## EDIF 200 to OrCAD Capture Schematic Translator EDIF2CAP

#### **User's and Reference Manual**





Copyright © 2003 Cadence Design Systems, Inc.

All rights reserved. OrCAD and Cadence and the OrCAD and Cadence logos are registered trademarks.

All other are properties of their respective holders.

Copyright © 1999-2003 Electronic Tools Company.

Copyright © 2006-2011 Elgris Technologies, Inc.

All rights reserved. Confidential. May be photocopied by licensed customers of Cadence for internal business purposes only



This document may not be stored in a retrieval system, reproduced or transmitted in any form or by any means, either in whole or in part, without express prior written permission of Electronic Tools Company or Elgris Technologies, Inc. Copying includes translating into another language or format. Notice to Government users: Use, duplication or disclosure by the Government is subject to the restrictions as set forth in subparagraphs (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013. Unpublished--all rights reserved under the copyright laws of the United States.

This document contains proprietary information of Electronic Tools Company or Elgris Technologies, Inc. Electronic Tools Company has prepared this document for use only by employees of Electronic Tools Company or Elgris Technologies, Inc. and use by customers of Electronic Tools Company or Elgris Technologies, Inc. under the current license and maintenance agreements of Electronic Tools Company or Elgris Technologies, Inc. The only undertakings of Electronic Tools Company or Elgris Technologies, Inc. respecting the information in this document are as contained in such contracts, and nothing contained in this document shall be construed as changing said contracts. Information in this document is subject to change without notice and does not represent a commitment on the part of Electronic Tools Company or Elgris Technologies, Inc. Any use of this information except as defined by a written contract with Electronic Tools Company or Elgris Technologies Inc., is not authorized and, with respect to any such unauthorized use, neither Electronic Tools Company, Elgris Technologies Inc, nor any of the contributors to this document makes any representation or warranty, nor shall any warranty be implied, as to the completeness, accuracy, or usefulness of the information contained in this document or that such use of information may not infringe privately owned rights, nor do they assume any responsibility for liability or damage of any kind which may result from such use of such information.

ELECTRONIC TOOLS COMPANY, 928 First Street West, Sonoma, California 95476, USA Phone: 707-996-3320; Fax: 707-939-0246; e-mail: sales@e-tools.com

ELGRIS TECHNOLOGIES, INC. 465 Stony Point Road, 236, Santa Rosa, California 95401, USA Phone: 707-237-2794; Fax: 707-573-0237; e-mail: sales@elgris.com



#### **Overview**

This section provides a general description of the EDIF 200 schematic applications and presents the basic concepts you must understand to use them.

EDIF, which stands for Electronic Design Interchange Format, is an industry standard to facilitate formatting and exchanging electronic design data between EDA (Electronic Design Automation) systems. It is designed to account for all types of electronic design information, including schematic design, symbolic and physical layout, connectivity, and textual information, such as properties.

EDIF was originally proposed as an industry standard by Mentor Graphics, Motorola, National Semiconductor, Texas Instruments, Daisy Systems, Tektronix, and the University of California at Berkeley, all of which collaborating embarked on its development. Since that time, EDIF has been accepted by more and more companies. EDIF V 2 0 0 was approved as a standard by the Electronic Industries Association (EIA) in 1987, and by the American National Standards Institute (ANSI) in 1988.

Figure 1 illustrates the process of creating a OrCAD Capture design from an EDIF 200 file.

From other EDA system

EDIF 200 File

Configuration File

Configuration File

OrCAD Capture Design

Figure 1: Creating a Design from an EDIF 200 File

Overview 1



The following subsections describe both the process and elements involved in translating an EDIF 200 file into a OrCAD Capture design using EDIF2CAP.

EDIF 200 File. The input to EDIF2CAP is an EDIF 200 file created within some EDA environment. This file must conform to the constructs defined in the *Electronic Design Interchange Format Version* documents, available from the Electronics Industries Association (EIA).

Although the input EDIF 200 file may contain both schematic and netlist and connectivity views, EDIF2CAP only supports schematic views (that is, netlist and other views found in the file are ignored).

**EDIF2CAP** and the Configuration File. The configuration file is an ASCII file containing commands that you can use to control the operation of EDIF2CAP. Although it is an optional item, the configuration file is usually involved in a design or library translation. You create the configuration file as you would any other text file.

**EDIF2CAP.** It performs three main tasks. First, it reads the configuration for your design translation. Next, it reads the EDIF 200 file. During the read, both the syntax and semantics of the EDIF 200 file are checked to ensure the file conforms to EDIF 200 standards and can be used to write a design. Finally, it writes out an OrCAD Capture representation of the EDIF 200 file.

**Syntax Checks.** Before the file is translated into an OrCAD Capture representation of the design, EDIF2CAP performs a series of syntax checks on the input EDIF 200 file. During these checks, EDIF2CAP detects and reports any incorrect constructs or incorrect grammar, such as unmatched parentheses or the referencing of objects that have not been previously defined. (Forward-referencing is not legal in EDIF 200).

**Semantic Checks**. EDIF2CAP also performs a series of semantic checks on the input EDIF file. For example, an EDIF file may attempt to reference a component which is undefined. During the semantic checks, EDIF2CAP also detects and reports any application-specific actions that do not fall into the realm of EDIF2CAP. For example, an EDIF 200 file may contain views other than schematic views.

**The OrCAD Capture Design.** After reading and processing the EDIF 200 file, EDIF2CAP creates a OrCAD Capture representation of the design that can be used by OrCAD Capture 10.0 and above.

2 Overview



## Requirements

Windows systems running OrCAD Capture 10.0 and above.

### **Major Features of the EDIF2CAP Schematic Translator**

- EDIF2CAP supports EDIF Version 2 0 0.
- EDIF2CAP can translate all design including symbols and schematic pages;
- EDIF2CAP can translate only symbols when symbol library is translated
- EDIF2CAP translates all levels of Hierarchy of the design
- EDIF2CAP translates all design data including connectivity and properties that reside on the objects
- EDIF2CAP translates occurrence data if special configuration command is provided
- EDIF2CAP translates packaging information, including translation of Heterogeneous packages for annotated designs
- EDIF2CAP is configured with the help of the configuration file
- EDIF2CAP issues warnings and error messages, helping you to debug your translation.

### **Running EDIF to OrCAD Capture**

Below are instructions on how to translate an EDIF 2 0 0 file into OrCAD Capture library or OrCAD Capture Schematic:

- 1. Click File -> Import Design
- 2. Click on EDIF tab.
- 3. In the "Open" field, type the name of the EDIF 2 0 0 file you wish to translate to OrCAD Capture
- 4. In the "Save As" field the name of the OrCAD Capture file to which EDIF 2 0 0 file will be translated will appear. Please, modify this name either by entering it or by using "Browse" button.
- 5. In the "Configuration file" field, type path to the EDIF2CAP configuration file, or use the "Browse" button to locate it.

Requirements 3



- 6. Press on the OK button.
- 7. The LOG file with the name "edi2cap.log" will be created in the directory where OrCAD Capture design is created. The Session log will contain the contents of the "edi2cap.log" file.
- 8. If you were translating a design, please look at the translated OrCAD Capture design opened in OrCAD Capture.
- 9. If you were translating a library, open the translated library in the OrCAD Capture by using File/Open in OrCAD Capture.

### Standard configuration files

For translation from "generic" Vendor and for the roundtrip translations "edi2cap.cfg" configuration file is provided.

For translation from the other Vendors "view2cap.cfg" and "ment2cap.cfg" configuration files are provided.

## **Key Concepts**

### Translation of Designs and Libraries

EDIF2CAP will try to determine the top cell of the design you are translating. The contents of the view of this cell will become Root Schematic after translation to OrCAD Capture.

While translating the design EDIF2CAP will create OrCAD Capture libraries with all Packages, Library Parts, Net Symbols that were created during the translation.

EDIF2CAP is trying to keep the Customer's data and does not override existing libraries that have the same name as the translated library. When it tries to create a library and finds that the library with the same name exists, EDIF2CAP creates a new library with an indexed name.

If you don't want EDIF2CAP to create libraries with the indexed names select a directory for the translation where no libraries exist before the translation.

When only libraries should be translated from your EDIF 2 0 0 file, please set ConvertOnlyLibs = 1 in the [OrCAD Writer] section of the configuration file.



You could set **ComponentsMode** = **1** in the [OrCAD Writer] section of the configuration file to make translator create a separate library for every created Package.

### Pin spacing

EDIF2CAP is calculating pin spacing basing on your design data automatically, but you can change it manually if you are not satisfied with the result or would like to use a different pin spacing in your design.

The two parameters that affect scaling are **PinToPin** and **Grid**.

If **PinToPin=0** in the [OrCAD Writer] section of the configuration file then EDIF2CAP is calculating minimal pin spacing automatically.

EDIF2CAP looks at all pin locations of all instances that are present in the design (or pin locations of all symbols in the library) and then computes the minimal pin to pin spacing that will make all pins appear on grid in OrCAD Capture.

EDIF2CAP issues an "Advice" that you can find in the log about the result of its computation. You can follow this advice in the following run of the translator.

When it is ran in automatic mode (**PinToPin=0**) EDIF2CAP tries to make all objects and nets appear on grid in OrCAD Capture. If it is impossible EDIF2CAP places some objects off-grid.

When **Pin2Pin** is specified by the Customer it is necessary to set additional parameter **UnitMeasurement**. It will tell EDIF2CAP in what units the **Pin2Pin** is set. It is also recommended to set **EnableOffgridObjects=1** in this case, so the translator will be able to put some objects off-grid to produce better connections. If **EnableOffgridObjects=0** EDIF2CAP tries to place everything on grid, modifying pins positions on the Symbol if they were off-grid.

The **Grid** parameter determines how many EDIF 2 0 0 Units will result in one OrCAD Capture grid. By default **Grid** = **0** which means that EDIF2CAP will determine automatically how many EDIF 2 0 0 units will be in one OrCAD Capture grid.

It is not recommended to set Grid to any other value than 0 until you want to scale your design.

OrCAD Capture has standard pin spacing equal to 0.1 inch, so it is recommended to set the following parameters in the configuration file converting from a system that has a different pin spacing:

Key Concepts 5



UnitMeasurement=inch
Pin2Pin = 0.1
EnableOffgridObjects = 1
Grid = 0

Generation of additional Hierarchical Ports, Off-Page connectors, Aliases on Nets and Busses

Some EDIF 2 0 0 files don't contain Hirarchical Ports to establish connectivity between 2 levels of hierarchy or Off-page connectors to establish connectivity between the pages of one schematic view.

The commands **GeneratePortSymbols** = 1 and **GenerateOffPage-Connectors** = 1 from the [OrCAD Writer] section of the configuration file make EDIF2CAP generate necessary Hierarchical Ports or Off-Page Connectors to keep the connectivity from the EDIF 2 0 0 file.

The command **GenerationVisible = 1** makes EDIF2CAP to create all generated objects visible.

EDIF2CAP places Aliases on the Nets and Busses that don't have visible display in the EDIF 2 0 0 file. Those Aliases are visible if **GenerationVisible = 1** or they are placed in the background color if **GenerationVisible = 0** 

When EDIF2CAP generates Aliases for Nets and Busses it uses the text height provided in the **DefaultNetNameScale** command. This height is expressed in terms of **Pin2Pin**.

In the following example DefaultNetNameScale is set to be 0.3 of Pin2Pin:

**DefaultNetNameScale = 0.3** 

### Translation of Bus and Net names

Since OrCAD Capture doesn't support bundle names like "b1,b2,b3" bundles are converted to busses and the resulted bus receives a name made from bundle members and the width of the bundle.

So instead of bundle named "b1,b2,b3", bus with name "b1\_b2\_b3[0..2]" will be created and all its members will be named accordingly.

Please, use the Mapper Section of the configuration file to change the translated name if you don't like the way of its automatic translation.

6 Key Concepts



OrCAD Capture expects a Bus member name to be the base name for the Bus name followed by a digit. If the base name itself ends on a digit it could result in wrong member Net naming during the translation.

For example Bus name CLOCK5[0..2] will result in 3 Member names: CLOCK50, CLOCK51, CLOCK52. To prevent it use **ChangeBus-BaseThatEndsOnDigit = 1** command. If it is set the previous Bus name will be translated to CLOCK5\_[0..2], with 3 Member names CLOCK5\_0, CLOCK5\_1, CLOCK5\_2.

During translation it is possible to use **SystemNetNamePrefix** option to specify the prefix that was used for system generated Nets on the system where the EDIF 2 0 0 file you are translating was created. For example use the following command to specify that "UN" was the prefix of the system-named Nets on the system where the EDIF 2 0 0 file was produced:

SystemNetNamePrefix = "UN"

EDIF2CAP will use OrCAD Capture system generated names for the Nets that have names starting with "UN" in the EDIF 2 0 0 file.

### Translation of Rippers with complex geometry from EDIF 2 0 0

OrCAD Capture supports Bus Entries(Rippers) of limited geometry. Connections between Nets and Busses are done by name in Capture. EDIF2CAP tries to replace translated Rippers with the default OrCAD Capture ones. When it becomes impossible Rippers from EDIF 2 0 0 file are converted to comment graphic on OrCAD Capture page. This keeps the schematic picture in tact and the connectivity is preserved.

# Translation of Arrayed Instances from EDIF 2 0 0

Wide instances (arrayed instances) are not supported by OrCAD Capture, so each arrayed instance is converted by EDIF2CAP to several PartInstances that are located in one point. A warning is issued to document the situation.

# Pin name and number displaying

**PinDisplayStyle = orcad** command forces EDIF2CAP to use OrCAD Capture built-in way of displaying pin names and numbers. The position, font and style of displays will be determined by OrCAD Capture. It is the most preferable way of translation.

If **PinDisplayStyle** = **vendor** is used the positions of pin name and number displays are taken from the EDIF 2 0 0 file.

Key Concepts 7



# Translation of colors and fonts

EDIF2CAP converts colors and fonts from the original system to the OrCAD Capture automatically. There are several commands that affect the color conversion

**OrcadConventions** = 1 will make EDIF2CAP to translate the EDIF 2 0 0 file in such a way that it will appear as natural OrCAD design after translation. Please check [OrCAD Writer] section of the configuration file for a better understanding what additional benefits **OrcadConventions** = 1 provide. It is recommended to use this command when translating from the other Vendors.

GenerateIniFile = 1 will force the translator to create an INI file using the data from the EDIF 2 0 0 file you are translating. This file will contain translated colors and fonts for different object types. It can be later renamed to Capture.INI and used with OrCAD Capture.

When **GenerateIniFile** = 1 is used the colors are not embedded into the objects during translation by EDIF2CAP - they can be later changed in OrCAD Capture using your preferences. It is recommended to use this command when translating from the other Vendors.

# Mapping properties and names while writing to OrCAD Capture

# Substituting names

EDIF2CAP provides a configuration file commands that let you substitute characters and names between the EDIF 200 and OrCAD Capture systems.

For example, to substitute the net name "EDIF\_NET" from the EDIF 2 0 0 file with the name "CAPTURE\_NET" in the OrCAD Capture design enter the following command in the Mapper Section of your configuration file:

setup name substitution "CAPTURE\_NET" 'EDIF\_NET" -NET

To substitute the port name "EDIF\_PORT" from and EDIF file with the name "CAPTURE\_PIN" in the OrCAD Capture design enter the following command in the Mapper Section of your configuration file: setup name substitution "CAPTURE\_PIN" "EDIF PORT" -PIN

See more about substitution of names, characters and strings in the Mapper Section of the configuration file.



# Mapping Properties

EDIF 200 properties could be mapped to OrCAD Capture properties. The following subsection explains how and when you would want to perform this task.

If in OrCAD Capture there is a property whose purpose is equivalent to a property in the system where the EDIF 2 0 0 file was generated, you can map these properties using the **Property** command. For example, if your original system ( where EDIF 2 0 0 file was generated) uses the property "PKG\_TYPE" to specify the type of the footprint to use, you can map it to OrCAD Capture property "PCB Footprint"; you could map the two properties in the following way:

property "PCB Footprint" "PKG\_TYPE" INST STRING

The OrCAD Capture design produced will contain a property called "PCB Footprint" in place of every occurrence of the EDIF 2 0 0 "PKG\_TYPE" property on the instances.

# Changing object type while writing to OrCAD Capture

In some Vendor systems the object types don't map easily to OrCAD object types.

OrCAD Writer solves this problem with the help of "ADD" command in the Mapper Section of the configuration file.

In the following example it is specified that a Cell with name "GND" needs to become a Power object in OrCAD Capture:

ADD "CELL" = "GND" {ETC CLASS=POWER OBJECT;type=string}

All instances of this Cell will become Power objects.

For more info about this command please check the documentation for the Mapper Section of configuration file.



# Translation of packaging info from EDIF 200 to OrCAD Capture

EDIF2CAP translator tries to create Packages in Capture Library basing on Instance Designators from the EDIF 2 0 0 file. Instances that have same Instance Designators are united into Packages.

Most of the times the EDIF 2 0 0 will have Instance Designators on the Instances. If it doesn't but does have Properties with same meaning, please map your properties to the properties understandable by the translator. To do it, add the following command to the Mapper section of the configuration file:

property "USER\_REFERENCE\_DESIGNATOR" yourvendor\_inst\_designator\_property

It is necessary to set Instance Designators properly when you write the EDIF 2 0 0 file before converting to OrCAD Capture. That means running "Annotate" in the Capture case (or running Packager in the case of another Vendor), before you convert the database to EDIF 2 0 0 on the original system.

For the instances that have no Instance Designators in the EDIF 2 0 0 file, the translator generates Part References in the form: U?

The Package name is taken from the Property "Source Package" of the Instances grouped into one Package.

If this Property doesn't exist on the grouped Instances, the EDIF2CAP makes up the name of the resulting Capture Package using EDIF 2 0 0 Cell names.

If it is a homogeneous Package (Instances represent one Cell), the name of the Package will be the name of a Cell.

For heterogeneous Package (Instances represent different Cells), the name of the Package is identical to the name of the first Cell.

The names of the Packages must be less then 32 characters in length (Capture limitation), so they are truncated appropriately if necessary.

The position of an Instance in a Package is taken from the Property "PositionInPackage" of that Instance. If this Property doesn't exist on the Instance the position of Instance in the Capture Package is computed, according to the Port Designators (pin numbers) of that instance or other information.



If your EDIF 2 0 0 file doesn't have those properties "Source Package", "PositionInPackage" on the Instances, but does have properties with other names, but with the same meaning, please map your properties to the properties understandable by the converter. To do it, add the following commands to the mapping section of the configuration file:

property "Source Package" "vendor\_package\_name\_property" INST STRING

property "PositionInPackage" "vendor\_position\_in\_package\_property" INST STRING Examples:

property "Source Package" "DEVICE" INST STRING property "PositionInPackage" "SEC" INST STRING

If you don't have "position\_in\_package\_property" in your Instances, Translator will try to determine positions of the Instances in the Package automatically.

If for some reasons Designators are not properly set in the EDIF file, the Packaging may not satisfy you. Please unselect "Use Designators for packaging" option in the command line in that case. It will tell the EDIF200 to Capture translator to avoid using EDIF Instance Designators during conversion. In that case translator will package the data according to the "Source Package" property on the Instances in the EDIF file. All Instances having that "Source Package" Property with the same Value will be packaged into same Packages.

# **EDIF 200 to OrCAD Capture Translator Configuration** file

OrCAD Capture to EDIF 200 Translator Configuration file contains two sections:

- [OrCAD Writer Section] contains Translator options (arguments).
- [Mapper Section] contains commands to add/remove/map properties and names.

Default name for the configuration file used in the Import to Capture is **edi2cap.cfg** 



# OrCAD Writer Section

#### **SuppressWarnings**

Full name: SuppressWarnings

Syntax:  $Suppress\ Warnings = 0\ or\ Suppress\ Warnings = 1$ 

Default: SuppressWarnings = 0

By default, translator outputs warnings errors and translation actions to the log. To disable output of this information set **SuppressWarnings** = **1**. Summary information (Errors/Warnings count) is always output to log.

#### **ConvertOnlyLibs**

Full name: Convert Only Libraries

Syntax: ConvertOnlyLibs = 0 or ConvertOnlyLibs = 1

Default: ConvertOnlyLibs = 0

If **ConvertOnlyLibs** option is 1 only capture library files (.olb) will be created. Schematic design information will be ignored and capture design file (.dsn) will not be created.

#### **ProcessBackAnnotation**

Full name: Process Back Annotation

Syntax:  $Processt\ Back\ Annotation = 0\ or\ Process\ Back\ Annotation = 1$ 

Default: *Process Back Annotation* = 1

Set **ProcessBackAnnotation** = **1** to translate backannotation occurrence data from the EDIF 2 0 0 file to OrCAD Capture design.

#### **ComponentsMode**

Full name: Components Mode

Syntax: Components Mode = 0 or Components Mode = 1

Default:  $Components \ Mode = 0$ 

This mode similar to **ConvertOnlyLibs** but each package from the design will be saved into separate Capture Library file (.olb). Those files will be placed into directories with the libraries' names. Package name will match file name so characters restricted by file system will be replaced with "\_".



#### <u>UnitMeasurement</u>

Full name: **Design unit measurement** 

Syntax: *UnitMeasurement* = *default or UnitMeasurement* = *inch or* 

UnitMeasurement = mm

Default: *UnitMeasurement* = *default* 

Specifies unit measurement with which the converted design was created (inches or millimeters). default UnitMeasurement means that measurement unit will be determined by translator automatically.

#### **PinToPin**

Full name: Minimal Pin to pin spacing

Syntax: PinToPin = <*float number*>

Typical values: 0 or 0.1 in inches or 1 in mms

Default: PinToPin = 0

OrCAD internal **PinToPin** spacing is 0.1 inch, EDIF file can have a different **PinToPin** spacing. By specifying this parameter you are forcing the translator to convert to the pin spacing of your choice.

Specify the spacing between pins in units set by **UnitMeasurement**.

If **PinToPin** is set to 0 translator determines pin spacing automatically.

For example design can come with 0.25 pin spacing. If the **PinToPin** parameter will be set to default, translator will determine that design has 0.25 pin spacing and will convert using 0.25 as **Minimal Pin to pin** spacing.

Most of the time you will want the pin spacing to be 0.1 inch, because it's OrCAD Capture default.

Please specify 0.1 inch when translator automatically had determined different pin spacing.

Make sure that you enable off-grid objects if the pin spacing you choose is different from the one calculated by the translator.

#### **Grid**

Full name: Grid Resolution Syntax:  $Grid = \langle number \rangle$ 

Typical values: 0 Default: Grid = 0

Set Grid resolution in Design Units. If Grid resolution equal to 0 trans-

lator will determine the grid resolution automatically.

This option is only for advanced users.



#### **EnableOffgridObjects**

Full name: Enable off-grid objects

Syntax:  $Enable\ off\ objects=0\ or\ Enable\ off\ objects=1$ 

Default:  $Enable\ off\ grid\ objects=0$ 

Set **EnableOffgridObjects** to 1 if you had specified non **default Pin-ToPin** and want some of your objects to be off-grid. When **default Pin-ToPin** is selected translator determines itself to place all objects on grid or not.

#### **GenerateIniFile**

Full name: Generate INI File

Syntax: GenerateIniFile = 0 or GenerateIniFile = 1

Default: *GenerateIniFile* = 1

Set **GenerateIniFile** to 1 to create Capture's initialization file. The name of the INI file will be <Design Name>.ini. INI file will store original colors and fonts heights from the EDIF file.

#### **OrcadConventions**

Full name: **OrCAD Conventions** 

Syntax: OrcadConventions = 0 or OrcadConventions = 1

Default: *OrcadConventions* = 1

**OrcadConventions** forces writer to generate design with OrCAD-specific style. Selecting **OrcadConventions** implies the following:

- OrCAD Capture style of page will be used, all colors will be set to default Capture colors
- Translator will try to restore OrCAD pin styles
- White background will be used

Translator will try to create a design that appears as natural "Capture" design.

#### <u>UseDesignatorsForPackaging</u>

Full name: Use designators for packaging

Syntax: UseDesignatorsForPackaging = 0 or UseDesignatorsFor-

Packaging = 1

Default: UseDesignatorsForPackaging = 1

**UseDesignatorsForPackaging** enables/disables usage of Reference Designators from the EDIF instances for Packaging. By default (recommended) they are used for Packaging.

This option is for Advanced Users only.



#### **DesignatorsEndOnLetters**

Full name: **Designators Can End On Letters** 

Syntax: Designators Can End On Letters =  $\theta$  or Designators Can End

On Letters = 1

Default: Designators Can End On Letters =  $\theta$ 

Set **DesignatorsEndOnLetters** to 1 to enable Reference Designators to end on letters. In analog designs, a letter at the end of the transistor reference designator means the power of this transistor, not the place of transistor in the Package. So for analog designs this option must be set. This option is for Advanced Users only.

#### **DesignatorsSameInHierarchy**

Full name: Duplicate designators in hierarchy are identical

Syntax: DesignatorsSameInHierarchy = 0 or DesignatorsSameInHierarchy = 1

Default: DesignatorsSameInHierarchy = 1

Specifies how to treat duplicated designators in hierarchies. If **DesignatorsSameInHierarchy** is set to 0- then Reference Designators are considered different if they are on different levels of the hierarchy in an EDIF file. By default **DesignatorsSameInHierarchy** is 1 and hierarchical level is not taken into account.

This option is for Advanced Users only.

#### <u>CaptureLibsWithUserPackages</u>

Full name: Take packages from Capture libraries

Syntax: CaptureLibsWithUserPackages = "<comma-separated string with paths to libraries>"

Default: CaptureLibsWithUserPackages = ""

**CaptureLibsWithUserPackages** specifies paths to OrCAD Capture Libraries (comma separated list) to look for Packages. If the translator finds appropriate Package it takes it (adds to design) from the existing OrCAD Capture Library.

This option is useful when you are doing translation for the second time and would like to take Packages from the libraries you created in the first run.

Suppose we have 2 existing Libraries "ttl.olb", "device.olb" located in directory c:\work\mylibs. In the following example EDIF2CAP will be instructed to search for the packages and symbols in the libraries "ttl.olb", "device.olb" before creation of the new packages and symbols in new libraries. If the package will be found in existing library it will be taken from it.



#### Example:

CaptureLibsWithUserPackages="c:\work\mylibs\ttl.olb,c:\work\mylibs\device.olb"

#### **GeneratePinNumbers**

Full name: Generate Pin Numbers

Syntax: GeneratePinNumbers = 0 or GeneratePinNumbers = 1

Default: GeneratePinNumbers = 0

Set **GeneratePinNumbers** to 1 if you want to generate pin numbers for pins that don't have pin number in an EDIF file. The first pin in the interface of the cell will get number "1", next "2", etc...

#### **GeneratePortSymbols**

Full name: Generate non-implemented port symbols

Syntax: GeneratePortSymbols = 0 or GeneratePortSymbols = 1

Default: *GeneratePortSymbols* = 1

Set **GeneratePortSymbols** to 0 if you don't want to generate on the schematic Hierarchical Ports that were absent in an EDIF file. By default translator generates missing hierarchical Ports on the Schematic Page if the ports they implement were present in EDIF netlist.

It is necessary to have Hierarchical Ports on the Schematic to establish connectivity between 2 levels of hierarchy.

#### <u>GenerateOffPageConnectors</u>

Full name: Generate non-implemented off-page connectors

Syntax: GenerateOffPageConnectors = 0 or GenerateOffPageConnectors = 1

Default: GenerateOffPageConnectors = 1

Disable **GenerateOffPageConnectors** if you don't want to enable generation of OffPageConnectors that were absent in an EDIF file. By default translator generates missing OffPageConnectors between same Schematic Nets on different Pages of the Design.

#### **GenerationVisible**

Full name: Make Generated Objects Visible

Syntax: GenerationVisible = 0 or GenerationVisible = 1

Default: Generation Visible = 1

Set **GenerationVisible** to 1 to make generated (if **GenerateOffPage-Connectors** or **GeneratePortSymbols** are enabled) Hierarchical Ports and OffPageConnectors visible.

Generated aliases become visible too if **GenerationVisible** is enabled.

If **GenerationVisible** is disabled then generated aliases are created in the background (white) color.



#### **SystemNetNamePrefix**

Full name: System net name prefix

Syntax: SystemNetNamePrefix = "<string>"

Default: SystemNetNamePrefix = ""

**SystemNetNamePrefix** allows you to specify prefix used in system-generated net names present in the EDIF file. Specifying this option for net names forces translator to create OrCAD system net names instead of the ones (with this prefix) used in the system from which the EDIF file was obtained.

Example: if you have system nets with "UN" prefix in the original system from which EDIF file

was generated, use SystemNetNamePrefix="UN" and all nets with names starting from "UN"

will have standard OrCAD generated names like "N0001", etc...

#### **BackgroundTextScale**

Full name: Background text scale

Syntax: *BackgroundTextScale* = <*number*>

Default: BackgroundTextScale = 0.8

**BackgroundTextScale** sets the background text scale (value 0.8 means that it is 0.8 of **PinToPin**). It allows to influence the size of texts generated in the background. This is especially important for printing.

The aliases are generated in the background if **GenerationVisible** is not enabled.

#### **DefaultNetNameScale**

Full name: **Default net names text scale** 

Syntax: *DefaultNetNameScale* = <*number*>

Default: DefaultNetNameScale = 0.3

During the conversion to OrCAD Capture translator has to create Aliases for OrCAD Capture nets. The Aliases are created in the background color if **GenerationVisible** is disabled or in the foreground color if **GenerationVisible** is enabled. Choosing **DefaultNetNameScale** sets the net names text scale of generated aliases (value 0.8 means that it is 0.8 of **PinToPin**).



#### **StandardPageSize**

Full name: Standard Page Size

Syntax: StandardPageSize = 0 or StandardPageSize = 1

Default: StandardPageSize = 0

Set **StandardPageSize** to 1 if you want translator to try to use standard page sizes for the created pages during conversion. If the **UnitMeasurement** is in millimeters, then metric page sizes will be used.

#### **EliminatePageText**

Full name: Eliminate Page Text

Syntax: EliminatePageText = 0 or EliminatePageText = 1

Default: EliminatePageText = 0

Set **EliminatePageText** to 1 if you want translator to avoid translation of page texts during conversion. This option helps to clean the output after conversion from some EDIF files.

#### **SmashTitleBlock**

Full name: Smash Title Block

Syntax: SmashTitleBlock = 0 or SmashTitleBlock = 1

Default: SmashTitleBlock = 0

Select **SmashTitleBlock** if you want translator to convert Title Blocks to comment graphics on the page. This is important if the size of the

Title Block is near the size of converted page.

#### **PinDisplayStyle**

Full name: Pin Display Style

Syntax: PinDisplayStyle = orcad or PinDisplayStyle = vendor or Pin-

DisplayStyle = nopinnames

Default: *PinDisplayStyle = orcad* 

Set **PinDisplayStyle** = *orcad* to use standard Capture's built-in style of displaying. Capture itself will determine pin name display location and pin number display location and fonts to use.

Set **PinDisplayStyle** = *vendor* to use original Vendor style of displaying pin names and numbers. In this case pin name and number properties display are placed at the locations converted from the EDIF pin name and number displays.

Set **PinDisplayStyle** = *nopinnames* to prevent pin names to be displayed on the Part.



#### **TopView**

Full name: Top View Name for translation

Syntax: TopView = "<string>"

Default: TopView = ""

Specify **TopView** in the following format: Library\Cell\View. It will tell the converter what view to select as the root for the translation. If **TopView** is not set or set to empty string translator determines top view automatically. If many equal top views exists in the EDIF file translator converts first found.

#### **ReplacePartWithNoPinsWithTitleBlock**

Full name: Replace Part With No Pins With Title Block

Syntax: ReplacePartWithNoPinsWithTitleBlock = 0 or Replace-

PartWithNoPinsWithTitleBlock = 1

Default: ReplacePartWithNoPinsWithTitleBlock = 0

Set ReplacePartWithNoPinsWithTitleBlock to 1 to translate parts

with no pins to Title Blocks.

#### <u>ChangeBusBaseThatEndsOnDigit</u>

Full name: Change Bus Base That Ends On Digit

 $Syntax: Change Bus Base That Ends On Digit = 0 \ or \ Change Bus Base That Ends On Digit = 0 \ or \ Change Bus Base That Digit = 0 \ or \ Change$ 

tEndsOnDigit = 1

Default: ChangeBusBaseThatEndsOnDigit = 1

ChangeBusBaseThatEndsOnDigit helps to process busses with bases that end on digit. To separate bus base name that ends on digit from the range of the bus translator uses symbol '\_'. It will also add aliases to the nets, using bus base name and position of net in the bus to create the alias.

#### Example:

If this options is set then the Bus DATA2[0..12] with members: DATA20 ... DATA212 will be translated to DATA2\_[0..12] and DATA20 will have alias DATA2\_0, etc...



#### **Mapper Section**

#### Mapping commands:

#### <u>ADD</u>

#### Usage

```
ADD "CELL"="*" {<new_prop>=<new_value>}
ADD "CELL"="<edif200_identifier>"
{<new_prop>=<new_value>}
ADD <prop>=<value> {<new_prop>=<new_value>}
```

#### **Description**

A general mechanism to add properties to converted cells.

This command has the following behavior:

```
ADD "CELL"="*" {<new_prop>=<new_value>}
```

Property with name <new\_prop> and value <new\_value> will be added to all cells.

```
ADD "CELL"="<edif200_identifier>"
{<new_prop>=<new_value>}
```

Property with name <new\_prop> and value <new\_value> will be added to the cell with identifier equal to <edif200\_identifier>.

```
ADD ADD cyalue> {<new_prop>=<new_value>}
```

#### **Arguments**

```
string with property name.
```

<value>: symbol '\*' or property value.

Note: you must specify <prop>="\*" or <prop>="value", not <prop>=\* or <prop>=value.

<edif200\_identifier> - EDIF cell name.

<new\_prop>: Name of the new property to add to the cell.

<new\_value>: Value of the new property to add to the cell.

#### **Examples**

```
Add the property MYCELLPROP = "0" on all cells 

ADD "CELL" = "*" {MYCELLPROP = "0"}
```

In some Vendor systems the object types don't map easily to OrCAD object types. For example some EDIF200 Writers output Powers or TitleBlocks as logic Cells. OrCAD Writer solves this problem with the help of configuration file commands.



To specify that the Cell with name "GND" must be Power object in OrCAD use: **ADD** "CELL"="GND"

{ETC\_CLASS=POWER\_OBJECT;type=string}

All instances of this Cell will become Power objects.

To specify that the Cell with name "+3V" must be Power object in OrCAD use: **ADD** "CELL"="+3V"

{ETC\_CLASS=POWER\_OBJECT;type=string}

All instances of this Cell will become Power objects.

To specify that the Cell with name "TitleBlock" must be TitleBlock object in OrCAD use: **ADD** "CELL"="TitleBlock" {ETC\_CLASS=TITLEBLOCK;type=string}

All instances of this Cell will become Title blocks.

To smash the Cell with name "Example" to comment graphic in OrCAD use: **ADD** "CELL"="Example" {**ETC\_CLASS=COM-MENT;type=string**} All instances of this Cell will be smashed to comment graphics on OrCAD Page.

Special command can be used to control the level of design Hierarchy used for Packaging. If a cell in the EDIF file contains an underlying Schematic and you do not want to process this Schematic by EDIF2CAP packaging algorithm, use the following example:

ADD "CELL"="MyCell" {ETC\_CLASS=FLAT;type=string}

When the EDIF2CAP packaging algorithm processes the cell with property ETC\_CLASS with the value FLAT it will not process underlying Schematic.

To specify that the Cell is a PRIMITIVE in OrCAD use:

ADD "CELL"=" MyCell " {ETC\_CLASS= PRIMITIVE\_CELL;type=string}

All instances of this Cell will be considered PRIMITIVE.

To specify that the Cell is a MODEL in OrCAD use:

ADD "CELL"=" MyCell " {ETC\_CLASS= MODEL\_CELL;type=string}

#### **Property**

#### Usage

property propertyName\_capture propertyName\_edif owner propertyType



#### **Description**

This command enables mapping property name on the object.

#### **Arguments**

<Owner>: The object in which you want to map the Property.

<PropertyType>: Type of Property: INTEGER, REAL, BOOLEAN,
STRING.

#### **Example:**

property "Source Package" "PART\_NAME" INST STRING

The preceding example substitutes the name of "PART\_NAME" property on an EDIF Instance, with name "Source Package" on the corresponding Capture Instance.

If your EDIF file doesn't have Reference Designators on the Instances but does have Properties with same meaning, please map your properties to the properties understood by the converter. To do it, use mapping file commands:

property "USER\_REFERENCE\_DESIGNATOR" your\_vendor\_instance\_designator\_property

If your EDIF file doesn't have Reference Designator Templates on the Interface of a Cell

but does have Properties with same meaning, please map your properties to the properties understood by the converter. To do it, use mapping file commands:

property "USER\_REFERENCE\_TEMPLATE" your\_vendor\_symbol\_designator\_template\_property

If your EDIF file doesn't have Designators on the Ports of a Cell (designators on ports represent pin numbers in the original vendor system), but does have Properties with same meaning, please map your properties to the properties understood by the translator using a mapping file command:

property "USER\_PIN\_NUMBER" your\_vendor\_port\_pin\_number\_property

If your EDIF file doesn't have Properties "Source Package", "Position-InPackage" on the Instances, but does have Properties with other names, but with same meaning, please map your properties to the properties understood by the converter by using mapping file commands:

property "Source Package" "your\_vendor\_package\_property" INST STRING
property "PositionInPackage" "your\_vendor\_position\_in\_package\_property" INST STRING



#### Substitution of Names or Strings

#### Usage

setup name substitution "capture\_name" "edif\_name" owner setup string substitution "capture\_name" "edif\_name" owner setup character substitution "capture\_string" "edif\_string" mode

#### **Description**

The "setup name substitution" command lets you replace names of pins, nets, symbols, and instances.

The "setup character substitution" command lets you specify the characters and substitution set which should replace them in the Capture design being created. This command supports the substitution of (as an example) illegal characters in name strings. Two argument strings (capture\_string and edif\_string) specify character mapping. Characters in these strings will be mapped by position. So the first character in the edif\_string will be replaced by the first character in the capture\_string, second by second etc. If the edif\_string has a length greater than the capture\_string the characters in the tail of the edif\_string string will be removed from the names in the generated design. If the capture\_string is longer the tail of the capture\_string will be ignored.

The "setup string substitution" command allows you to change the value of the properties attached to pins, nets, instances and symbols.

#### Arguments

- <capture\_name>: Object name you want to replace with.
- <edif\_name>: The name of the object you want to find and replace in the source database.
- <owner>: specifies the type of object that owns the name or string:
- -PIN allows you to change the name or property of a pin.
- -NET allows you to change the name or property of a net.
- -INST allows you to change the name or property of an instance.
- -SYM allows you to change the name or property of a symbol (this corresponds to the EDIF cell).
- <edif\_string>: The string of characters that must be replaced with characters specified in the capture\_string
- <capture\_string>: The string of characters that must replace characters
  specified in the edif\_string
- <mode>:
- -Names: The mode switch specifies the type of objects for which the substitution applies.



#### **Examples:**

1) When writing a Capture file, if you want to replace EDIF Instance name "EdifInstanceNumber123" with the Capture Instance name "I123", use the following command:

setup name substitution "I123" " EdifInstanceNumber123" -INST

2) This command replaces all '<'and '>' characters by '[' and ']'. As example Bus name A<1..3> will be replaces by A[1..3]. setup character substitution "[]" "<>" -Names

#### forgetproperty

#### **Usage**

forgetproperty <prop>

#### **Description**

A general mechanism to remove properties from parts, instances, nets, pins and other objects.

#### **Arguments**

The Name of the property to remove

#### **Examples**

This command removes property BadInternalHiddenEdif-Property from objects in the design.

forgetproperty "BadInternalHiddenEdifProperty"

## **OrCAD Capture to EDIF 200 Mapping**

This section explains the similarities and differences between terminology in the OrCAD Capture and EDIF 200 systems. It is not intended to be a comprehensive mapping, but rather an aid to familiarize the OrCAD Capture user with EDIF 200 terminology, and vice versa. For more information on any of the EDIF concepts presented in this section, refer to the *Electronic Design Interchange Format* documents, which can be obtained from the Electronic Industries Association. For more information on the OrCAD Capture design concepts presented in this section, refer to the OrCAD Capture user's manual.

#### Libraries

Libraries, in both EDIF and OrCAD Capture environments, are groupings of parts and designs based on a set of common characteristics. A single EDIF file may contain descriptions of many libraries. A single design can reference parts in several different libraries.



### Cells, Parts, Schematics and Instances

The EDIF 2 0 0 cell represents OrCAD Capture cell. An EDIF 2 0 0 view represents OrCAD Capture Library Part, contents of the EDIF 2 0 0 view represent OrCAD Capture Schematic. EDIF 2 0 0 cell can have multiple views that can be used for different purposes. Sometimes the views of the cell are used to represent alternative symbols (e.g. DeMorgan equivalents). In OrCAD Capture only 2 Views of the Library Part are allowed. They are called "Normal" and "Convert".

If EDIF 2 0 0 cell has 2 views that cannot be considered "Normal" and "Convert" or EDIF 2 0 0 cell has more than 2 Views, then EDIF 2 0 0 cell with multiple views will be translated to several cells with one ("Normal") view in OrCAD Capture. EDIF 2 0 0 instance will be translated to OrCAD Capture instance. OrCAD Capture Package could be Homogeneous or Heterogeneous. OrCAD Capture Packages are restored using EDIF 2 0 0 Instance Designators and specific packaging properties on EDIF 2 0 0 objects.

# Attributes and Properties

Both EDIF and OrCAD Capture systems use objects called *properties*, which give more information about an object than just basic connectivity or graphics. Properties have two parts: a *name* and a *value*.

Additionally, EDIF has a set of commonly used properties called *attributes*. The EDIF attribute names are special reserved words (or keywords). As with any property, each EDIF attribute has an owner-either instance, net, or port (pin)--for which it is valid.

Because of the many types of properties design objects can have, only certain types of properties have EDIF attributes. For example, the OrCAD Capture instance property part reference has an equivalent EDIF *attribute*, named *designator*, whose owner is an instance.

Table 1 lists and briefly explains each of the EDIF attributes.

**Table 1: EDIF Attributes Summary** 

Attribute	EDIF Type	Description
acload	miNoMax	An attribute of a port used to express external load capacitance.
criticality	integer	A positive or negative integer value used to describe the relative importance of a net to other nets for routing purposes.
dcfaninload	number	An attribute used to compute the fan-in load of an output or input port.



Table 1: EDIF Attributes Summary, continued

Attribute	EDIF Type	Description
dcfanoutload	number	An attribute used to compute the fan-out load of an output or inout port.
dcmaxfanin	number	An attribute that specifies the maximum allowed fanin of input or inout ports.
dcmaxfanout	number	An attribute that specifies the maximum allowed fanout of output or inout ports.
designator	string	An attribute which specifies a pin number or a reference designator for a cell instance.
direction	INOUT INPUT OUTPUT	An attribute used to specify direction of a port, such as input, output, or inout.
unused	N/A	An attribute of ports, port instances, and off-page connectors indicating the object is not used in the interface of this particular view.

#### **Ports and Pins**

The EDIF term *port* encapsulates the OrCAD Capture term *pin* and term *Hierarchical Port*. *Ports* are connection points to cells.

#### **Nets**

In both OrCAD Capture and EDIF terminologies, a net is used to connect instances in a design. In EDIF V  $2\,0\,0$ , the "net" construct together with the "joined" construct, describes how a net is connected to ports of instances in a design.

#### **Net Buses**

OrCAD Capture buses map to EDIF V 2 0 0 "net array" constructs.