



Testabdeckungsanalyse

Embedded Board Test für Ihre Platine



Die folgenden Formate werden von GÖPEL electronic für eine Boundary Scan Testabdeckungsanalyse unterstützt.

Häufig verwendete Formate

Format	Typ	Endung	Beschreibung	Link zum Beispielfile:
Altium Designer	Netzliste	*.net	Schaltplan öffnen -> Menu -> Design -> Netlist for Project -> Protel2	
Altium Designer	Layout	*.pcb	Layout öffnen -> Save Project As -> Export Protel PCB 2.8 ASCII	
ODB++	Netzliste & Layout			



Alle Netzlistenformate

Format	Beschreibung	Syntaxbeispiel
Altium Designer	Schaltplan öffnen -> Menu -> Design -> Netlist for Project -> Protel2 *.net	
Agilent_3070		!----- !Agilent_3070 PHYSICAL DESCRIPTION FILE !WED DEC 06, 2000 03:29:39 PM ! !BOARD_XY !----- ... !! BAUELEMENTEINFORMATION CAPACITOR C1 10U 20 20 F "ELECT_260_10UF"; C2 100N 20 20 F "0805_0_1UF"; ... RESISTOR R1 100 5 5 F "2010_100"; R2 22K 5 5 F "1206_22K"; PIN LIBRARY U4 PN"MC34064A" "SO8_MC34064A"; U5 PN"74HCT132" "SO14_74AHCT132"; !! NETZE UND PINS CONNECTIONS AGND C1.2 C2.2 ... NETZ2 U4.1 C1.1; ...



Format	Beschreibung	Syntaxbeispiel
ALLEGRO_CAD		<pre> A!NET_NAME!REFDES!PIN_NUMBER!COMP_PART_NAME!COMP_VALUE!COMP_TOL!COMP_PACKAGE!COMP_DES CRPTION! J!/beispiel/boardfile.brd!Mon Dec 14 16:35:49 1998 ... 40000.00!0.01!mils!BSP1!80.6748 mil!14! S!!Q1!4!LT1587CM!!!DPAK_3LEAD!3A LOW DROPOUT REGUL.FIX.1V5 DPAK! S!!!!!!!!! S!!ZM7!AB18!CAPOC48!!!SBC32A34V04!IBM CAP OC48 CCGA32MM! S!!!!!!!!! S!!ZM7!W17!CAPOC48!!!SBC32A34V04!IBM CAP OC48 CCGA32MM! S!!!!!!!!! S!!ZM7!T08!CAPOC48!!!SBC32A34V04!IBM CAP OC48 CCGA32MM! S!!!!!!!!! S!!ZM7!T18!CAPOC48!!!SBC32A34V04!IBM CAP OC48 CCGA32MM! ... - So sehen die Netzeinträge aus: S!UN824C01A150PTEST0!R317!2!RES_5_62!220!+5-5!RES0603_REF!RESISTOR 220 OHM 5% 1/16W SMT0603! S!UN824C01A150PTEST0!ZM39!7!24C01A!!!DIP08_300_SOCKET_SMT!SOCKET SMT/PIH 8POS .300! S!UN824C01A150PTEST0!!!!!!! </pre>
ALLEGRO/ECI-Format	<p>Das ALLEGRO/ECI-Format steht ab CASCON-Version 3.2.2 zur Verfügung. Zum Import der Daten wird eine Datei benötigt.</p> <p>Der Abschnitt der Bauelementeliste wird mit dem Schlüsselwort \$PACKAGES eingeleitet und mit dem Schlüsselwort \$A_PROPERTIES abgeschlossen.</p> <p>Die Bauelemente sind typorientiert in der Datei abgespeichert. Der Typbezeichner vor dem „!“-Zeichen (Ausrufezeichen) wird mit dem nachfolgenden Bezeichner zu einem Typnamen verbunden. Sie sind dabei</p>	<p>Beispiel:</p> <pre> \$PACKAGES ... TYPE1 ! 'abc'; C16 C85 J1C J1D TYPE2 ! 'cde'; J3A J3B J3C J3D TYPEXY! 'xyz'; J4A L7 J4C J4D , J5E J5F J5Z J6A JP3 J6C J6D J6E ... \$A_PROPERTIES </pre> <p>Die Netzliste beginnt mit dem Schlüsselwort \$NETS und wird ebenfalls mit dem Schlüsselwort \$A_PROPERTIES beendet. Das Dateiende wird mit dem Schlüsselwort \$END markiert.</p> <p>Die Netze werden einfach mit Namen, in Hochkomma eingeschlossen, aufgelistet. Danach werden die Pins in einer Liste aufgeführt. Das Semikolon (;) leitet den Listenbeginn ein.</p> <p>Beispiel:</p> <pre> \$NETS </pre>



Format	Beschreibung	Syntaxbeispiel
	durch einen Unterstrich (_) im Namen getrennt.	<pre> ... 'Netz1\$1' ; P91.1 U45.44 U49.44 U85.12 'Netz2' ; C16.1 C85.1 C86.1 C87.1 C165.1 C170.1, JP3.2 L7.1 L18.1 L26.1 P28.1 'Netz_xy' ; C17.1 C89.1 C166.1 C167.1, \$A_PROPERTIES ... \$END </pre>
ARIADNE-Format		<pre> *ARIADNE* *PCB* *PART* D7 R8069BJ[PLCC]:PLCC44 D2 R8075J[PLCC]:PLCC28 D17 R8070J[PLCC]:PLCC68 X4 DS09BW:DS09BW ...<Bauelementeliste geht weiter > *NET* *SIGNAL* \$1 12 D7.31 D2.2 D17.5 *SIGNAL* \$2 12 D4.6 D2.4 *END* </pre>
CADENCE_3-Format	<p>Um mit dem CADENCE_3 Filter Daten zu importieren, sind zwei Dateien notwendig.</p> <p>Bauelementeliste: *.DAT</p> <p>Netzliste: *.DAT</p>	<p>Jeder Bauelementeeintrag wird mit dem Schlüsselwort PART_NAME eingeleitet. Es folgt der Bauelementebezeichner und danach der Typstring, der in Hochkommas eingeschlossen wird. Im Typstring wird eine Liste angegeben, getrennt durch Komma-Zeichen (,). Der zweite Listeneintrag wird als Typbezeichner importiert. Das Ende eines Datensatzes wird durch ein Semikolon (;) beschrieben. Der Typstring kann auch leer sein. In diesem Fall wird der Typ NOTYPED vergeben.</p> <p>Beispiel:</p>



Format	Beschreibung	Syntaxbeispiel
		<p>PART_NAME U1 '18374-18374,SCAN18374TSSC,SMT,A';</p> <p>...</p> <p>PART_NAME U2 "';</p> <p>...</p> <p>END.</p> <p>Jeder Eintrag in der Netzliste wird mit dem Schlüsselwort NET_NAME eingeleitet. Es folgt der Netzname, der in Hochkommas eingeschlossen wird, gefolgt von den jeweiligen Pineinträgen. Jeder Pineintrag wird mit dem Schlüsselwort NODE_NAME eingeleitet. Das Datensatzende ist das Semikolon (;).</p> <p>Beispiel:</p> <p>NET_NAME 'A_INTER7' ...; NODE_NAME U4\$+* 44 ...; NODE_NAME U2 13 ...; NET_NAME 'A_INTER8' ... NODE_NAME U4 43 ...; NODE_NAME U2 14 ...; ... END.</p>
CADENCE_GEN-Format	Das CADENCE_GEN-Format benötigt für den Import zwei Dateien. Sollen die Daten aus derselben Datei gelesen werden, so muss das CAD-File zweimal eingetragen werden.	<p>Bauelemente: REFDES Netze: NET_NAME Pins: PIN_NUMBER</p> <p>Desweiteren entspricht das CADENCE_GEN-Format dem ALLEGRO CAD-Format. Die Reihenfolge der Elemente ist</p>



Format	Beschreibung	Syntaxbeispiel
		hier variabel. Die Bezeichner werden so importiert, wie sie in der Datei stehen. Das Ausrufezeichen (!) trennt die Dateneinträge.
CADENCE_NETLIST-Format	Um den Import mit dem CADENCE_NETLIST-Format durchzuführen, ist eine Datei erforderlich.	<pre> PROJEKTNAME MN10 16 NET_3 EPM7128AX144_QFPIO3V3CMOS P3 78 NET_4 CONN80R2 MN10 10 NET_1 EPM7128AX144_QFPIO3V3CMOS P3 73 NET_2 CONN80R2 R4 18 NET_1 MNR14X </pre>
CADIF-Format	Es ist eine Netzlistendatei (*.PAF) erforderlich. Die Netznamen werden aus einer Signalliste aus dieser Datei gelesen. Die Syntax ist eine Klammersyntax.	<pre> (CADIF (FORMAT CADIF 4 0) (DESIGN ... (SIGNALLIST (SIGNAL S1 (NAME "\$1I1979\$1N1290") (ATTRIBUTE "SPACING_CLASS" "OTHER")) (SIGNAL S2 (NAME "\$1I1979\$1N1297") (ATTRIBUTE "SPACING_CLASS" "OTHER")) ... (PACKAGE PAC3 (NAME "C01N00000") (PACKDESC (SYMNAME "C01N00000") (LEADFORM SMD) (PACKTERM T1 (NAME "1") (ID 1)) -- PIN INFO (PACKTERM T2 (NAME "2") (ID 2)) -- PIN 2 (PART PRT1 (NAME "ATE") -- IC TYPE ATE PARTDESC PACKREF PAC3) ... (COMPONENT C1 (NAME "AT13101") -- IC-NAME (COMPDEFN (PARTREF PRT1) (PACKALTREF ALT1)) ... (NET N1 (SIGNALREF S1) (NETJOINS </pre>



Format	Beschreibung	Syntaxbeispiel
		<pre>(COMPPINREF T2 (COMPREF C1)) (COMPPINREF T1 (COMPREF C347))) ... (FILECHECK 134614401 176520 2053043)))</pre>
CADNETIX-Format		<pre>PARTS LIST 00104097 00104097 IC4 00108077 00108077 IC1C 00108077 00108077 IC93 00231002 00231002 L1 ... NET LIST NODE 1 \$ IC1C 144 TP6 1 R9 1 NODE 2 \$ TP35 1 R164 1 IC4 89 IC4 87 \$ IC4 39 IC4 37 ... NODENAME BCLK_CPU \$ IC1A 128 IC2A 124 IC47 84 TP179 1</pre>
CADSTAR-Format	<p>Typischerweise werden zwei Dateien für das CADSTAR-Format benötigt: Die Bauelementeinformationen sind im *.GAT -File verfügbar und die der Netze/ Pins im *.NET -File.</p>	<pre>Drawing name : PIO.CSD U00001 N374T U1 SCAN18374T U00002 N374T U2 SCAN18374T U00003 N374T U3 SCAN18374T ... U00012 RES R32 1K U00013 RES R14 1K ...</pre> <p>Der Bauelementebezeichner wird von der dritten Spalte gelesen, der Modelltyp von der vierten Spalte.</p>



Format	Beschreibung	Syntaxbeispiel																																				
Capture_9_PCB-Format	Für den CAD-Datenimport des Capture_9_PCB-Formats sind zwei Dateien erforderlich. Die Bauelementliste erhält defaultmäßig die Erweiterung *.CMP und die Netzliste die Erweiterung *.NET. Dieser CAD-Import ist ab der Version 3.4.2 von SYSTEM CASCON™/ CASCON POLARIS™ standardmäßig verfügbar.	<p>Die Syntax der Bauelementedatei ist in der folgenden Tabelle dargestellt:</p> <table border="1"> <thead> <tr> <th>Befehl</th> <th>Beispiel</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>.PCB</td> <td>.PCB</td> <td>Format Header</td> </tr> <tr> <td>.COM</td> <td>.COM</td> <td>Bauelementliste</td> </tr> <tr> <td>.REM</td> <td>.REM EPF10k30</td> <td>Einleitung Bauelement</td> </tr> <tr> <td>.REM <DEV> <Type></td> <td>.REM U1 EPF10k30</td> <td>Bauelementeeintrag</td> </tr> <tr> <td>.EOD</td> <td></td> <td>Dateiende</td> </tr> </tbody> </table> <p>Die Netzliste ist ähnlich wie die Bauelementliste aufgebaut. Es folgt die Syntaxtabelle für die Netzliste:</p> <table border="1"> <thead> <tr> <th>Befehl</th> <th>Beispiel</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>.PCB</td> <td>.PCB</td> <td>Format Header</td> </tr> <tr> <td>.COM</td> <td>.COM</td> <td>Kennzeichen Netzliste</td> </tr> <tr> <td>.REM</td> <td>.REM VCC_NET</td> <td>Einleitung Netzeintrag</td> </tr> <tr> <td>.REM <netname> <DEV> <Pin></td> <td>.REM VCC_NET U1 1</td> <td>Bauelementeeintrag</td> </tr> <tr> <td>.EOD</td> <td></td> <td>Dateiende</td> </tr> </tbody> </table>	Befehl	Beispiel	Beschreibung	.PCB	.PCB	Format Header	.COM	.COM	Bauelementliste	.REM	.REM EPF10k30	Einleitung Bauelement	.REM <DEV> <Type>	.REM U1 EPF10k30	Bauelementeeintrag	.EOD		Dateiende	Befehl	Beispiel	Beschreibung	.PCB	.PCB	Format Header	.COM	.COM	Kennzeichen Netzliste	.REM	.REM VCC_NET	Einleitung Netzeintrag	.REM <netname> <DEV> <Pin>	.REM VCC_NET U1 1	Bauelementeeintrag	.EOD		Dateiende
Befehl	Beispiel	Beschreibung																																				
.PCB	.PCB	Format Header																																				
.COM	.COM	Bauelementliste																																				
.REM	.REM EPF10k30	Einleitung Bauelement																																				
.REM <DEV> <Type>	.REM U1 EPF10k30	Bauelementeeintrag																																				
.EOD		Dateiende																																				
Befehl	Beispiel	Beschreibung																																				
.PCB	.PCB	Format Header																																				
.COM	.COM	Kennzeichen Netzliste																																				
.REM	.REM VCC_NET	Einleitung Netzeintrag																																				
.REM <netname> <DEV> <Pin>	.REM VCC_NET U1 1	Bauelementeeintrag																																				
.EOD		Dateiende																																				
CC-/CCZ-Format	Das CC-/CCZ-Format kann CAD-Daten aus Mentor CamCad XML-Files (*.CC) verarbeiten, die optional als *.ZIP-File vorliegen können (*.CCZ).	<pre><?xml version="1.0" encoding="ISO-8859-1"?> <CCDoc version="6.0"> <Keywords> <Keyword index="0" valueType="1" group="1" CCKeyword="NAME" inherit="1"/> <Keyword index="1" valueType="1" group="1" CCKeyword="NETNAME" inherit="1"/> <Keyword index="2" valueType="1" group="1" CCKeyword="REFNAME" inherit="1"/> ... <Keyword index="4" valueType="3" group="1" CCKeyword="PINNR" inherit="1"/></pre>																																				



Format	Beschreibung	Syntaxbeispiel
		<pre> ... <Keyword index="7" valueType="1" group="1" CCKeyword="\$DEVICE\$\$" inherit="1"/> <Keyword index="8" valueType="1" group="1" CCKeyword="VALUE" inherit="1"/> ... <Keyword index="12" valueType="2" group="1" CCKeyword="VOLTAGE" inherit="1"/> <Keyword index="13" valueType="1" group="1" CCKeyword="PARTNUMBER" inherit="1"/> <Keyword index="14" valueType="1" group="1" CCKeyword="DEVICETYPE" inherit="1"/> ... <Keyword index="21" valueType="4" group="1" CCKeyword="POWERNET" inherit="1"/> ... <Keyword index="44" valueType="1" group="8" CCKeyword="\$PACKAGE\$\$" inherit="1"/> </pre>
CLINK_DIF-Format		<pre> { JOB BOARD1 { PCB BOARD1 { ENVIRONMENT { SOURCE "CADES-G 3.94" } { VERSION 4.0 } { DATE 23-04-99} { TIME 18:08:07} { UNITS 1/1000 ZOLL } { LAYER 6 } { TOP_LAYER 1 } { BOTTOM_LAYER 6 } { NO_NET 232 } { NO_COMP 636} } </pre>



Format	Beschreibung	Syntaxbeispiel																								
		<pre> { BOARD ... {COMP -- BAUELEMENTEINFORMATION {COMP_DEF {NAME D301} {PART_NR 3BHC120035R0004 } } {PIN_DEF -- NETZINFORMATION {PIN 1 {NET BLA20} {ICT 667}} {PIN 2 {NET BLA18} {ICT 683}} {PIN 3 {NET BLA16} {ICT 642}} } </pre> <p>Die Einträge sind nach Bauelementen und Pins sortiert.</p>																								
EAGLE_VER_2_2-Format	Für den Import des Formats EAGLE_VER_2_2 sind zwei Dateien erforderlich. Die Bauelementeliste hat die Erweiterung *.XLS (EXCEL) und die Netzliste die Erweiterung *.NETS. Die Erweiterungen sind frei wählbar, sollen aber hier genannt sein, da oftmals an ihnen das Format erkannt wird.	<p>Beispiel:</p> <pre> NET 'NET1' C1-1' 'IC2-2' NET 'NET2' U10-1' 'U9-2' </pre>																								
EAGLE_VER_4_0-Format	Für den Netzlisten-CAD-Import sind zwei Dateien notwendig. Die Datei für die Bauelementeliste hat defaultmäßig die Erweiterung *.PART. Die Netzlistendatei hat die Erweiterung *.NET.	<p>Die Syntax der Bauelementeliste soll am folgenden Beispiel aufgezeigt werden:</p> <p>Partlist Exported from example.sch at 29.09.2003 11:02:06 EAGLE Version 4.03 Copyright (c) 1988-2001 CadSoft</p> <table border="1"> <thead> <tr> <th>Part</th> <th>Value</th> <th>Device</th> <th>Package</th> <th>Library</th> <th>Sheet</th> </tr> </thead> <tbody> <tr> <td>J363</td> <td>CONON51BSC</td> <td>CONON51BSC</td> <td>CONON_BSC_ADAPTIERUNG_STECKER</td> <td>1</td> <td></td> </tr> <tr> <td>J376</td> <td>ML20</td> <td>ML20</td> <td>con-harting-ml</td> <td>1</td> <td></td> </tr> <tr> <td>J377</td> <td>ML20</td> <td>ML20</td> <td>con-harting-ml</td> <td>1</td> <td></td> </tr> </tbody> </table>	Part	Value	Device	Package	Library	Sheet	J363	CONON51BSC	CONON51BSC	CONON_BSC_ADAPTIERUNG_STECKER	1		J376	ML20	ML20	con-harting-ml	1		J377	ML20	ML20	con-harting-ml	1	
Part	Value	Device	Package	Library	Sheet																					
J363	CONON51BSC	CONON51BSC	CONON_BSC_ADAPTIERUNG_STECKER	1																						
J376	ML20	ML20	con-harting-ml	1																						
J377	ML20	ML20	con-harting-ml	1																						



Format	Beschreibung	Syntaxbeispiel
		<p>Aus der Spalte Part wird der Bauelementebezeichner gelesen. Aus der Value-Spalte wird der Bauelementetyp abgeleitet. Wenn diese leer ist, wird der Typ aus der Spalte Device gelesen. Aus der Spalte Package wird das zugehörige Package ins *.DIF-File übernommen.</p> <p>Die Netzliste hat eine einfache Struktur:</p> <p>Beispiel: Netlist Exported from example.sch at 29.09.2003 11:01:55 EAGLE Version 4.03 Copyright (c) 1988-2001 CadSoft Net Part Pad Pin Sheet</p> <p>Change Class 0</p> <pre> _1 J363 1 1 1 J377 3 3 1 </pre>
EDIF-Formate		<p>Die EDIF-Files sind leicht an ihrem Header zu erkennen. Er hat folgende Form:</p> <pre> (EDIF UC TPE2000 -- EDIF HEADER (EDIFVERSION 2 0 0) -- VERSION (EDIFLEVEL 0) -- DAS LEVEL MUSS 0 SEIN (KEYWORDMAP (KEYWORDLEVEL 0)) -- MUSS AUCH 0 SEIN (STATUS (WRITTEN (TIMESTAMP 2000 03 14 13 35 21) (PROGRAM "CAPTURE.EXE" (VERSION "9.10.157")) (COMMENT "ORIGINAL DATA FROM ORCAD/CAPTURE SCHEMA")) </pre>
FABMASTER FATF-Format	Das FABMASTER FATF-Format besteht aus einer CAD-Datei mit der Endung	; FABMASTER FATF Beispiel :FABMASTER FATF REV 11.2;



Format	Beschreibung	Syntaxbeispiel
	*.ASC. Es handelt sich hierbei um ein Fabmaster Output Format.	<pre> ... :PARTS 1,"R209","35111000", "0603R_REFLOW",-643700,266900,1800,B; 2,"R521","35111000", "0603R_REFLOW",-105000,244400,2700,B; ... 1646,"D203","35190697", "SO-8_LP1270_CBW3810_REFLOW",-605000,242500,0,B; ... :EOD ... :NETS ... ; 5. net GND: 5,"GND",S, ((2116,2),(2120,2),(2121,2),(2122,2),(2118,2),(2117,2), (2119,2),(2123,2),(2134,5),(2152,4),(1645,5),(2132,6),(2132,29),(1643,5), ... (1638,2),(1637,2),(1636,2),(1635,2)); ... :EOD </pre>
FABMASTER_NTR-Format	Das FABMASTER_NTR ASCII-Dateiformat benötigt zum Import der Bauelementeliste und der Netzliste in SYSTEM CASCON™/ CASCON POLARIS™ zwei Files (Default-Erweiterung *.ASC).	<pre> ... PART X Y ROT GRID T/B 'DEVICE', 'OUTLINE' IC1 3.870 2.831 180.0 N14 (T) '228015190', 'SO20' IC2 2.099 2.389 180.0 L14 (T) '228028960', 'SO8_LARGE' IC3 2.536 2.395 180.0 M14 (T) '226497760', 'SO14' IC4 0.851 2.269 90.0 K14 (T) '228011780', 'SO52' IC5 3.971 2.168 180.0 N14 (T) '225463212', 'SO28_LARGE' ... </pre> <p>Die Einträge unter PART und DEVICE werden eingelesen.</p> <p>Die <u>Netzlistendatei</u> hat folgende Struktur: Header: Der Header besitzt die gleiche Form wie die Bauelementeliste.</p>



Format	Beschreibung	Syntaxbeispiel
		<p><i>Netzliste:</i> Die Netzliste hat folgende Syntax:</p> <p>Beispiel:</p> <pre>#1 (S) P3V3 -- DATENSATZ NETZNAME X.94 -- PIN 94 ZU BAUELEMENT X X.104 ... #2 (S) GND X.93 C71.1 ...</pre>
GENCAD_14-Format	Für den Import des GENCAD-Formats wird eine Datei benötigt. Das Format wurde in der Version 1.4 implementiert. Es besteht aus einem Header, einem Bauelemente- und einem Netzlistenabschnitt.	<p>Der Header wird überlesen und hat folgende Form:</p> <pre>\$HEADER GENCAD 1.4 USER TRANSLATOR GENCAD OUTPUT V:18 DRAWING C:\TEMP\ REVISION ... ATTRIBUTE HEADER "... " UNITS USER 1000 ORIGIN 0 0 INTERTRACK 0 \$ENDHEADER</pre> <p>Die Bauelementeliste folgt dem Header. Aus dem DEVICE-String werden die Bauelementenamen gebildet und aus dem TYPE-String oder SHAPE-String der Typ. Falls ein Unterstrich vorhanden ist, wird dieser und alles Weitere ignoriert.</p> <pre>\$DEVICES ... DEVICE C1_XXX_YYY ... TYPE CAP DEVICE U1_XXX_YYY</pre>



Format	Beschreibung	Syntaxbeispiel
		<pre> ... TYPE EPF10K10TC144 ... DEVICE U2 ... SHAPE PLCC52 ... \$ENDEVICES Aus dem Bauelementenamen C1_XXX_YYY wird C1 gebildet. Der Typstring wird übernommen. Die Netzliste hat folgende Form: \$SIGNALS SIGNAL AD0 NODE C1 1 NODE U1 1 ... SIGNAL AX NODE C1 2 ... \$ENDSIGNALS Für jedes Pin gibt es einen NODE-Eintrag in der Netzbeschreibung.</pre>
GENCAD2_14-Format	Dieses Format entspricht im Wesentlichen dem GENCAD_14-Format.	<p><u>Unterschiede:</u></p> <p>Der Bauelementeabschnitt wird mit \$COMPONENTS eingeleitet und mit \$ENDCOMPONENTS abgeschlossen. Das Schlüsselwort für den Eintrag eines Bauelements heißt COMPONENT. Das Schlüsselwort für den Eintrag eines Bauelementetyps heißt DEVICE. Wenn Bauelementebezeichner und Typbezeichner gleich sind, wird der Token hinter SHAPE als Typname verwendet. Dieses Format nutzt die CAD Format Specifications (siehe Advanced Options des CAD-Imports für ein Board).</p>
HP Board File Format	Die HP Board File-Netzliste besteht aus einer Datei	<p>Im Beispiel wird für den Kondensator C1 der Bauelementetyp ELECT_260_10UF importiert. Im *.DIF -File sieht der Eintrag für C1 wie folgt aus:</p> <pre>(DEV C1</pre>



Format	Beschreibung	Syntaxbeispiel
		(TYPE 'ELECT_260_10UF'))
IPC2581-Format	Dieses Format stellt eine Weiterentwicklung des ODB++(X) Formats durch Vereinigung mit dem GenCAM (IPC-2511) Format zum Standard IPC-2581 durch das gleichnamige Consortium dar	<p>Das Format kann in zwei Varianten auftreten: es gibt den ursprünglichen Standard von 2004, sowie eine Neufassung von 2007 (bei dem einige Listen-Tags eingespart wurden). Beide Varianten werden mit dem gleichen CASCON-Importfilter unterstützt.</p> <p>Beim CASCON-CAD-Import für IPC2581 ist nur die Eingabe des Pfadnamens der Input-Datei (XML-Datei mit Default-Erweiterung *.cvg) erforderlich.</p> <p>Fragment eines Beispiel-Quellfiles mit den für CASCON relevanten XML-Tags (Std. 2004):</p> <pre><?xml version="1.0" encoding="UTF-8"?> <IPC-2581 xsi:schemaLocation="http://webstds.ipc.org/2581 2581Schema.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://webstds.ipc.org/2581"> <Content> ... other <StepListRef>bd-sample</StepListRef> <!-- Std. 2007: <StepRef> --> ... more StepListRef possible <LayerDescRef>bd-sample</LayerDescRef> <!-- Std. 2007: list of <LayerRef> --> ... other </Content> ... other <Ecad name="bd-sample:bd-sample"> <CadHeader units="MILLIMETER"/> <CadData> <LayerDesc name="bd-sample:bd-sample"> <!-- Std. 2007: tag removed --> <Layer name="A-Component" layerFunction="COMPONENT" polarity="POSITIVE" context="BOARD" side="TOP"/> ... more Layer </LayerDesc> ... other <StepList name="bd-sample:bd-sample"> <!-- Std. 2007: tag removed --> <Step name="bd-sample"> ... other <Package height="7.1" name="bd-sample:SW-IMD3-15.23*13.6"</pre>



Format	Beschreibung	Syntaxbeispiel
MENTOR_MIF-Format	Für den Import des MENTOR_MIF-Formats ist nur eine Datei notwendig.	<pre> type="PLASTIC_DIP"> ... more Package <Component packageRef="bd-sample:IC-S8-225-i-F{B}" layerRef="B-Component" refDes="bd-sample:IC18" part="LM218D" ...> <!-- ??? --> <Attribute name = "PART_NAME" value = "UNDISCLOSED"/> <!-- optional --> <NonstandardAttribute name="chgFootComp" type="STRING" value="YES"/> <!-- optional --> ... <Component part="MCR03_10K" refDes="routed:R600" layerRef="A-Component" packageRef="routed:1608chip" ...> <NonstandardAttribute name="ADM_PKG_TYPE" type="STRING" value="SMD,R-SQ-1608"/> <NonstandardAttribute name="ADM_COMP_PLACED_LAYER" type="STRING" value="1"/> more Component <LogicalNet name="bd-sample:RESET" netClass="SIGNAL"> <LogicalNetPin pin="13" componentRef="bd-sample:IC11"/> <LogicalNetPin pin="1" componentRef="bd-sample:R32"/> <LogicalNetPin pin="1" componentRef="bd-sample:CN5"/> </LogicalNet> ... more LogicalNet <LayerFeature> ... more LayerFeature </Step> </StepList> </CadData> </Ecad> ... other </IPC-2581> </pre>
Der Bauelementeabschnitt hat folgende Syntax:		



Format	Beschreibung	Syntaxbeispiel
	Sie hat defaultmäßig die Erweiterung *.MIF.	<pre> %INSTANCE REF:"R56" -- BAUELEMENT COMP:"RES" -- TYP DES BAUELEMENTS A (P_MA6_SEL) PIN_NO:"1" -- PINBEZEICHNER PINTYPE:"IXO", B (T_MA(6)) -- NETZBEZEICHNER PIN_NO:"2" PINTYPE:"IXO"; %ENDINSTANCE </pre> <p>Die Netze werden im Netzlistenteil hintereinander wie folgt aufgeführt:</p> <pre> %NET P_MA6_SEL T_MA(6), %ENDNET </pre>
NEUTRAL-Format	Das NEUTRAL-Format speichert alle Informationen, die SYSTEM CASCON™/ CASCON POLARIS™ benötigt, in einer Datei, die die defaultmäßige Erweiterung *.NTR hat. Kommentare werden (zeilenweise) hinter dem „#“-Zeichen (Raute) erwartet.	<pre> Beispiel: ##### #BAUELEMENTINFORMATION ##### ... COMP C1002 ELKO_D 33U/16V SMD_TAKOD 93.345 217.805 1 90 ... COMP U1 MACH211-7JC-SOCKEL MACH211-7JC-SOCKEL PLCC_PLCC_44 190.5 111.76 1 270 ... </pre> <p>Das Bauelement C1002 wird mit dem Typ ELKO_D importiert. Das Bauelement U1 wird mit dem Typ MACH211-7JC-SOCKEL importiert.</p> <p>Die Netzinformation wird wie folgt im NEUTRAL-Format gespeichert:</p>



Format	Beschreibung	Syntaxbeispiel
		<pre>##### #NETZINFORMATION ##### NET GND N_PROP (NET_TYPE,"POWER") N_PIN C1-2 135.89 25.7175 TAKOD_PAD2 1 N_PIN C2-2 248.92 121.6025 TAKOD_PAD2 1 N_PIN C3-2 243.84 121.6025 TAKOD_PAD2 1 ...</pre> <p>Die Bauelementliste befindet sich in der Datei hinter der Netzliste.</p>
Format ODB++	<p>Das ODB++ Format von Valor ist eine ASCII-orientierte Datenbank für EDA-Aufgaben. Ab CASCON Version 4.01 unterstützt CASCON dieses Format für den CAD-Import. Input-Files sind dabei</p> <ul style="list-style-type: none"> <input type="checkbox"/> das Components-File und <input type="checkbox"/> das Netzlistenfile von ODB++. <p>Das Components-File hat immer den Namen <i>components.z</i> und liegt i. Allg. in gezippter Form vor (Standard-UNIX-Format). In den ODB++ Tools kann das Komprimieren der Components-Daten unterbunden werden, was für den Import in CASCON unbedingt erforderlich ist.</p>	<p>Beispiel:</p> <pre># CMP 0 CMP 13 -0.04 1.22 270.0 N B70 2248827-0001 ;1,2,4=11,5=0.100000,6=0.035273 PRP REFLOC 'IN,0.2,-0.225,270,CC,0.035,0.035,0.009,STD,1' TOP 0 -0.198 1.62 270.0 N 223 0 1 1 TOP 1 0.118 1.62 270.0 N 223 1 1 2 TOP 2 -0.04 1.22 270.0 N 466 0 2 3 # ...</pre> <p>Im Beispiel ist der Name des Bauelements B70, der Typbezeichner ist 2248827-0001. Die TOP-Zeilen enthalten die Pininformation des Bauelements. Hier sind Pins des Bauelements von 1 bis 3 vorhanden. Die Zahlen 223 und 466 stellen die Netzindizes als Referenz zur Netzlistendatei dar.</p> <p>Die Netzliste trägt den Namen <i><ODB++Ordner>\steps\pcb\eda\data</i> meistens ohne Erweiterung. Sie liegt immer in ASCII-Form vor und enthält eine lineare Aufzählung der verwendeten Netze.</p> <p>Beispiel:</p> <pre>... #NET 0 NET &1N1096</pre>



Format	Beschreibung	Syntaxbeispiel
		...
		#NET 1 NET &1N1526
		...
		#NET 2 NET &1N289
		...
		#NET 3 NET &1N289
		...
		#NET 4 NET &1N338
		...
		#NET 5 NET &1N338
		...
		#NET 6 NET \$NONE\$
		...
		Es ist möglich, dass die Bauelementeinformationen in zwei Dateien untergebracht ist, die den gleichen Namen besitzen und in verschiedenen Verzeichnissen liegen, wie z.B.:
		<Project>\comp+_bot\components <Project>\comp+_top\components
		Beim ODB++ CAD-Import erscheint ein Dateiauswahldialog, in dem der Nutzer eine zweite Bauelementedatei angeben kann. Die Auswahl der Datei ist optional.



Format	Beschreibung	Syntaxbeispiel
Format ODB++ pro	<p>Das ODB++ Format (proprietäres Format ursprünglich von der Fa. Valor, seit 2010 von Mentor Graphics übernommen, siehe http://en.wikipedia.org/wiki/ODB%2B%2B) ist eine ASCII-orientierte Datenbank für EDA-Aufgaben in Form einer hierarchischen Dateiverzeichnis-Struktur.</p> <p>Für den Datentransfer ist es üblich, dass diese Filestruktur in Form eines einzelnen komprimierten Archiv-Datenfiles (*.tar, *.tgz oder *.zip) verwendet wird. Der Filename des Archivfiles (Basisverzeichnis der Filestruktur) wird im ODB++ Umfeld als „job_name“ bezeichnet.</p>	<p>unterstützt dieses Format für den CAD-Import seit der Version 4.01. Ab Version 4.6.3 ist der direkte Import aus den Archivfiles möglich, dazu wird das externe Tool <i>7za.exe</i> benutzt (Kommandozeilenversion des freien Datenkompressionsprogramms „7-Zip“, http://www.7-zip.org).</p> <p>Für CASCON relevante Input-Files in der Filestruktur sind dabei</p> <ul style="list-style-type: none"> <input type="checkbox"/> das allgemeingültige Jobverwaltungsfile (Informationen über den Inhalt der Verzeichnisstruktur) <code><job_name>/matrix/matrix</code> <input type="checkbox"/> Referenzlisten für Netze, Packages usw. in der Datei <code><job_name>/steps/<step_name>/eda/data</code> <input type="checkbox"/> die Zuordnungen zwischen Bauelementen, Netzen und Packages in den Components-Files <code><job_name>/steps/<step_name>/layers/comp+_bot/components</code> und <code><job_name>/steps/<step_name>/layers/comp+_top/components</code> <p>Das Archiv kann mehrere Arbeitsstände (oder auch Boards) als sogenannte „Steps“ enthalten, deren Daten dann in dem Verzeichnis <i>step_name</i> (z.B. „pcb“) liegen. In diesem Fall muss der Name des Steps, der importiert werden soll, in den CAD-Import-Optionen angegeben werden. Ohne Angabe wird der zuerst gefundene Step importiert und eine entsprechende Warnung im Report angezeigt. Gegebenenfalls muss dann der CAD-Import nach Angabe des richtigen Steps wiederholt werden.</p> <p>Die Daten-Files sind meist in gezippter Form (z.B. <i>components.Z</i>) im Archiv enthalten und werden automatisch dekomprimiert. In neueren ODB++ Versionen können noch Dateien <i>components2</i> bzw. <i>components3</i> vorhanden sein, die Bauelementeinformationen für Stücklisten enthalten. Diese werden beim Import ebenfalls berücksichtigt.</p> <p>Beim CASCON-CAD-Import für ODB++ ist nur die Eingabe eines einzigen Inputfiles erforderlich. Das ist entweder der Pfadname des gepackten Archivfiles oder (wenn die Filestruktur schon entpackt ist) der Pfadname einer der obigen Dateien. Weitere benötigte Files werden dann automatisch im Verzeichnis gesucht. Wird ein Archivfile verwendet, entsteht nach dem ersten Importversuch immer die entsprechende Filestruktur mit den verwendeten Inputfiles im CASCON UUT-Projektordner, damit im Fehlerfall über Filereferenzen im Report die Verursacherstelle angezeigt werden kann.</p> <p>Die Referenzlistendatei, z.B. <code><job>\steps\pcb\eda\data</code>, ist ein ASCII-File und enthält u.a. eine lineare Aufzählung der verwendeten Netze und Packages. Die Zuordnung zwischen Components und Netzen erfolgt über die hier definierten Referenznummern:</p> <pre> ... #NET 0 NET \$NONE\$ </pre>



Format	Beschreibung	Syntaxbeispiel
		<pre> SNT TOP B 5 0 ... #NET 1 NET DGND SNT TOP T 72 82 ... # PKG 0 PKG 0603C 0.066929 -0.019685 -0.0216535 0.086614 0.0216536 ... PIN 1 S -0.037401574803 0.000000000000 0 U U ... PIN 2 S 0.037401574803 0.000000000000 0 U U ... # PKG 1 PKG CON36_A 0.1 -0.2799213 -0.07559055 2.180709 0.2362205 ... #... Die Components-Files bestehen aus einer Folge von Records, die der folgenden Beschreibung genügen: ... # CMP 0 CMP 7 35.77 21.98 180 N R101 12.1K ;0=1,1=0.0000 PRP TYPE_NAME 'RESISTOR' TOP 0 35.77 21.98 0 N 141 0 R101-1 TOP 1 35.85 21.98 0 N 143 0 R101-2 # # CMP 1 CMP 3 29.1365351 22.3499996 180 N R162 100 ;0=1,1=0.0000 TOP 0 29.1365351 22.3499996 0 N 162 0 R162-1 TOP 1 29.2034652 22.3499996 0 N 126 0 R162-2 # # BOM DATA CPN 10020344 </pre>



Format	Beschreibung	Syntaxbeispiel
		PKG IPN 10020344 ... Im Beispiel ist der Name des ersten Bauelements R101 , der Typebezeichner ist 12.1K , das Package hat die Referenznummer 7 . Mit den PRP-Definitionen können zusätzliche Eigenschaften gesetzt werden, z.B. der Typ- oder Package-Name überschrieben werden. Die Verwendung der Properties ist in den CAD-Import-Optionen anzugeben. Die TOP (Toeprints) beschreiben die Pins des Bauelements und deren Netzverbindung. Hier sind die Pinnamen von 1 und 2 sowie deren Anschluss an die Netze mit den Referenznummern 141 und 143 definiert (eine eventuelle -1 entspricht nicht angeschlossenen Pins). Die Zahlen 223 und 466 stellen die Netzindizes als Referenz zur Netzlistendatei dar. In neueren ODB++ Versionen ist es möglich, dass BOM- (Bill Of Material) Attribute zu den Bauelementen angefügt werden (siehe Zeilen nach # BOM DATA). Diese können optional für die Typnamen in der generierten Netzliste benutzt werden, indem dafür die nach einem der Schlüsselwörter MPN, CPN, PKG, IPN folgenden Nummern oder Bezeichner zugewiesen werden (Details siehe Tabelle unten, Option Use BOM Data). Es können nur entweder Properties (PRP) oder BOM benutzt werden. Eine ausführliche ODB++ Formatbeschreibung als PDF-Datei erhält man nach Registrierung bei der ODB++ Solution Alliance.
Format ODB++ XML	Das Format ODB-XML oder ODB++(X) entspricht inhaltlich dem ODB++, aber die Datenbank ist in einem einzelnen XML-File enthalten. Die Entwicklung des Formats bei der Fa. Valor erfolgte ab dem Jahr 2000 und endete 2008 (Quelle: http://en.wikipedia.org/wiki/ODB%2B%2B). Parallel dazu wurde das Format in den Standard-IPC2581 integriert und weiterentwickelt, siehe Abschnitt 3.17 IPC2581 Format.	Beim CASCON-CAD-Import für ODB-XML ist nur die Eingabe des Pfadnamens der Input-XML-Datei erforderlich. Fragment eines Beispiel-Quellfiles mit den für CASCON relevanten XML-Tags: <pre> <?xml version = "1.0" encoding = "ISO-8859-1"?> <ODX> <ODX_CAD> <ODX_HEADER ... > ... </ODX_HEADER> <CAD_DATA> <CAD_DATA_HEADER ... /> <CAD_LAYER_LIST> </pre>



Format	Beschreibung	Syntaxbeispiel
		<pre> <CAD_LAYER NAME = "comp+_top" ... /> ... more CAD_LAYER <CAD_STEP_LIST> <CAD_STEP NAME = "pcb"> ... other tags <CAD_PACKAGE_LIST> <CAD_PACKAGE NAME = "to92 T "> ... <CAD_PIN_LIST> <CAD_PIN NUMBER = "1" NAME = "1" TYPE = "THRU" ETYPE = "U" MTYPE = "U"> ... more CAD_PIN ... more CAD_PACKAGE </CAD_PACKAGE_LIST> <CAD_COMPONENT_LIST> <CAD_COMPONENT NAME = "c2" PACKAGE = "cap-07200250 0508 T " PART = "??" ... > <CAD_PRP_LIST> <CAD_PRP NAME = "PART_NAME" TEXT = "UNDISCLOSED"/> ... more CAD_PRP ... more CAD_COMPONENT </CAD_COMPONENT_LIST> <CAD_LNET_LIST> <CAD_LNET NAME = "_210"> <CAD_LNP_LIST> <CAD_LNP C = "mm5387" P = "34" T = "34"/> <CAD_LNP C = "con5" P = "14" T = "14"/> </CAD_LNP_LIST> </CAD_LNET> ... more CAD_LNET </CAD_LNET_LIST> ... other tags </CAD_STEP> ... more CAD_STEP possible </pre>



Format	Beschreibung	Syntaxbeispiel
		<pre data-bbox="862 229 1108 459"> </CAD_STEP_LIST> <CAD_FILE_LIST> ... </CAD_FILE_LIST> </CAD_DATA> </ODX_CAD> </ODX> </pre> <p data-bbox="862 501 2072 831"> Wie beim ODB++ können mehrere Steps in der CAD_STEP_LIST mit dem Tag CAD_STEP mit entsprechendem NAME, z.B. „pcb“, vorhanden sein. In diesem Fall muss der zu verwendende Step in den Optionen angegeben werden, ansonsten wird der zuerst gefundene Step benutzt (eventuell mit Warnung im Report). Die Package-Namen werden in der CAD_PACKAGE_LIST und die Netznamen in der CAD_LNET_LIST definiert. Die Zuordnung zu Bauelementen und Pins erfolgt hier direkt über die Namen (nicht wie bei ODB++ über Referenznummern). Die einzelnen Anschlusspunkte eines Netzes in der CAD_LNP_LIST sind durch Bezug auf Bauelemente-Name C, Pin-Nummer P und Toeprint-Name T realisiert. Die CAD_COMPONENT_LIST definiert die verwendeten Bauelemente mit Name, Package und Type (PART). Alternativ können Package und Type auch über Eigenschaften in der CAD_PRP_LIST überschrieben werden, wenn das in den CAD-Import-Optionen angegeben wird. </p>
ORCAD_NETLIST-Format	Das ORCAD_NETLIST-Format wird vom OrCad Capture_9 CAD-System erzeugt und benötigt für den CAD-Import in die CASCON GALAXY®-Software eine Datei, die defaultmäßig die Erweiterung *.ONL hat.	<p data-bbox="862 868 1794 900">Die Syntax entspricht einer wohldefinierten Klammersyntax, die mit dem Schlüsselwort</p> <pre data-bbox="862 932 1086 1034"> (PCB <ProjektName> ...) </pre> <p data-bbox="862 1075 1032 1107">eingeleitet wird.</p> <p data-bbox="862 1139 1265 1171">Ein Bauelementeintrag hat die Form</p> <pre data-bbox="862 1203 1256 1362"> (COMPONENT <TypBezeichner> (PLACE <BauelementeBezeichner> ...) ... </pre>



Format	Beschreibung	Syntaxbeispiel
		<p>)</p> <p>wobei <TypBezeichner> den Bauelementetyp und <BauelementeBezeichner> den Namen des Bauelements repräsentieren.</p> <p>Ein Netzeintrag hat die Form</p> <p>(NETWORK</p> <p>(NET <NetzName></p> <p>(PINS</p> <p><BauelementeBezeichner>-<PinBezeichner></p> <p><BauelementeBezeichner>-<PinBezeichner></p> <p>...</p> <p>)</p> <p>)</p> <p>...</p> <p>)</p>
ORCAD_3-Format	Das ORCAD_3 Format wird aus einer Netzlistendatei mit Default-Erweiterung *.NET importiert.	<p>Beispiel:</p> <p>NETLIST,2</p> <p>(DRAWING,ORCAD.NET,1-1</p> <p>...</p> <p>(SYM 1,-1, 1784</p> <p>(DATA ,2, U10 -- BAUELEMENTEBEZEICHNER</p> <p>...</p> <p>(DATA ,3, 74AHC573 -- BAUELEMENTETYP</p> <p>...</p> <p>)</p> <p>Die Netzinformation mit den Pins sieht wie folgt aus:</p> <p>Beispiel:</p>



Format	Beschreibung	Syntaxbeispiel
		<pre> NETLIST,2 (DRAWING,ORCAD.NET,1-1 ... (SIG,,N03064,1-1,5,N03064 ... PIN,1-1,917,R82,23,1 PIN,1-1,1419,TP152,23,1 PIN,1-1,1884,U9,23,1 PIN,1-1,1784,U10,23,1 ...) </pre>
OrCAD Wire List Format	<p>Das OrCAD Wire List Format steht ab der CASCON-Version 4.1.1 zur Verfügung.</p> <p>Für den Import des OrCAD Wire List Formats wird eine Datei mit der Erweiterung <i>*.NET</i> benötigt, die die Bauelementeinformation und die Netzlisteninformation enthält.</p>	<pre> Wire List ... <<< Component List >>> ... 01uFC0603 BC1 C0603 CY7C1350BTQFP100 U39 TQFP100 MT48LC2M32B2TG-7TSOP-86 U4 TSOP-86 ... <<< Wire List >>> NODE REFERENCE PIN # PIN NAME PIN TYPE PART VALUE [00001] N51410 U29 W5 BR_ BiDirectional MPC8265TBGA480 R119 2 2 Passive 220603 ... [00090] DAT46 U29 B6 D46 BiDirectional MPC8265TBGA480 RS17 2 2 Passive 10K10P-8R U4 33 DQ17 BiDirectional MT48LC2M32B2TG-7TSOP-86 P3 35 E1 Passive MICTOR38MICTOR38 </pre> <p>Der Bauelementebezeichner wird aus der zweiten Spalte und der Typbezeichner aus der ersten Spalte der Bauelementeliste gelesen.</p>



Format	Beschreibung	Syntaxbeispiel
		<p>Der Pinbezeichner steht in der zweiten Spalte der Pinliste des jeweiligen Netzes hinter dem Bauelementebezeichner des Pins.</p>
<p>PADS_POWER-Format</p>	<p>Das PADS_POWER-Format kann mit Dateien der Versionen 2.0 und 3.0 importiert werden. Für den Import der Bauelementeliste und der Netzliste wird eine Datei benötigt.</p>	<p>*REMARK* PADS-POWER VERSIONEN 2.0 UND 3.0</p> <p>*PART* ITEMS</p> <pre> IC6 PENTIUM/SPGA@UG296X1800Y1800A R82 R1206@R02X122Y0A0 R80 R1206@R02X122Y0A0 ST14 STA30BA@ST30S28E2A28 C75 C1206@C02X122Y0A0 C58 B32529-A@C02S200E0L28 ST15 STA30BA@ST30S28E2A28 C44 C1206@C02X122Y0A0 </pre> <p>...</p> <p>*NET*</p> <p>Die Typbezeichner der Bauelemente werden bis zum @-Zeichen importiert, d.h. aus der Zeile</p> <pre>C75 C1206@C02X122Y0A0</pre> <p>wird</p> <pre>DEV C75 (TYPE C1206))</pre> <p>in das Net and Component List File <i>*.DIF</i> importiert.</p> <p>An die Bauelementeliste schließt sich die Netzliste an:</p> <p>Beispiel:</p> <pre> *SIGNAL* HA20M3V# R77.2 IC6.AK8 ST14.28 *SIGNAL* HDPA2 </pre>



Format	Beschreibung	Syntaxbeispiel						
		IC6.C25 R104.2 *SIGNAL* HHITM# IC6.AL5 IC7.T8 *SIGNAL* HDP A3 IC6.D18 R92.2 *END* -- DATEIENDE						
P_CAD-Format	Um das P_CAD-Format zu importieren, sind zwei Dateien erforderlich. Die folgende Tabelle gibt die Default-Erweiterungen an:	Im folgenden Beispiel wird die Syntax der Bauelementliste erläutert: Beispiel: <pre> %***** % % PROGRAM : PC-FORM VERSION 6.00 * % DATE : MAR 16 1993 * % TIME : 01:50:36 PM * % FILE IN : SS10TEST.XNL * % FILE OUT : SS10TEST.MAT * % FORMAT : P-CAD MATERIALS LIST * % %***** </pre> <table border="1"> <thead> <tr> <th>Datentyp</th> <th>Dateierweiterung</th> </tr> </thead> <tbody> <tr> <td>Bauelementedaten</td> <td>*.MAT</td> </tr> <tr> <td>Netzl listedaten</td> <td>*.WRL</td> </tr> </tbody> </table> <pre> ITEM QTY COMP-NAME REFERENCE-DESIGNATOR DESCRIPTION ----- 1 72 W1K R0105 R0106 R0103 R0102 R0104 R0107 2 82 W10K R0101 R1602 R1603 R1605 R1606 R1616 R1615 R1614 R1610 R1609 R1608 R1607 3 4 MBUS100 J0102 J0101 MBUS-J1 MBUS-J2 4 2 W619%1 R1601 R1604 5 12 W2K2 R0108 R0801 R1-5 R1-4 </pre>	Datentyp	Dateierweiterung	Bauelementedaten	*.MAT	Netzl listedaten	*.WRL
Datentyp	Dateierweiterung							
Bauelementedaten	*.MAT							
Netzl listedaten	*.WRL							



Format	Beschreibung	Syntaxbeispiel
		<pre> R1-7 R1-8 R1-9 R2-5 R3-5 R4-5 R5-5 R24-2 6 1 L64862 U0101 7 1 L64860 U0201 </pre> <p>Die Liste der Bauelemente gibt den Index, danach die Stückzahl, dann den Bauelementetyp und zum Abschluss die Liste der Bauelemente an, die unter diesem Typ im Projekt vorhanden sind.</p> <p>Die Datei der Netzlisteninformation hat im P_CAD-Format folgendes Aussehen:</p> <p>Beispiel:</p> <pre> % NODE REF. DESIGNATOR-PIN % ---- ----- % IRL11 J0101-93 U1002-17 TP20162-1 MBUS-J1-93 IC6-11-17 IC6-11-8 MAD59 J0102-86 J0101-86 U0101-H02 U0201-X16 MBUS-J1-86 </pre> <p>Unter dem Bezeichner NODE sind die Netznamen gespeichert. Es folgt eine Liste der Pins des jeweiligen Netzes.</p>
PCAD_2000-Format	Für den Import des Formats PCAD_2000 wird eine Datei benötigt. Defaultmäßig hat sie die Erweiterung <i>*.NET</i> .	<p>Die Syntax der Datei ist wie folgt aufgebaut:</p> <p>Beispiel:</p> <pre> ... (NETLIST </pre>



Format	Beschreibung	Syntaxbeispiel
		<pre> (COMPINST "U1" -- BAUELEMENT (ORIGINALNAME "EPM7128" -- TYP DES BAUELEMENTS) (COMPINST "U2" (ORIGINALNAME "EPM7128") ...)) ... (NET "NET1" (NODE "U1" "1") (NODE "U2" "1") ... (NET "GND" (NODE "U2" "2") -- BAUELEMENTEPIN) -- NETZE ...) -- ENDE DER NETZLISTE </pre>
PCAD_2001 Format	<p>Das PCAD_2001 CAD-Format des CAD-Systemherstellers Actell wird von SYSTEM CASCON™ ab Version 4.0.2 unterstützt.</p> <p>Für den Netzlistenimport wird eine Datei benötigt, die defaultmäßig die Erweiterung <i>*.NET</i> hat.</p>	<p>Beispiel:</p> <pre> {COMPONENT EXAMPLE.PCB {ENVIRONMENT BOARD.PCB} {DETAIL {SUBCOMP {I SS34 SMD.PRT D3 {CN 1 GND 2 NET16 } } {I TESTPAD SMD.PRT PLL.LOCK {CN 1 NET00406 </pre>



Format	Beschreibung	Syntaxbeispiel
		<pre> } } {I CY2308-2SMD.PRT IC1 {CN 1 NET00694 2 NET00697 3 NET00698 4 +3.3V 5 GND 6 NET00700 7 NET00701 8 NET00693 ... } } } } } } </pre> <p>Die Zeile {I enthält u.a. den Bauelementetyp, z.B. CY2308-2, und den Devicenamen, z.B. IC1. Ab der Anweisung {CN folgt eine Liste mit der Pinnummer und dem Netzbezeichner, bis eine schließende Klammer folgt. Der CAD-Filter trennt die Bezeichner am Leerzeichen, Tabulatorzeichen, am Zeilenende bzw. wenn die Syntaxelemente { oder } folgen.</p>
SPECIAL_5-Format	Das SPECIAL_5-Format erfordert nur eine Netzlistendatei, die defaultmäßig die Endung *.AUG besitzt. Es gibt CAD-Systeme, die auch eine NETS.TXT Datei erzeugen. Das Besondere an dieser Datei ist, dass die Netzlisteninformation vor der Bauelementeinformation erscheint. Für den Import spielt dies aber keine Rolle.	<p>Beispiel:</p> <pre> ;; V4.1.0 %NET %PAGE=0000 %PRIOR=0 -- NETZLISTENINFORMATION \ -5V_SNS\ \J1\ -\75\ \J1\ -\76\ \J2\ -\58\ \R241\ -\1\ \ -12V\ \C6\ -\2\ \C7\ -\2\ \C97\ -\2\ \E123\ -\1\ \E703\ -\1\ \J2\ -\73\ * \P1\ -\A31\ \R242\ -\1\ ... </pre>



Format	Beschreibung	Syntaxbeispiel
		<pre> BAUELEMENTEINFORMATION %PART \3432-D302\ * \J1\ \XC4025E,HQ304\ * \U7\ * \U8\ </pre>
SPECIAL_6-Format	Für den Import des SPECIAL_6-Formats sind zwei Dateien nötig:	<p>Für den Import des SPECIAL_6-Formats sind zwei Dateien nötig: Die erste Datei ist die „Bill of material“ und hat folgende Syntax:</p> <pre> # BOARD STATION BOM FILE # REF PART NO GEOMETRY DESCRIPTION B1 MCSO_10MHZ CLOCK-MCSO-LCC OSZILLATOR,OSZILLATOR, OSZILLATOR, 10MHZ, STD C1 5L.5224.032.40 CC0805 CAPACITOR, CAPACITOR,SMD, 27P, 5% </pre> <p>Der Index der Spalte des Bauelementetyps kann in den Optionen des CASCON CAD Imports vorgegeben werden.</p> <p>Die zweite Datei ist die Netzliste und hat folgende Syntax:</p> <pre> # BOARD STATION NETS FILE FORMAT 1.0 NET '/A(1)' U9-20 U10-20 U13-5 NET '/A(2)' U9-19 U10-19 U13-6 </pre>
SPECIAL_7-Format	Dieses Format benötigt zwei Dateien für den CAD-Daten Import.	<p>Beispiel:</p> <pre> 28 93N54503-S472-F500 R251..R255,R257..R259, B 28 93 N54503-S472-F500 R262...R265 B 29 4 N54503-S473-F500 R260,R261,R266,R267 B 30 2 N54503-S612-F900 R170,R176 B 31 1 N54503-S821-F500 R188 B 32 1 N65126-S80-M200 L1 T </pre>



Format	Beschreibung	Syntaxbeispiel
		<pre>33 1 N71201-S12-A1 D1000 T 34 2 N74206-S50-A1 TS1,TS4 T 35 6 N74206-S51-A1 TS1000...TS1005 T</pre> <p>Die Syntax der Netzliste ist wohldefiniert und lässt sich an einem Syntaxbeispiel am schnellsten erläutern:</p> <p>Beispiel:</p> <pre>NET '/N\$842' SZ4-3 P100-E53 NET '/N\$847' SZ6-3 P100-E55 NET '/CSIO_4' U1006-77 P02-14 R137-2 R99-1 (PCB_BA_NAME,"/CSIO_4") NET '/CSIO_5' U1006-76 P02-16 R91-1 R138-2 (PCB_BA_NAME,"/CSIO_5") NET '/CSIO_6' U1006-75 P02-18 R103-1 R140-2 (PCB_BA_NAME,"/CSIO_6") NET '/DSACK0' U1006-55 P05-25 U1-194 R100- (PCB_BA_NAME,"/DSACK0") NET '/DSACK1' U1006-56 P05-23 P03-18 R89-1 U1-196 (PCB_BA_NAME,"/DSACK1")</pre>
SPECIAL_10-Format	Dieses CAD-Format ist ein spezielles Format, welches für den Import nur eine CAD-Datei benötigt.	<p>Die Informationen sind spaltenweise angeordnet. Leider gibt es keine Schlüsselwörter, die das Format charakterisieren.</p> <p>Jedes Pin eines Bauelements wird einzeln aufgeführt. Es folgt ein Beispiel. Die Kopfzeile ist nicht Bestandteil der Datei, sondern dient nur der Beschreibung.</p> <pre>Nr. Netz Typnummer Bauelement Pin</pre> <p>Beispiel:</p> <pre>6 SA_A[12] 8325 U14 4 67 SA_A[12] 8325 U15 3 ...</pre>
SPECIAL_11-Format	Das SPECIAL_11-Format wird aus einer Datei mit der defaultmäßigen	Das SPECIAL_11-Format wird aus einer Datei mit der defaultmäßigen Erweiterung *.DAT importiert. In einer Zeile erscheinen die Netze, Bauelemente und Pins, wie das folgende Beispiel zeigt:



Format	Beschreibung	Syntaxbeispiel
	Erweiterung *.DAT importiert.	<pre> CONCISE NET LIST - 1 ... -- NETZ BAUELEMENT PINNUMMER TYP AB_ATD- D14 38 LVTH16374-B0617490,74LVTH16374B AB_ATD- D15 5 LVTH273-B0617471,74LVTH273,74LB ... END CONCISE NETLIST </pre>
SUPERMAX Input Wire List Format	Für den Import des Formats ist eine Datei erforderlich. Sehr oft besitzt sie die Erweiterung *.NET.	<p>Der Bauelementeabschnitt hat die Form:</p> <pre> ***** Components ***** .cmp (dev=BATTERY HOLDER,art=DH0632,typ=BQ_D30.96_LS20.5) B1 .cmp (dev=100nF,art=CC0845,typ=CO0603) C1cmp (dev=74LVCH16244,art=VD6226,typ=SSOP48_L15.85_W7.5_P0.635_M) U98 .cmp (dev=74LVCH16244,art=VD6226,typ=SSOP48_L15.85_W7.5_P0.635_M) U99 .cmp (dev=240MHz,art=MB1107,typ=r_xtal-vectron-vtype) X2 .cmp (dev=24.576 MHz,art=MB1109,typ=r_xtal-smx86) X3 </pre> <p>Der Bauelementebezeichner steht am Ende einer Bauelementezeile. Der Typbezeichner wird aus der Eigenschaft dev gelesen.</p> <p>Der erste Eintrag wird nach dem Import wie folgt im *.DIF -File dargestellt:</p> <pre> (DEV B1 (TYPE 'BATTERY HOLDER')) </pre> <p>Die Netzliste folgt der Bauelementeliste und hat die Form</p> <pre> ***** Wire List ***** DBFLS_N : U100 M20; R576 2; N40239141 : U49 22; TP16 1; DELX2 : U124 18; U38 G2; </pre>



Format	Beschreibung	Syntaxbeispiel
		<pre> ENTAP1_H : U40 19; R200 1; U39 11; N197626 : U32 B7; R138 1; N552085 : J4 54; J6 23; CFMIAD4 : U47 M6; U82 c4; U79 c4; U80 c4; U81 c4; U84 c4; \ U88 c4; ... ***** The end ***** </pre> <p>Es werden alle Zeichen bis zum Doppelpunkt (;) als Netzname importiert. Danach folgt eine Liste der Pins (1. Bauelementebezeichner, 2. Pinbezeichner). Wird eine Pinliste über mehrere Zeilen geführt, so wird die Zeile mit einem Backslash (\) verlängert, sodass der Importfilter weiterliest, bis er ein Zeilenende findet, welches nicht mit einem ‚\‘ versehen ist.</p>
SUPERMAX PCB Layout Format	Das SUPERMAX PCB Layout-Format wird aus einer Datei importiert. Defaultmäßig hat diese Datei die Erweiterung *.IPL.	<p>Die Syntax wird am folgenden Beispiel erkennbar:</p> <p>Beispiel:</p> <pre> .REM DAS IST KOMMENTAR .END .. -- ES FOLGT DIE BAUELEMENTEINFORMATION .COM 1 60000 155000 0 0 N N 0 2 /TEST/FILE .TXT 68610 152290 111 3 0 LB 6670 4 U01 .TXT 67500 150000 14 3 0 LB 70 4 U-74LS74/OS ... -- ES FOLGEN DIE NETZE .WLG 0 N NETZ1 .WLP 2 162500 167500 N 0 14 U02 255 2 .WLG 0 N NETZ2 .WLP 1 60000 155000 S 2 1 U01 -- PIN 0 1 .WLP 2 162500 137500 N 0 11 U02 – PIN </pre>



Format	Beschreibung	Syntaxbeispiel
SUPERMAX Wire List Output Format	Für den Import des SUPERMAX Wire List Output Formats wird eine Datei mit der Erweiterung *.OWL benötigt, welche die Bauelementeinformation und Netzlisteninformation enthält.	<p>Die Syntax soll an einem Beispiel aufgezeigt werden:</p> <p>Beispiel:</p> <pre> SUPERMAX E-CAD forward annotation from Mentor, version 4.5.4 ... ***** components *****cmp C1 (typ=0402rc,height=0.838mm,art=133.0097,dev=C1330097) .cmp U1 (typ=qfp_100sq,height=1.2mm,art=126.1122, dev=XC2C128_7VQ100C) ... ***** wirelist ***** ... '/CPU_A(2)':\ U6000 AC10 (option="pinuse:bidirectional"); \ U7000 AG10 (option="pinuse:bidirectional"); \ U9204 A1 (option="pinuse:input"); \ ... U1 71 (option="pinuse:bidirectional"); \ U4503 37 (option="pinuse:bidirectional"); \ U9000 A15 (option="pinuse:bidirectional"); '/CPU_A(20)':\ U6000 AH13 (option="pinuse:bidirectional"); \ U7000 AJ26 (option="pinuse:bidirectional"); \</pre> <p>Der Bauelementebezeichner folgt dem Schlüsselwort .cmp. Der Typbezeichner wird aus dem Parameter dev ausgelesen. Im Beispiel ist das der U1 mit dem Typ XC2C128_7VQ100C. Die Netznamen können in Hochkommas (")-stehen und werden von einem Doppelpunkt (;) gefolgt. Danach folgt die Auflistung der Pins des Netzes mit ihrem Bauelementebezeichner und der Pinnummer.</p>
TL_NETLIST-Format	Für den CAD-Import des TL_NETLIST-Formats ist eine Datei erforderlich. Diese hat in den meisten Systemen die	<p>Das Format wird mit dem Schlüsselwort PC_BOARD eingeleitet. Ein Bauelementeintrag hat folgende Form:</p>



Format	Beschreibung	Syntaxbeispiel
	Endung *.TL.	<pre> PC_BOARD ... ! NET_LIST !! COMPONENTS !! @ COMPONENT[1] !! @ ! REFERENCE_DESIGNATOR := 'C1'; !! @ ! TYPE := NORMAL_COMPONENT; !! @ ! PART_CODE_ID := 'V26803-B54-V78'; !! @ ! LIBRARY_IDENTIFIER := 'CSM-10U/16V-B'; !! @ ! PIN_IDS := PHYSICAL; !! @ ! COMPONENT_RULES !! @ ! !..FIN_COMPONENT_RULES; !! @ !..FIN_COMPONENT[1]; !! @ COMPONENT[2] !! @ ! REFERENCE_DESIGNATOR := 'C10'; !! @ ! TYPE := NORMAL_COMPONENT; !! @ ! PART_CODE_ID := 'V26803-B51-V2'; !! @ ! LIBRARY_IDENTIFIER := 'CSM-100N/50V/X7R'; !! @ ! PIN_IDS := PHYSICAL; !! @ ! COMPONENT_RULES !! @ ! !..FIN_COMPONENT_RULES; !! @ !..FIN_COMPONENT[2]; ... <HIER FOLGEN WEITERE BAUELEMENTEINTRÄGE> !! @ !..FIN_COMPONENT[n]; ... </pre> <p>Als Bauelementebezeichner werden die Daten hinter dem Schlüsselwort REFERENCE_DESIGNATOR importiert. Im obigen Beispiel sind das die Bezeichner C1 und C10.</p> <p>Als Bauelementetypen werden die Daten hinter dem Schlüsselwort PART_CODE_ID importiert. Im obigen Beispiel wird dem Bauelement C1 der Typ V26803-B54-V78, dem Bauelement C10 wird der Typ V26803-B51-V2 zugeordnet.</p> <p>Ein Netzeintrag in diesem Format hat folgende Form:</p>



Format	Beschreibung	Syntaxbeispiel
		<pre> PC_BOARD ... ! NET_LIST <HIER FOLGEN DIE EINTRÄGE FÜR DIE BAUELEMENTE> ... !!NETS !! @ NET[1] !! @ ! IDENTIFIER := '\$+3,3V'; !! @ ! VOLTAGE := '3,3'; !! @ ! PINS !! @ !! PIN[1] !! @ !! @ REFERENCE_DESIGNATOR := 'C16'; !! @ !! @ PIN_ID := '1'; !! @ !! @ SEQ_NO := 0; !! @ !! @ ..FIN_PIN[1]; !! @ !! PIN[2] !! @ !! @ REFERENCE_DESIGNATOR := 'C17'; !! @ !! @ PIN_ID := '1'; !! @ !! @ SEQ_NO := 0; !! @ !! @ ..FIN_PIN[2]; ... <HIER FOLGEN WEITERE NETZEINTRÄGE> !! @ !..FIN_NET[n]; ... </pre> <p>Als Netzname werden die Einträge des Schlüsselworts IDENTIFIER importiert. Im obigen Beispiel ist es das Netz \$+3,3V. Die Pininformation für den Bauelementebezeichner wird aus dem PIN-Eintrag REFERENCE_DESIGNATOR und die für die Pinnummer aus dem PIN-Eintrag PIN_ID gelesen.</p> <p>Der entsprechende Pineintrag in der *.DIF -Datei hat folgende Form:</p> <pre> (CLU '\$+3,3V' (L C16 1) </pre>



Format	Beschreibung	Syntaxbeispiel
		(L C17 1) ...)
TopCAD Exchange-Format	Für dieses Format wird eine Datei benötigt, die die Default-Erweiterung *.TXF besitzt.	<pre> component_list { component { din {"IC4"} symbol { name {" OSC-SMD "} ... } } } net_list { net { name {"\$NET1 "} supply {0} distance {0.13335} width {0.0254} pin_list { pin { din {"IC4"} name {"1"} ... } } } } } </pre>

Format	Beschreibung	Syntaxbeispiel
		<p>Hierbei handelt es sich um ein geschachteltes Klammerformat, der Bauelementebezeichner wird aus dem din-Parameter gelesen. Der Typbezeichner wird aus dem Symbolnamen eingelesen. Der Netzname wird aus dem name-Parameter des net-Eintrages ausgelesen. Die Pininformation wird aus din und name des pin-Eintrages gelesen.</p> <p>Die *.DIF -Dateieinträge haben dann die Form:</p> <pre>(DEV IC4 (TYPE OSC-SMD)) ... (CLU '\$NET1' (P IC4 1) }</pre>
UNICAD-Format	Das UNICAD-Format wird aus einer Datei mit der Default-Erweiterung *.NL importiert und ist ab CASCON-Version 3.4.3b verfügbar.	<p>Die Bauelementeinformation wird aus dem Abschnitt \$D NFDEVICE eingelesen. Die Pininformation wird aus dem Abschnitt \$D NFPIN eingelesen. Die allgemeine Syntax des Formates lässt sich wie folgt darstellen:</p> <pre>\$D NFHEAD <BESCHREIBUNG> <DATEN> \$D NFDEVICE <BAUELEMENTELISTE> \$D NFPIN <NETZ- UND PINLISTE> \$D NFSIGNAL <LISTE DER NETZE/ SIGNALE></pre>
Veribest-Format.	Um das Veribest-Format zu importieren, muss ein CAD-File angegeben werden.	<pre># DIE BAUELEMENTELISTE # NAME TYP TYP C10 VJ0805Y104JXXAT CAP RC0805 IC10 MAX811MEUS-T MAX811 SOT143 IC11 SN74LVC1G00DCK S_NAND DCK5 # NETZE</pre>



Format	Beschreibung	Syntaxbeispiel
		<pre># NAME BAUELEMENTE MIT PINS NET 'VCC' IC10-4 C10-4 ...</pre>
VICTORY_NL-Format	Die VICTORY-Netzliste besteht aus einer Datei.	<p>Ein Bauelementeeintrag hat die Form:</p> <pre>DEVICE = <DEVICE_NAME> CHAR <DEVICE_TYPE> ! <KOMMENTAR>;</pre> <p>Ein Netzeintrag hat die Form:</p> <pre>NODE NAME <NETZNAME> = NET_1 <PINLISTE> <PINLISTE >:= <DEVICE_NAME>_<PIN> [<DEVICE_NAME>_<PIN>];</pre> <p>Beim CAD-Import werden nur die Bauelemente und Netze importiert, bei denen die Schlüsselworte DEVICE und NODE in der ersten Spalte der Datei stehen.</p> <p>Beispiel:</p> <pre>DEVICE R33 = CHAR T\$0051; ! DEVICE R34 = CHAR T\$0051; ! ... NODE NAME 14N1884 = R34_1 U31_172 ; NODE NAME 14N1885 = R33_1 U31_165 ;</pre>
VIEWLOGIC-Format	Das VIEWLOGIC-Format besteht aus zwei Dateien.	<p>In der ersten Datei werden die Informationen über die Bauelemente in einer Liste wie folgt abgespeichert:</p> <pre><Datensatznummer> <Typname> <BauelementName></pre> <p>Andere Informationen werden überlesen. Die Syntax der Netzlistendatei entspricht der Syntax des VEWLOGIC_2/PKG-Formats und wird im nächsten Abschnitt beschrieben.</p>



Format	Beschreibung	Syntaxbeispiel						
VIEWLOGIC_2/PKG-Format	<p>Das VIEWLOGIC_2/PKG-Format besteht aus zwei Dateien:</p> <table border="1"> <thead> <tr> <th>Default-Erweiterung</th> <th>Kurzbeschreibung</th> </tr> </thead> <tbody> <tr> <td>*.PKG</td> <td>Bauelementeliste (Bauelement, Typ)</td> </tr> <tr> <td>*.NET</td> <td>Netzlistendatei</td> </tr> </tbody> </table>	Default-Erweiterung	Kurzbeschreibung	*.PKG	Bauelementeliste (Bauelement, Typ)	*.NET	Netzlistendatei	<p>In der *.PKG -Datei werden die Bauelementenamen wie folgt gespeichert:</p> <pre><BauelementeListe>:= <BauelementeTyp>; <ÜBERLESEN>;<BauelementNamenListe> [<BauelementeTyp>;<ÜBERLESEN>;<BauelementNamenListe> ...] <BauelementeListe>:= <BauelementName>[,<BauelementName>,...]</pre> <p>Beispiel: BCM3037;PQFP64;U14 CAP;A0805;C8,C9,C10,C23,C24,C25,C26,C27,C28,C29,C30,C31,C32,C33,C34,C35,C36,C37,C38,C39,C40,C41,C42,C43,C44,C45,C46,C47,C48,C49,C50,C51,C54,C55,C56,C57,CAPPL,CAP,C1,C2,C3 ...</p> <p>In der Netzlistendatei sind die Informationen über die Netze enthalten. Dort werden auch die Pininformationen gespeichert.</p> <pre><NetzListe> := <NetzName>;<PinListe> <PinListe> := <BauelementName> ^ <PinName>, [<BauelementName> ^ <PinName> ...]</pre> <p>Beispiel: ... UP_TDO;U14^30,U23^13 USER0;U10^13,U19^138 VCC;C16^1,C17^1,C18^1,C19^1,C20^1,C38^1,C65^1,C66^1,C67^1,C68^1,C71^1,C72^1,C73^1,C74^1,C75^1,C76^1,C77^1,C78^1,C79^1,C80^1,C81^1 ...</p>
Default-Erweiterung	Kurzbeschreibung							
*.PKG	Bauelementeliste (Bauelement, Typ)							
*.NET	Netzlistendatei							
ZUKEN_CCF-Format	<p>Das ZUKEN_CCF-Format beinhaltet die Netzlisteninformation in einer Datei. Als Default-Erweiterung wird *.CCF verwendet.</p>	<p>Im Header der Datei steht als Formatidentifikation \$CCF. Nach dem Schlüsselwort DEFINITION folgt die Bauelementeliste. Die Bauelemente sind nach Typen geordnet. Nach jedem Typ folgt die Aufzählung der Bauelementebezeichner. Die Netze werden hinter den Schlüsselworten NET, POWER und GROUND angegeben. Es werden die Netznamen,</p>						



Format	Beschreibung	Syntaxbeispiel
		<p>gefolgt von einem Doppelpunkt (:), und danach die Bauelementepins des Netzes angegeben. Das Semikolon (;) ist das Ende der jeweiligen Liste. Die einfache Syntax der ZUKEN_CCF-Datei soll durch ein Beispiel erläutert werden:</p> <p>Beispiel:</p> <pre> /*=====*/ /* ZUKEN CR3000/PWS CCF FORMAT NETLIST */ /* DESIGN NAME: BOARD_1 */ /*=====*/ \$CCF { DEFINITION { 0419670 :DR7, D\$R8; 68P-V-63-5-S :C206, C205; 68-0_25-5-S :R111, R110; 0327933 :ST21; CSE10PG-S :LD2; 150-0_25-5-S :R60; ... } NET { &N_001 :IC60(11), C136(1); N_002 :IC31(53), IC37(53), IC32(53), IC38(53),IC42(53),IC48(53); &N_XXX :C206(2), R111(1); ... } } </pre>
ZUKEN_REDAC-Format	Um die Netzliste im ZUKEN_REDAC-Format zu importieren, ist nur eine Datei anzugeben.	<pre> .HEA <HEADER> .TIM ZEIT .JOB PROJEKT_NAME ... </pre>



Format	Beschreibung	Syntaxbeispiel
		ADD_COM <BAUELEMENT_NAME> <BAUELEMENT_TYP> ...



Alle Layout Formate

CAD Tool	Firma	Format	Endung	Beschreibung
ACCEL PCB	Altium	P-CAD	*.pcb, *.asc	Layout öffnen -> Save Project As -> Export Protel PCB 2.8 ASCII
Allegro	Cadence	CADENCE	pcb, .cad, .fab	
		ODB++	.tgz, .tar, .gz, zip	
Altium Designer	Altium	ODB++	.tgz, .tar, .gz, zip	
		P-CAD	*.pcb, *.asc	
		PROTEL	*.pcb	
ARES Proteus	Labcenter Electronics	ODB++	.tgz, .tar, .gz, zip	
ASTER Quick Start	ASTER Technologies	AQS	.aqs	
Bartels Auto Engineer (BAE)	Bartels System GmbH	GENCAD	.cad	
		ODB++	.tgz, .tar, .gz, zip	
BoardStation BoardStation XE FabLink XE	Mentor Graphics	NEUTRAL	.nf, .neu, .vss,	
			.fm, neutral_file	
		GENCAD	.cad	
		ODB++	tgz, .tar, .gz	
Cadlink II	ASTER Technologies	RPDATA	rpdata.asc	
Cadstar	Zuken	GENCAD	.cad	
PCB Design		CADIF	.paf	
		ODB++	.tgz, .tar, .gz, zip	
CAMCAD	Router Solutions Inc	CAMCAD	.cc, .ccz	
CR3000	Zuken	CADIF	.paf	
PWS		ODB++	.tgz, .tar, .gz, zip	
CR5000	Zuken	CR5000-BD	.pcf, ftf	
Board Designer		CADIF	.paf	
		GENCAD	.cad	
		ODB++	.tgz, .tar, .gz, zip	
Docica 3.2&4.1	Alcatel	DOCICA	.doc	
Eagle CAD	CadSoft	FATF	.fat, fatf.asc,	



CAD Tool	Firma	Format	Endung	Beschreibung
			*.far, *.faz	
		GENCAD	cad	
EDWinXP	Visionics	GenCAM Standard	.gcm	
		ODB++	.tgz, .tar, .gz, zip	
Enterprise 3000	Valor	ODB++	.tgz, .tar, .gz, zip	
Expedition PCB	Mentor Graphics	GENCAD	.cad	
		ODB++	.tgz, .tar, .gz, zip	
Gencad Data	Teradyne (ex Genrad)	GENCAD	.cad, .gcd	
GenCAM Standard	IPC	GenCAM Standard	.gcm	
IPC-2581	IPC	IPC2581	*.cvg, *.xml	
IPC-356	IPC	IPC356	*.ipc, *.356	
i3070	Agilent	i3070	board	
Migger	Larisys	FATF	.fat, fatf.asc	
ODB++	Valor	ODB++	.tgz, .tar, .gz, zip	
ODB++XML	Valor	ODB++	.xml.gz	
OrCAD Layout	Cadence	OrCAD	.min	
		GENCAD	.cad	
		ODB++	.tgz, .tar, .gz, zip	
PADS	Mentor Graphics	PADS	.asc	
Power PCB		ODB++	.tgz, .tar, .gz, zip	
Pantheon	Intercept Technology Inc	GENCAD	.cad	
		ODB++	.tgz, .tar, .gz, zip	
P-CAD	Altium	P-CAD	*.pcb, *.asc	
		PDIF	.pdf	
		ODB++	.tgz, .tar, .gz, zip	
Protel	Altium	PROTEL	*.pcb	
		ODB++	.tgz, .tar, .gz, zip	
Pulsonix	Westdev Ltd	GENCAD	.cad	
		ODB++	tgz, .tar, .gz	
Seetrax Ranger	Seetrax CAE Ltd.	GENCAD	.cad	
Supermax ECAD	Dansk Data Elektronik (DDE)	ODB++	tgz, .tar, .gz	
	Mentor Graphics			
TestExpert	UGS	FATF	.fat, fatf.asc	



CAD Tool	Firma	Format	Endung	Beschreibung
(Fabmaster)				
	Tecnomatix			
TestSight	DeMille Research	GENCAD	.cad	
Trilogy 5000	Valor	ODB++	.tgz, .tar, .gz, .zip	
UNICAD	Cadence	UNICAD	.arc, .txt	
Unicam 6.2	UGS	RPDATA.DLL		
	Tecnomatix			
Veribest PCB	Intergraph	GENCAD	.cad	
Visula	Zuken	CADIF	.paf	



ISO 9001 zertifiziert

 GÖPEL electronic GmbH

Göschwitzer Str. 58/60
07745 Jena/Thüringen

03641 - 6896 0 Fon
03641 - 6896 944 Fax

sales@goepel.com
www.goepel.com

 sales@goepel.co.uk

 sales@goepelusa.com

 sales@goepel.asia

 sales@goepel.in