdir/sysdyn. Set up a symbolic link to point
# $HOME/CATIA/batchwrite to $MODELDIR to define $MODELDIR as a CATIA
# model file.
# This is to avoid problems with long pathnames in SYSDYN file.

# Run CATDATA utility on the CATIA model file (directory) to get the list
# of models as tmp$$

# Edit the file, tmp$$, to extract only CATIA model names into the scratch
# file catlist$$

# Loop through the list of CATIA models and convert each to IGES

  # Copy $model to model$$.  Use model$$ for further processing.

  # convert solids to surfaces, if present.

  # convert units to inches.
```

Write IIF file from $MODELTITL

Convert IIF to IGES file

Delete the temporary CATIA model, $MODELTITL

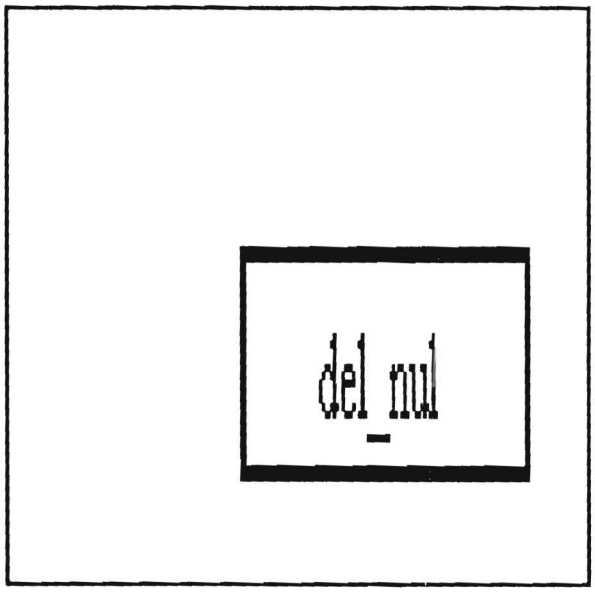
Remove the symbolic link from $HOME/CATIA/batchwrite to $MODELDIR



DEL_NUL.SCR

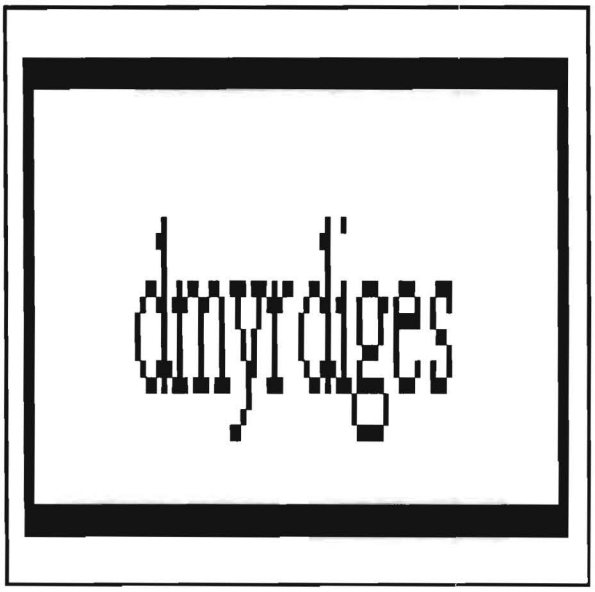
```
#~Name      : del_nul.scr
#~Originator: Gayle G. Casey and Vasu Kaladi
#           : CALS Technology Center, Georgia Tech
#           : For support:
#           :  Gayle G. Casey
#           :  (404) 894-9811
#           :  casey@cad.gatech.edu
#~Latest    : Tuesday, April 12, 1994
#~Product   : IGES/Works 1.4
#~Command   : This utility is called only by another utility.
#~Comments  : The del_nul script is an IGES/Works Script to delete
#           : null entities from IGES files.  Specifically, nulls
#           : of form 6004 and 6007 output from CATIA.

#  define all the variables that are needed
#
# variables needed for incorporation into direct conversion
#
# set the directory name for the direct conversion utilities and the
# corresponding temporary data files in $direct_dir.

# first open the file del_nul_dat
#the first line is the name of the iges file to be processed

# read one record at a time (read the iges file name)

#parse the record that has been read

#close the file - you are done with the file.

#  allocate a model named test

#  read the iges file

#  select all the null elements

# if there are null entities in the file, delete them

#  output all the remaining entities to the input IGES
#  file $in_iges(overwite).
```

free the current model



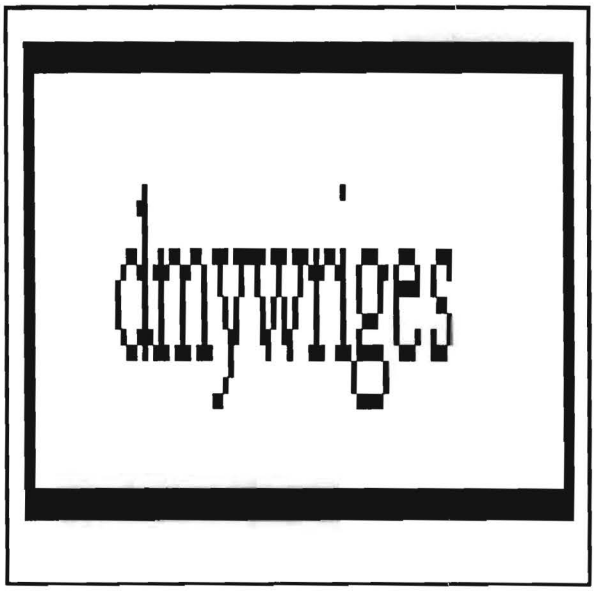
DMYRDIGES

```
#               Unix Script: dmyrdiges
#
# Author:  V. M. Kaladi
#          CALS Technology Center, Georgia Tech
#          PDCSF Project
#
# Unix Script to output a message in place of the utility to
# read the flavored IGES file into the target CAD system.
```



DMYWRIGES

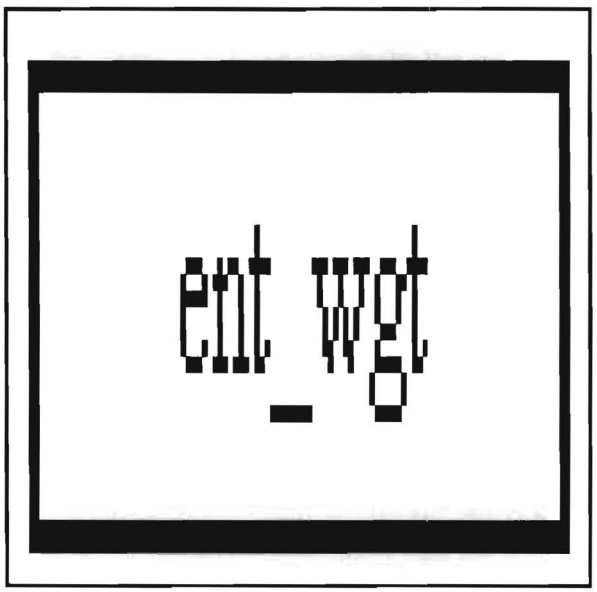
```
#               Unix Script: dmywriges
#
# Author:  Gayle G. Casey
#          CALS Technology Center, Georgia Tech
#          PDCSF Project
#
# Unix Script to output a message in place of the utility to
# write an IGES file from a native CAD file.
```



ENT_WGT.SCR

```
#~Name      : ent_wgt.scr
#~Originator: Gayle G. Casey
#           : CALS Technology Center, Georgia Tech
#           : For support:
#           :  Gayle G. Casey
#           :  (404) 894-9811
#           :  casey@cad.gatech.edu
#~Latest    : Wednesday, April 13, 1994
#~Product   : IGES/Works 1.4
#~Command   : This utility is called only by another utility.
#~Comments  : The ent_wgt script is an IGES/Works Script to change the
#           :  line weight of each independent entity.
#
#  The name of the IGES file is passed via file named ent_wgt.dat
#
#  Files created:  original iges file with all entity weights
#           changed to 1.
#
# define all the variables that are needed !
#
# variables needed for incorporation into direct conversion

# set the home directory name

# set the directory name for the direct conversion utilities and the
# corresponding temporary data files in $direct_dir.

############ SECTION TO READ FROM DAT FILE ##################
# open the file $ent_wgt_dat file
#
# read the first record - the name of the iges file
#
# parse the record and assign the names of original IGES file
#
#  close the data file $ent_wgt_dat
############ END OF SECTION TO READ FROM DAT FILE ###############

# allocate a model named test - temporary
#
# read the iges file in to this model
```

#
check to see if file is from CATIA
#
If file is from CATIA, select all independent entities and
change the entity weight to 1 for all of them.
#
If the file is not from CATIA, say so and end program.
#
free all the allocated models



PDCSF

Product Data Conversion & Storage Facility

Assisted Conversion From CATIA

User's Guide

Prepared by
CALS Technology Center
Georgia Institute of Technology

PDCSF

Product Data Conversion
&
Storage Facility

Table of Contents:

Assisted Conversion from CATIA
Process Overview

The goal of this assisted conversion process is the accurate transfer of 3-D geometry from CATIA to another CAD system. This transfer can be broken into two distinct phases. During the first phase the CATIA model is translated to IGES format. The second phase consists of converting the IGES file format into a CAD model in another file format. The assisted conversion process has been designed to identify, isolate and compensate for information losses that may occur during these two distinct steps.

It is necessary for the person performing the conversion to be familiar with CAD technology and, in specific, the particular system under consideration. It is not the objective of PDCSF to train users in the use of any particular CAD system. It is important to note, in the case of CATIA, that the filename syntax is such that CATIA model names do not correspond to the filenames in UNIX. It is possible to use the UNIX command "strings" to look at the ASCII strings in the map file accompanying the fic* files to find out which fic* file corresponds to which CATIA model. If this filename terminology is unclear, please consult CATIA documentation.

The phase one conversion is performed primarily in the CATIA environment. Some of the utilities used in this phase are designed to run in the UNIX environment, while others have been designed to operate within the CATIA application. Utilities developed for use in the UNIX environment can be invoked by typing the utility name at the UNIX prompt and pressing the return key. Utilities developed for use within the CATIA application can be invoked by either selecting directly from the menu with the mouse or by pressing the middle mouse button in the menu zone and releasing the button over the appropriate menu selection. A detailed explanation of these phase one utilities and their native environments can be found in the document titled "Utilities Developed For CATIA Programming Language."

If the user finds that an error has occurred during phase one of the conversion process, he should complete the following procedure. First, take note of the error that was committed for future avoidance. Next, complete the current step of the conversion process. After completing this step, the user should exit CATIA if he is within the application. At the UNIX prompt, he should execute the rst_catia (restart CATIA) utility. This utility will remove any intermediate files that have been created during the phase one conversion. At this stage it is safe to begin phase one again from step three.

Phase two of the conversion process takes place primarily within

the IGES\Works application. To execute a utility within IGES\Works, it is necessary to select the "User Functions" button and then select the button named for the desired utility. It is necessary to select the "User Functions" button every time a utility is executed. A detailed explanation of these phase two utilities can be found in the document titled "Utilities Developed for Phase II Conversion."

At some points in the second phase, it may be necessary to have other processes running concurrently with IGES\Works. The user can open a new window by clicking on the background (outside the IGES\Works window) and opening a new or existing UNIX shell. The user can view the log file from this UNIX shell.

If the user finds that an error has occurred during phase two of the conversion process, he should complete the following procedure. First, take note of the error that was committed for future avoidance. Next, complete the current step of the conversion process. After completing this step, the user should select the "Restart" menu button from the "User Functions" menu in IGES\Works. This utility will remove any intermediate files that have been created during the phase two conversion. At this stage it is safe to begin phase two again from step one.

It is absolutely crucial that the user is aware of the contents of the model at all stages of the conversion process. Without a thorough knowledge of the model, it is impossible to accurately identify, isolate and compensate for losses that may occur. The user must take the time to visually examine all parts of the model. The functionality of all of the CAD systems and IGES\Works provides the capability to zoom in and out on a model and to change the orientation for better viewing. It is recommended that the user employ these devices to gain a full comprehension of the contents of the model.

In some cases, it may be necessary to perform a mathematical form conversion during phase two of the assisted conversion process. CATIA employs the NURB form which can be rejected or distorted by systems employing the Parametric form. In the case of CADKEY, the math conversion must be performed.

Assisted Conversion From CATIA

This document outlines the steps for converting from CATIA format to another CAD format.

It is assumed that the model for conversion is in the transparent file system under:

$HOME/convert/catia.

The model will consist of a directory named file_name under which there will be a file named map and a file named ficx. If this is not the case, it is necessary that the user obtain this model from the PDCSF repository, or use the CATIA IMPORT/EXPORT utilities to obtain the model from another site or from an external media.

PHASE ONE: NATIVE TO IGES PHASE

STEP 1: LOGIN TO DEDICATED CATIA MACHINE

User will then be in UNIX.

STEP 2: RUN THE ORG_FILE UTILITY IN UNIX

Change the working directory to $HOME/convert/catia.

Type the command:

org_file file_name

As part of the ORG_FILE process, the directory structure given in the figure titled "File Naming Conventions - Conversion From CATIA" is created.

STEP 3: WRITE THE IGES FILE FROM THE CATIA MODEL IN UNIX

In a terminal window, from the directory $HOME/convert/catia, type:

writeiges file_name

This will write the IGES file from the model. The IGES file is created after the units are converted to inches, if necessary, and any solid elements present are broken into surface and wireframe elements. Necessary flavorings are applied to the IGES file.

STEP 4: RUN THE START_CATIA UTILITY IN UNIX

Verify that the appropriate window environment has been initiated.

First change the working directory to $HOME/convert/catia. The user invokes the UNIX procedure, START_CATIA that starts the CATIA application.

a) This command is invoked as

start_catia file_name phase1

This action starts up CATIA.

b) Select COLD START.

c) Press the KEY permanent function.

d) Select the keyboard named PDCSF to access the customized function keyboard for assisted conversion.

e) Select FILE.

f) Key in /M DISPLAY.

The CATIA model is displayed in the isometric view and autoscaled in the CATIA graphics window.

Visually examine the contents of the CATIA model. The user must have a complete understanding of the CATIA model before proceeding to the next step.

If the displayed geometry is an engineering drawing, stop here. The assisted conversion process does not apply to an engineering drawing. Use the direct conversion procedure.

At the end of this step, the dedicated CATIA machine will have the CATIA model displayed on it.

On-line instructions for phase I assisted conversion may be accessed at any time, within CATIA, by keying in /M HELPAST.

STEP 5: RUN THE SUMMARY UTILITY IN UNIX

In a terminal window, from the directory $HOME/convert/catia, type:

summary file_name

This utility will inform the user of the contents of the CATIA model, OK. This step may be repeated as desired.

STEP 6: VIEW THE LOG FILE IN UNIX

The log file from the writeiges operation is stored in the file $HOME/convert/catia/file_name/phase1/file_name.log. This log file can be viewed using any text editor.

Are the problem elements identified?
Is the internal id of the problem element available?
In general, such internal ids are not available in a CATIA log file.

The log file will indicate the following:

Has any random error occurred?
(list of internal elements that failed)

Be aware of the contents of the model.

STEP 7: RUN THE COMPARE UTILITY IN CATIA

The CATIA part, OK, is already loaded and displayed on the dedicated CATIA machine.

Activate IGES/Works on the dedicated CATIA machine or mobile x-terminal. View file_name.igs in IGES/Works. This can be accomplished by running START_IW inside UNIX.

Now, key in /M COMPARE to invoke this utility.

At this stage, the user should visually compare the model displayed on the CATIA screen against that displayed on the IGES/Works screen. The comparison is a time consuming process. The purpose is to identify any surface that did not get transferred.

Each face and surface in the CATIA model is highlighted one by one and the user is prompted to select those that were lost. This utility will only accept yes/no answers from the built-in yes/no buttons and will not accept yes/no or y/n responses from the keyboard.

STEP 8: RUN THE ISOLATE UTILITY IN CATIA

Key in /M ISOLATE. This will produce two separate CATIA models: SI, containing only the problem surfaces and faces and OK, containing all the rest of the elements.

STEP 9: RUN THE EXTRCSI UTILITY IN CATIA

a) Key in /M EXTRCSI.

At this point, on-line instructions for extraction will be displayed. These instructions may be accessed at any time without interrupting processing by keying in /M HELPSI.

b) Select CURVE1 from function menu by pressing the middle mouse button in the menu zone.

c) Select the menu item, BOUNDARY. For each face or surface that is displayed, select the identifier for the face or surface with the mouse.

d) Select the menu item, ISOPARAM. For each face or surface that is displayed:

i) Select the identifier for the face or surface with the mouse.

ii) When prompted with:

KEY NUM/0 // SEL PT/LN

key-in the no. of isopars desired.

e) Select the following by pressing the middle mouse button in the menu zone:

LAYER
LAYER
CHANGE

The list of layers is presented in a scrollable window. The current layer is highlighted in red. Select the next higher layer in this list. Click on OFF to clear this list window from the screen.

If there are no elements visible on the screen, you are done. Continue with the next Step.

If there are faces or surfaces present in the new layer, continue with part b of this Step.

STEP 10: RUN THE CLEANSI UTILITY IN CATIA

Key in /M CLEANSI. This creates the SI CATIA model with all the problem surfaces/faces replaced by their supporting information elements.

STEP 11: EXIT FROM CATIA

STEP 12: RUN THE WRITESI UTILITY IN UNIX

In a terminal window, from the directory $HOME/convert/catia, type the command:

```
writesi file_name
```

STEP 13: RUN THE WRITEOK UTILITY IN UNIX

In a terminal window, from the directory $HOME/convert/catia, type the command:

```
writeok file_name
```

This will write an IGES file corresponding to the elements that had no problem converting from CATIA to IGES (CATIA model OK.)

Phase one of the conversion process is finished. Two IGES files have been produced. One file is named phase1/ok.igs and contains all the geometric elements that exhibit no problems during the native to IGES conversion process. The other file is named phase1/si.igs and contains intelligent supporting information for those geometric elements that are lost during the native to IGES conversion process. If the phase1/si.igs file is present, it means that there were problem elements found during phase 1. Otherwise, there were no problem elements during phase 1 and phase1/ok.igs is the same as file_name.igs.

PHASE TWO: IGES TO NATIVE PHASE

It is recommended that the user read the document titled "Assisted Conversion Phase II - Process Overview" before continuing with the conversion process.

The majority of the phase two activities are accomplished within the IGES/Works application. The IGES/Works application can be started on the Sun workstation or the dedicated conversion machine via an x-session. The choice of workstation is left to the user. The various IGES/Works utilities that have been developed for the PDC&SF project can be invoked from within the IGES/Works application.

STEP 1: START IGES/WORKS AND SELECT THE PART TO WORK ON

This step can be accomplished in two parts:

a) Starting an IGES/Works session for phase 2 conversion by typing:

start_iw phase2

from the x-server environment of the host machine.

b) Pick the menu item "Select Part" in the IGES/Works window and follow the prompts.

All IGES/Works utilities are selected from within the "User Functions" list. To use a utility, it is necessary to select the "User Functions" button first and then the button named for the desired utility.

The user will be asked to name the source CAD system and to pick from the available part names.

STEP 2: VIEW THE PHASE2/OK.IGS FILE

Select the "View" menu button and pick "OK Elements" from the menu.

STEP 3: SUMMARY OF THE PHASE2/OK.IGS FILE

Select the "Summary" menu button and pick "OK Elements" from the menu to obtain a summary of the contents of the phase2/ok.igs file. It is important to be aware of the contents of the file.

STEP 4: PARSE THE PHASE2/OK.IGS FILE

Select the "Parse" button from within IGES/Works and pick the target CAD system name from the menu when prompted. The Parse utility will list the potential problems in conversion and suggest actions to be taken. At the end of Parse, certain required actions such as elimination of independent views, blanking of construction geometry, etc., are automatically carried out on the file phase2/ok.igs..

If the phase2/ok.igs file is found to contain Engineering Drawing Entities (#404), the parse program will display a message asking the user to stop further assisted conversion.

STEP 5: CONVERT THE MATHEMATICAL REPRESENTATIONS IF NEEDED
(ONLY FOR CADKEY)

No math conversions are needed if the target system is CADDS, I/EMS, BRAVO3 or IDEAS. Math conversion is needed only for CADKEY.

Select the "To PS" menu button to carry out the necessary conversions on phase2/ok.igs and phase2/si.igs.

NOTE: At present, I/EMS utilities are used to carry out these mathematical form conversions.

These utilities will change the mathematical forms and produce two sets of files:

phase2/ok_old.igs (the original math form)

phase2/ok.igs (new math form)

phase2/si_old.igs (the original math form)

phase2/si.igs (new math form)

The original math forms are source specific while the new math forms are target specific.

STEP 6: ISOLATE THE PREDICTABLE ERRORS

Select the "Isolate Pred" button from within IGES/Works.

This utility will produce one new IGES file, phase2/pred.igs, containing all the elements subject to predictable loss. The remaining elements are saved to a file that is written over the phase2/ok.igs file.

STEP 7: CONVERT THE PHASE2/OK.IGS FILE TO TARGET CAD FORMAT

This step is required to detect random losses.

Select the "To Native" menu button from within IGES/Works. A menu is provided to select the IGES file to be converted to the CAD format. At this step select phase2/ok.igs as the input IGES file to be converted. Also select the target CAD system name when prompted by a menu.

The following files are created as a result:

(1) phase2/ok.xxx - target CAD model
(2) phase2/ok.log - log file

NOTE: At present, this conversion can be achieved from within the IGES/Works menu only when the target system is CADDS, I/EMS or IDEAS. In the case of CADKEY or BRAVO3, it is necessary to manually convert the phase2/ok.igs file at the dedicated CADKEY or BRAVO3 machines, respectively. Consult the document titled "Assisted Conversion Phase II - Process Overview" for more information on manual conversion.

STEP 8: VIEW THE LOG FILE IN IGES/WORKS

Select the "View Log File" menu button. Examine the phase2/ok.log file to see if any elements are dropped. Typically, the log file will show the Directory Entry Numbers (DE#) of those IGES elements that failed during the conversion. Note the Directory Entry Numbers of problem elements.

STEP 9: EDIT THE PHASE2/RAND.DAT FILE IN IGES/WORKS

Select the "Edit rand.dat" menu button. Enter the DE# of all the IGES elements that met with failure (random error).

STEP 10: COMPARE PHASE2/OK.IGS WITH THE TARGET CAD MODEL

Visual comparison is necessary to isolate any random errors not listed in the log file. This involves the following distinct steps:

Refer to the document titled "Assisted Conversion From target - Process Overview" for specific operating instructions for steps 10a and 10b.

(10a) Run START_target on the target CAD machine.
(10b) Run DISPLAY utility in the target CAD system. (Display the phase2/ok.xxx file)
(10c) Run the COMPARE utility from within IGES/Works. Select the "Compare" menu button.

This utility does the following:

a) Reads and Displays the phase2/ok.igs file.
b) Highlights each element in the file and allows the user to select it for isolation, if necessary.
c) Appends DE#'s of elements identified for isolation to the phase2/rand.dat file automatically.

At the completion of this step, the phase2/rand.dat file will contain the DE#'s of all the elements that are subject to random loss during the IGES to native conversion.

STEP 11: RUN ISOLATE_RAND ON PHASE2/OK.IGS FILE

Select the "Isolate Rand" menu button from within IGES/Works.

All elements subject to random error are now saved to the file named phase2/rand.igs.

At the end of this step, all problem elements (predictable and random) have been removed from the phase2/ok.igs file.

STEP 12: COMBINE PHASE2/PRED.IGS & PHASE2/RAND.IGS INTO PHASE2/PROB.IGS

Select the menu button "Combine" from within IGES/Works. At this step pick from the menu provided the item:

"phase2/pred.igs + phase2/rand.igs"

This will create the IGES file, phase2/prob.igs containing all the elements that are subject to error in the second phase.

STEP 13: CONVERT THE PHASE2/PROB.IGS TO SOURCE CAD (CATIA) FORMAT

Select the menu button "To Native" and pick from the menu provided, the item:

"phase2/prob.igs"

Also select the name of the CAD system (CATIA) to convert to when prompted with a menu.

This will convert phase2/prob.igs in IGES format to si2 in CATIA format.

If no errors (pred and rand) have been produced, the user can skip to step 18.

STEP 14: RUN THE START_CATIA ON THE DEDICATED SOURCE CAD (CATIA) MACHINE

Verify that the appropriate window environment has been initiated.

First change the working directory to $HOME/convert/catia. The user invokes the UNIX procedure, START_CATIA that starts the CATIA application.

a) This command is invoked as

start_catia file_name phase2

This action starts up CATIA.

b) Select COLD START.

c) Press the KEY permanent function.

d) Select the keyboard named PDCSF to access the customized function keyboard for assisted conversion.

e) Select FILE.

f) Key in /M DISPLAY.

The CATIA model is displayed in the isometric view and autoscaled in the CATIA graphics window.

STEP 15: RUN THE EXTRCSI UTILITY IN CATIA

a) Key in /M EXTRCSI.

At this point, on-line instructions will be displayed. These instructions may be accessed at any time without interrupting processing.

b) Select CURVE1 from function menu by pressing the middle mouse button in the menu zone.

c) Select the menu item, BOUNDARY. For each face or surface that is displayed, select the identifier for the face or surface with the mouse.

d) Select the menu item, ISOPARAM. For each face or surface that is displayed:

i) Select the identifier for the face or surface with the mouse.

ii) When prompted with:

KEY NUM/0 // SEL PT/LN

key-in the no. of isopars desired.

e) Select the following by pressing the middle mouse button in the menu zone:

LAYER
  LAYER
    CHANGE

The list of layers is presented in a scrollable window. The current layer is highlighted in red. Select the next higher layer in this list. Click on OFF to clear this list window from the screen.

If there are no elements visible on the screen, you are done. Continue with the next Step.

If there are faces or surfaces present in the new layer, continue with part b of this Step.

STEP 16: RUN THE CLEANSI UTILITY IN CATIA

Key in /M CLEANSI. This creates the SI CATIA model with all the problem surfaces/faces replaced by their supporting information elements.

STEP 17: EXIT FROM CATIA

STEP 18: RUN THE WRITESI2 UTILITY IN UNIX

In a terminal window, from the directory $HOME/convert/catia, type the command:

writesi2 file_name

STEP 19: CONVERT THE MATHEMATICAL REPRESENTATIONS IF NEEDED
(ONLY FOR CADKEY)

No math conversions are needed if the target system is CADDS, I/EMS, BRAVO3 or IDEAS. Math conversion is needed only for CADKEY.

Select the "To PS" menu button to carry out the necessary conversion phase2/si2.igs.

NOTE: At present, I/EMS utilities are used to carry out these mathematical form conversions.

These utilities will change the mathematical forms and produce one set of files:

phase2/si2_old.igs (the original math form)

phase2/si2.igs (new math form)

The original math forms are source specific while the new math forms are target specific.

STEP 20: COMBINE PHASE2/SI.IGS & PHASE2/SI2.IGS INTO PHASE2/TOTAL_SI.IGS

Select the menu button "Combine" from within IGES/Works and pick from the menu provided the item:

"phase2/si.igs + phase2/si2.igs"

This will create phase2/total_si.igs containing supporting information for all elements that are subject to error in the entire conversion process (phase 1 and phase 2).

STEP 21: CONVERT THE PHASE2/OK.IGS TO TARGET CAD FORMAT

Select the menu button "To Native" from within IGES/Works. Pick "phase2/ok.igs" from the menu as the name of the IGES file to convert to target CAD format.

Also select the name of the target CAD system to convert to when prompted with a menu.

This produces two files:
1) phase2/ok.xxx - the target CAD model.
2) phase2/ok.log - the log file.

STEP 22: CONVERT THE PHASE2/TOTAL_SI.IGS TO TARGET CAD FORMAT

Select the menu button "To Native" from within IGES/Works. Pick "phase2/total_si.igs" from the menu as the name of the IGES file to convert to target CAD format.

Also select the name of the target CAD system to convert to when prompted with a menu.

This produces two files:
1) phase2/total_si.xxx - the target CAD model.
2) phase2/total_si.log - the log file.

STEP 23: RE-CREATE THE PROBLEM ELEMENTS ON THE DEDICATED TARGET CAD SYSTEM
(manual)

The re-creation is entirely dependent on the functionality of the target CAD system.

NOTE 1: It may be necessary to re-limit (trim / extend) the surfaces after re-creation.

NOTE 2: It may be necessary to use the alternate supporting information to re-create non-simply connected surfaces. This option is not available in the current release of the PDC&SF utilities.

NOTE 3: Some CAD systems may not allow for any re-creation at all.

Consult system specific investigation results for available recreation options.

STEP 24: COMBINE THE TARGET CAD MODELS OK & TOTAL_SI INTO ONE MODEL

Use the functionality of the target CAD system to do this.

BRAVO:	Use ADD MODEL command.
CADDS:	Insert one part into the other part.
CADKEY:	Save as and then later Merge as a pattern file.
I-DEAS:	Load Objects To PEARL & Unload To Model File.
I/EMS:	Reference one file within the other and copy elements.

STEP 25: ALTERNATE STEP: COMBINE USING IGES/WORKS

If combining two models into a single model is not possible within the target CAD system, this step is suggested.

Select the menu button "Combine" from within IGES/Works and pick from the menu provided, the item:

"phase2/ok.igs + phase2/total_si.igs"

This will create the file phase2/target.igs.

STEP 26: ALTERNATE STEP: CONVERT PHASE2/TARGET.IGS TO TARGET CAD SYSTEM NATIVE FILE

This step is taken only if phase2/target.igs is created in the previous step.

Select the menu button "To Native" and pick "phase2/target.igs" as the name of the input IGES file when prompted with a menu.

The converted file can be used to do the re-creation of entities coming from phase2/total_si.igs in the target CAD system.

STEP 27: MOVE COMPLETED FILE TO STORAGE LOCATION OR ENDUSER

When the conversion expert is certain that assisted conversion on the current model has been completed, it is necessary to move the completed model to a more permanent storage location. This can be done in many ways, for example:

a) ftp model to another account
b) put model on tape or disk
c) put model into repository

Be certain that all desired information is stored before executing the next step.

NOTE: Failure to store information will result in loss of all work done in the assisted conversion process.

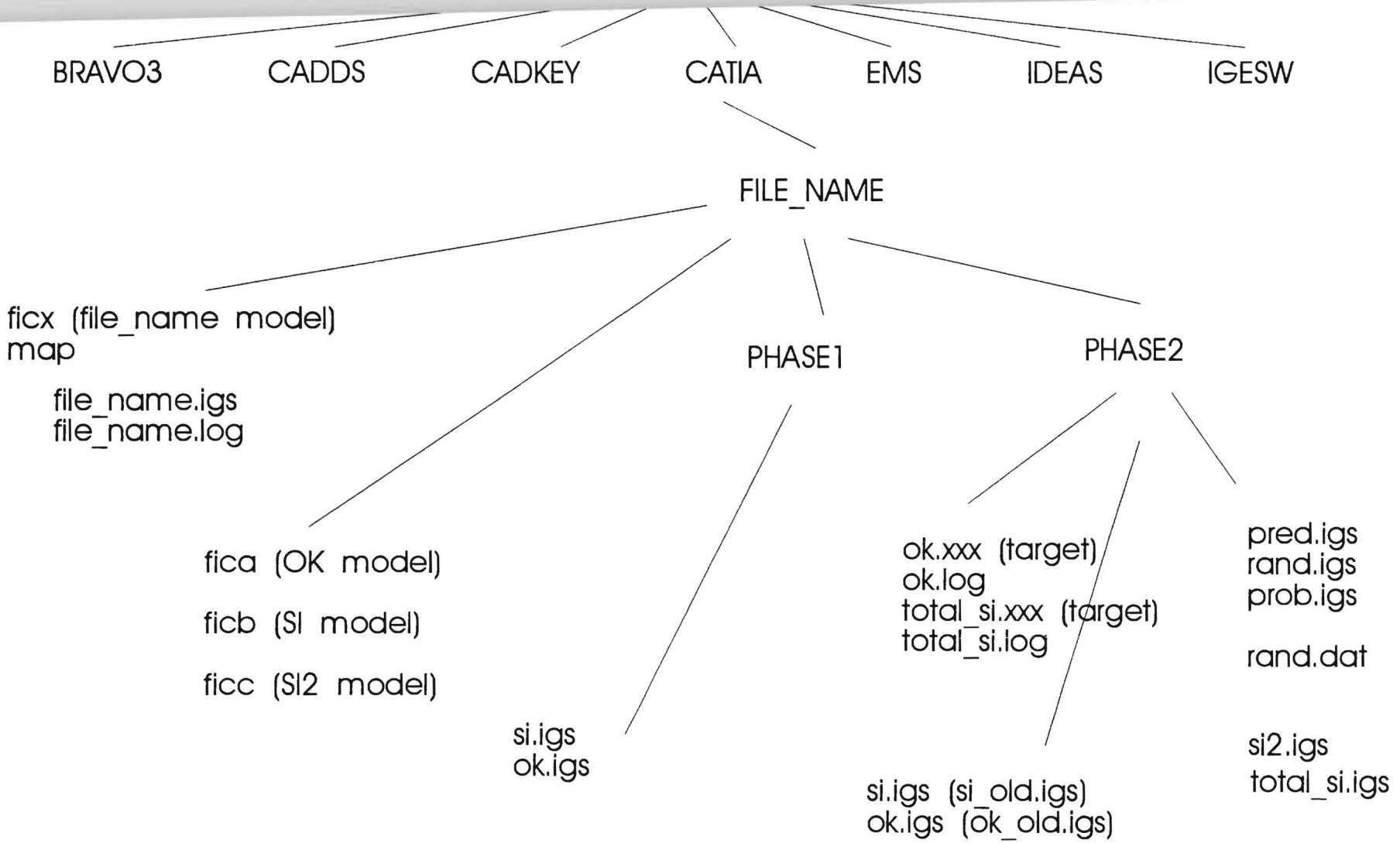
STEP 28: RUN THE CLEANUP UTILITY IN UNIX

Once the user is certain that all desired information has been properly stored, the working directory (file_name) should be removed from the transparent file system.

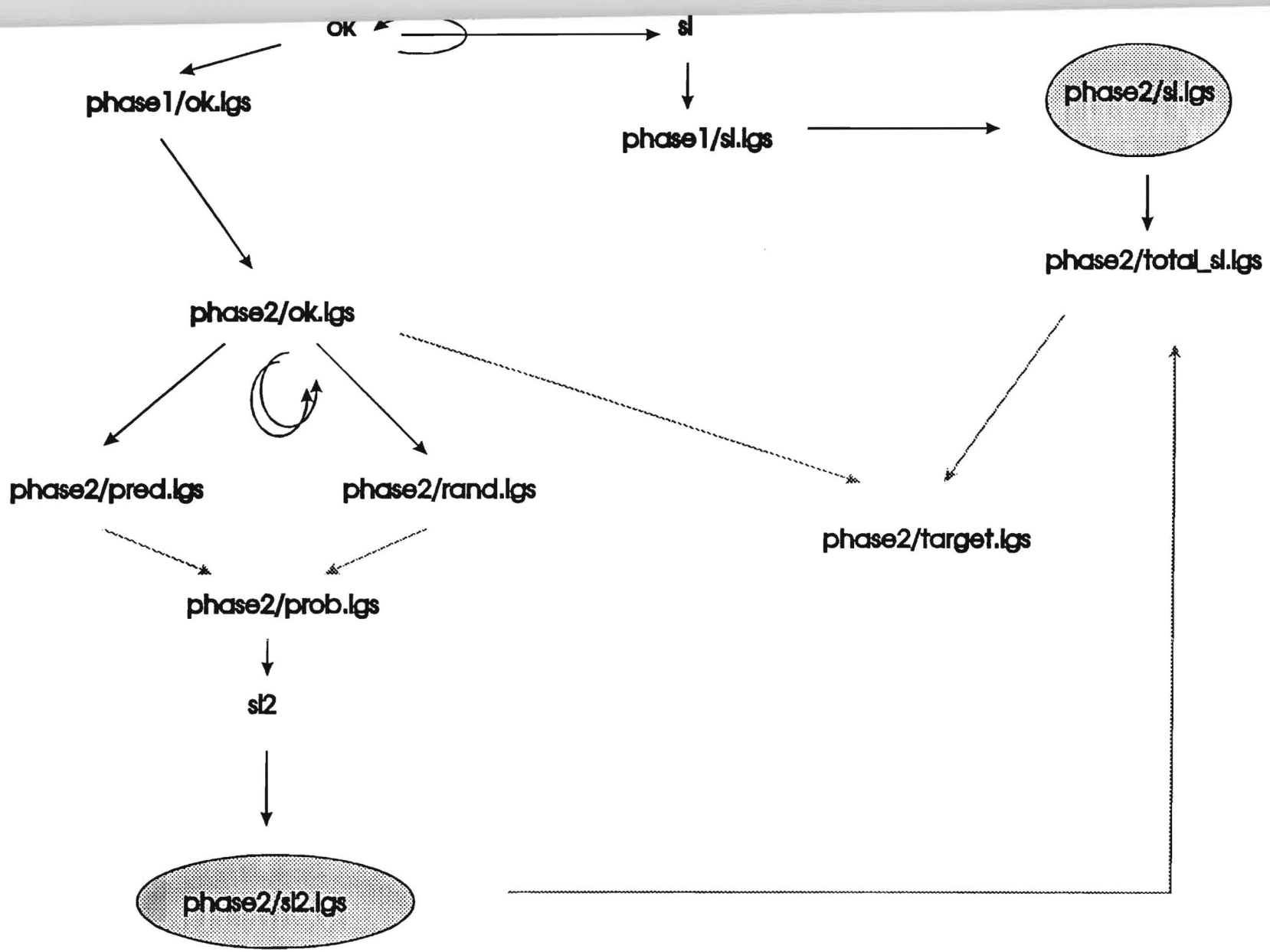
Change the working directory to $HOME/convert/catia.

Run the cleanup utility in UNIX. The name of the part is the only argument to this command:

cleanup file_name



FILE NAMING CONVENTIONS
CONVERSION FROM CATIA


ok
sl
phase1/ok.lgs
phase1/sl.lgs
phase2/sl.lgs
phase2/total_sl.lgs
phase2/ok.lgs
phase2/pred.lgs
phase2/rand.lgs
phase2/target.lgs
phase2/prob.lgs
sl2
phase2/sl2.lgs

CATIA Version 3 Utilities - Quick Reference

Phase I Utilities

ORG_FILE
Sets up the PDCSF directory structure.
org_file filename

WRITEIGES
Writes out an IGES file from the CATIA model model_name.
writeiges model_name

START_CATIA
Starts the CATIA system for Assisted conversion.
start_catia filename phasenum

DISPLAY
Displays CATIA model OK for phase I and SI2.
Key in /M DISPLAY from inside CATIA.

SUMMARY
Displays the summary of the contents of the CATIA model.
summary model_name

COMPARE
Highlights each surface to allow user to select those lost.
Key in /M COMPARE from inside CATIA.

ISOLATE
Removes problem surfaces from the OK model.
Key in /M ISOLATE from inside CATIA.

EXTRCSI
Extracts supporting information for problem surfaces and faces.
Key in /M EXTRCSI from inside CATIA.

CLEANSI
Removes all the problem surfaces from SI or SI2 model.
Key in /M CLEANSI from inside CATIA.

WRITESI
Writes an IGES file from the SI CATIA model.
writesi model_name

WRITEOK
Writes an IGES file from the OK CATIA model.
writeok model_name

Phase II Utilities

START_CATIA
Starts the CATIA system for Assisted conversion.
start_catia filename phasenum

DISPLAY
Displays CATIA model OK for phase I and SI2.
Key in /M DISPLAY from inside CATIA.

EXTRCSI
Extracts supporting information for problem surfaces and faces.
Key in /M EXTRCSI from inside CATIA.

CLEANSI
Removes all the problem surfaces from SI or SI2 model.
Key in /M CLEANSI from inside CATIA.

WRITESI2
Writes an IGES file from the SI2 CATIA model.
writesi2 model_name

CLEANUP
Deletes all files that are created during the conversion process.
cleanup dirname

Other Utilities

CTRDIGAUTO
Converts a file in IGES format to a CATIA model file.
ctrdigauto igesfilename native_filename

CTWRIGAUTO
Converts CATIA models to files in the IGES format.
ctwrigauto CATIA_modelfile igesfilename

HELPSI
Presents instructions for extracting supporting information.
Key in /M HELPSI from inside CATIA.

POSCATDIR
Post-processes CATIA models to the form required for repository.
poscatdir target_dir

PRECATDIR
Pre-proc CATIA models in repository format for direct conversion.
precatdir source_dir

RST_CATIA
Cleans up transparent file system after an error has occurred.
rst_catia file_name

Utilities Developed For CATIA Programming Language

The CATIA system provides programming tools to perform macro functions inside the CAD system. This includes FORTRAN callable subroutines (CATGEO and CATMSP), Interactive User Access (IUA) language and Graphic Interactive Interface (GII) functions. PDCSF project has used these tools extensively.

The following lists the various utilities developed by the PDCSF project for CATIA.

CLEANSI

Path: $HOME/CATIA/iuaproc/CLEANSI on ids6000
Language: IUA
Size: 3152 bytes
Function: Removes all the problem surfaces and faces from the SI or SI2 model leaving behind only supporting information for them.

Files Needed: CLEANSI,clnsi.f,CLNSIERR

Files Produced: none

Transparent File System: Necessary

Systems Integration:

Documentation: Extensive in-code documentation.

Usage Instructions: Key in /M CLEANSI from inside CATIA.

CLEANUP

Path: $HOME/convert/catia/cleanup on ids6000
Language: UNIX
Size: 572 bytes
Function: Deletes all files that are created during the conversion process.

Files Needed: cleanup

Files Produced: modelfile.igs, modelfile.log

Transparent File System: Necessary

Systems Integration:

Documentation: Extensive in-code documentation.

Usage Instructions: cleanup dirname

COMPARE

Path: $HOME/CATIA/iuaproc/COMPARE on ids6000
Language: IUA
Size: 5452 bytes
Function: To highlight each surface or face in the CATIA model to allow the user to select the ones that are lost in the CATIA to IGES translation during Phase one conversion on CATIA.

Files Needed: SELPROB,wrsi.f

Files Produced: SI CATIA model

Transparent File System: Necessary

Systems Integration:

Documentation: Extensive in-code documentation.

Usage Instructions: Key in /M COMPARE from inside CATIA.

CTRDIGAUTO

Path: $HOME/convert/catia/ctrdigauto on ids6000
Language: UNIX
Size: 2316 bytes
Function: Converts a file in IGES format to a CATIA model file.

Files Needed: igesfilename.igs.

Files Produced: ficn file corresponding to CATIA model igesfilename.

Transparent File System: Necessary

Systems Integration:

Documentation: Extensive in-code documentation.

Usage Instructions: ctrdigauto igesfilename native_filename
(Full pathnames with no extensions are needed for the file names.)

CTWRIGAUTO

Path: $HOME/convert/catia/ctwrigauto on ids6000
Language: UNIX
Size: 3610 bytes
Function: Converts all the CATIA models present in the CATIA model file to files in the IGES format.

Files Needed: xtrc_catlist, CATIA model file containing the CATIA models.

Files Produced:model.igs, model.log for each CATIA model.

Transparent File System: Necessary

Systems Integration:

Documentation: Extensive in-code documentation.

Usage Instructions: ctwrigauto CATIA_modelfile igesfilename
(Full pathnames with no extensions are needed for the file names. CATIA_modelfile is the full pathname to the directory which is the CATIA model file holding the CATIA models.)

DISPLAY

Path: $HOME/CATIA/iuaproc/DISPLAY on ids6000
Language: IUA
Size: 2824 bytes
Function: Displays CATIA model OK for phase I and SI2 for phase II Assisted Conversion.

Files Needed: DISPLAY, rdmodl.f and OK or SI2 CATIA model.

Files Produced:

Transparent File System: Necessary

Systems Integration:

Documentation: Extensive in-code documentation.

Usage Instructions: Key in /M DISPLAY from inside CATIA.

EXTRCSI

Path: $HOME/CATIA/iuaproc/EXTRCSI on ids6000
Language: IUA
Size: 3332 bytes

Function: Extracts supporting information for problem surfaces and faces.

Files Needed: EXTRCSI, prepsi.f, HELPSI, SI or SI2 CATIA model.

Files Produced:

Transparent File System: Necessary

Systems Integration:

Documentation: Extensive in-code documentation.

Usage Instructions: Key in /M EXTRCSI from inside CATIA.

HELPSI

Path: $HOME/CATIA/iuaproc/HELPSI on ids6000
Language: IUA
Size: 2752 bytes
Function: Presents the instructions for extracting isoparametric curves and bounding curves for problem surfaces and faces.

Files Needed: HELPSI (IUA procedure), HELPSI(IUA panel)

Files Produced: none

Transparent File System: Necessary

Systems Integration:

Documentation: Extensive in-code documentation.

Usage Instructions: Key in /M HELPSI from inside CATIA.

ISOLATE

Path: $HOME/CATIA/iuaproc/ISOLATE on ids6000
Language: IUA
Size: 3261 bytes
Function: Removes all the problem surfaces and faces from the OK model.

Files Needed: ISOLATE,clnok.f,clrsil.f

Files Produced:

Transparent File System: Necessary

Systems Integration:

Documentation: Extensive in-code documentation.

Usage Instructions: Key in /M ISOLATE from inside CATIA.

ORG_FILE

Path: $HOME/convert/catia/org_file on id6000
Language: UNIX
Size: 1383 bytes
Function: Sets up the PDCSF directory structure.

Files Needed: filename

Files Produced: phase1, phase2 (directories)

Transparent File System: Necessary

Systems Integration:

Documentation: Extensive in-code documentation.

Usage Instructions: org_file filename

POSCATDIR

Path: $HOME/convert/catia/poscatdir on ids6000
Language: UNIX
Size: 3523 bytes
Function: Post-processes CATIA models produced during direct conversion into the form required for storage in the PDCSF repository.

Files Needed: batchread/native (directory), appen_sysdyn, remov_sysdyn

Files Produced: target_dir (directory)

Transparent File System: Necessary

Systems Integration:

Documentation: Extensive in-code documentation.

Usage Instructions: poscatdir target_dir
(target_dir is the name of the directory under $HOME/convert/catia that holds the resulting CATIA models in the PDCSF repository format.)

PRECATDIR

Path: $HOME/convert/catia/precatdir on ids6000
Language: UNIX
Size: 3068 bytes
Function: Pre-processes CATIA models in the PDCSF repository format for use in direct conversion.

Files Needed: source_dir (directory), appen_sysdyn, remov_sysdyn

Files Produced: CATIA models in the model file, batchwrite (directory)

Transparent File System: Necessary

Systems Integration:

Documentation: Extensive in-code documentation.

Usage Instructions: precatdir source_dir
(source_dir is the name of the directory under $HOME/convert/catia that holds the original CATIA models in the PDCSF repository format.)

RST_CATIA

Path: $HOME/convert/catia/rst_catia on ids6000
Language: UNIX
Size: 3068 bytes
Function: Cleans up the transparent file system after an error has occurred in assisted conversion so the user can begin the conversion process again.

Files Needed: none

Files Produced: none

Transparent File System: Necessary

Systems Integration:

Documentation: Extensive in-code documentation.

Usage Instructions: rst_catia file_name

START_CATIA

Path: $HOME/convert/catia/start_catia on ids500
Language: UNIX
Size: 4434 bytes
Function: Starts the CATIA system for Phase I or Phase II Assisted conversion.

Files Needed: none

Files Produced: none

Transparent File System: Necessary

Systems Integration:

Documentation: Extensive in-code documentation.

Usage Instructions: start_catia filename phasenum

SUMMARY

Path: $HOME/convert/catia/summary on ids6000
Language: UNIX
Size: 1185 bytes
Function: Displays the summary of the contents of the CATIA model during Phase I Assisted Conversion.

Files Needed: summary, model_name (CATIA model file)

Files Produced: none

Transparent File System: Necessary

Systems Integration:

Documentation: Extensive in-code documentation.

Usage Instructions: summary model_name

WRITEIGES

Path: $HOME/convert/catia/writeiges on ids6000
Language: UNIX
Size: 4436 bytes
Function: Writes out an IGES file from the CATIA model during Phase I Assisted Conversion.

Files Needed: writeiges, model_name (CATIA model file)

Files Produced: model_name.igs, model_name.log

Transparent File System: Necessary

Systems Integration:

Documentation: Extensive in-code documentation.

Usage Instructions: writeiges model_name

WRITEOK

Path: $HOME/convert/catia/writeok on ids6000
Language: UNIX
Size: 3832 bytes
Function: Writes out an IGES file from the OK CATIA model during Phase I Assisted Conversion.

Files Needed: writeok, model_name (CATIA model file)

Files Produced: phase1/ok.igs, phase1/ok.log

Transparent File System: Necessary

Systems Integration:

Documentation: Extensive in-code documentation.

Usage Instructions: writeok model_name

WRITESI

Path: $HOME/convert/catia/writesi on ids6000
Language: UNIX
Size: 4178 bytes
Function: Writes out an IGES file from the SI CATIA model during Phase I Assisted Conversion.

Files Needed: writesi, model_name (CATIA model file)

Files Produced: phase1/si.igs, phase1/SI.log

Transparent File System: Necessary

Systems Integration:

Documentation: Extensive in-code documentation.

Usage Instructions: writesi model_name

WRITESI2

Path: $HOME/convert/catia/writesi2 on ids6000
Language: UNIX
Size: 4184 bytes
Function: Writes out an IGES file from the SI2 CATIA model during Phase II Assisted Conversion.

Files Needed: writesi, model_name (CATIA model file)

Files Produced: phase2/si2.igs, phase1/SI2.log

Transparent File System: Necessary

Systems Integration:

Documentation: Extensive in-code documentation.

Usage Instructions: writesi2 model_name

Phase II
Process Overview

Assisted conversion from a source CAD system to a target CAD system consists of two phases - phase one and phase two. The overview of the process of assisted conversion from each of the source CAD systems considered in this project can be found in the documents titled "Assisted Conversion from [source] - Process Overview" (where source is one of the following: BRAVO3, CADDS, CADKEY, CATIA, I/EMS or I-DEAS). The present document provides an overview of phase two of the assisted conversion process common to all the CAD systems.

Phase two activities are carried out mainly within the IGES/Works environment. These are started once the phase one activities are completed as documented in detail in each of the documents titled "Assisted Conversion from [source]". At the start of phase two, the directory structure for the assisted conversion process as well as the files produced at the end phase one should already have been created.

The IGES/Works environment for phase two conversion is started up by typing "start_iw phase2" in a shell window in the X-window environment of the PDCSF account. The user should be aware of the source CAD system and the name of the part being converted, since these need to be identified to the phase two conversion process. The user will now be provided with an IGES/Works environment with customized menus for phase two.

The first action that the user needs to take in phase two conversion is to pick the "Select Part" menu button. This action is essential before any other action in the phase two conversion process is started. If a mistake is made in selecting part, the user may press the button "Select Part" to make another selection. Also, if the user wishes to abort the phase two conversion process in the middle for any reason, the user could select the button "Restart". This will cause all work done from the beginning of phase two conversion to be lost and the process will be left in a state akin to the one just after "Select Part" was chosen. Files created during phase one are, however, left unaffected. However, if an error of some sort causes the IGES/Works environment to abort and the user is returned to the X-windows environment, begin the phase two processing from the "start_iw phase2" step. Execute the "Select Part" command. When IGES\Works asks if you would like to start this part afresh, type "yes". This will erase all the intermediate files created during the previous phase two conversion attempt for the same part and allow the user to start from step one of phase two again.

It is to be noted that all the display manipulation functions (eg., zoom, rotate, etc.) are fully available during the phase two conversion process in the IGES/Works environment.

Phase II Utilities - Quick Reference

START_IW
Starts up IGES/Works.
start_iw [phase2]

SELECT PART
Select the part to work on.
Select the "Select Part" menu button.

VIEW
The graphics is displayed in the graphics window.
Select the "View" menu button.

SUMMARY
Summary is displayed in the IGES/Works text window.
Select the "Summary" menu button.

PARSE
Parses the ok.igs file during the phase two conversion process.
Select the "Parse" menu button.

TO_PS
Converts the mathematical form to parametric splines.
Select the menu button "To PS".

ISOL_PRED
Separates the elements causing predictable errors from ok.igs.
Select the menu button "Isolate Pred".

TO_NATIVE
Reads the input IGES file and converts to the native CAD format.
Select the menu button "To Native".

VIEW_LOG
Displays the ok.log file.
Select the "View Log File" menu button.

EDIT_RAND
Create the rand.dat file.
Select the menu button "Edit rand.dat".

COMPARE
Displays the file ok.igs and highlights one surface at a time.
Select the menu button "Compare".

ISOL_RAND
Separates the elements causing random errors from ok.igs.
Select the menu button "Isolate Rand".

COMBINE
Merges two files into the appropriate resultant file.
Select the menu button "Combine".

Other Utilities

RESTART
Cleans up all the files created during the phase two conversion.
Select the "Restart" menu button.

SHOW_ALL
Displays all layers.
Select the "Show All" menu button.

SHOW_LAY
Displays only selected layer range.
Select the "Show Lay" menu button.