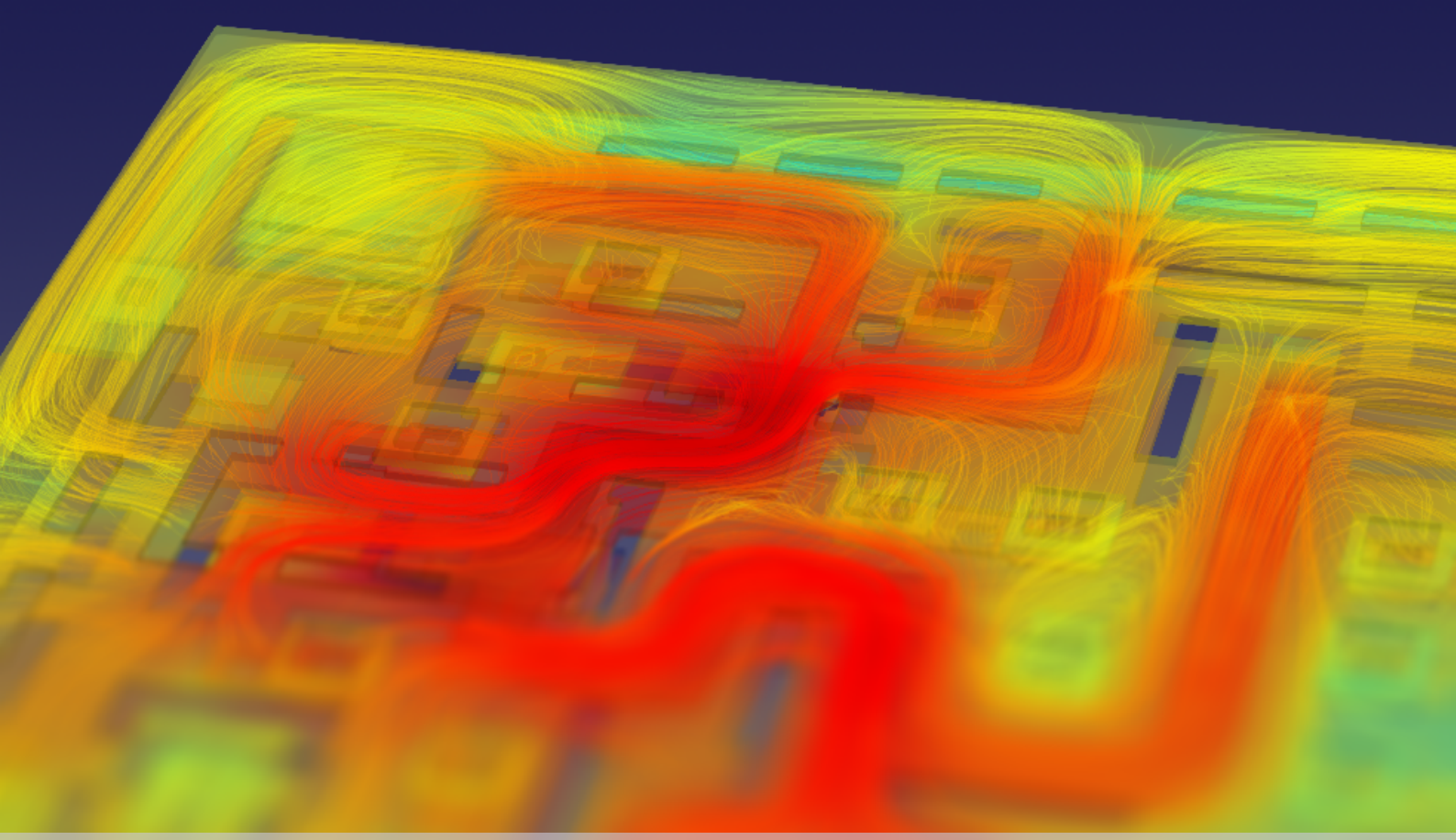

InductEx® 8

User Manual

SUN Magnetics

Copyright © 2026 SUN Magnetics (RF) (Pty) Ltd. All Rights Reserved for:
Inductex 2003-2026 Coenrad Fourie and TetraHenry 2014-2026 Kyle Jackman.

SUN Magnetics (RF) (Pty) Ltd
15 De Beer Street
Stellenbosch, 7600
Republic of South Africa

www.sun-magnetics.com

Permission is granted to anyone to make or distribute verbatim copies of this document as received, in any medium, provided that the copyright notice and the permission notice are preserved, and that the distributor grants the recipient permission for further redistribution as permitted by this notice.

InductEx, TetraHenry and JoSIM are registered trademarks of SUN Magnetics (RF) (Pty) Ltd.
CheckNet and MeshViewer are trademarks of SUN Magnetics (RF) (Pty) Ltd.
AutoCAD and DXF are trademarks of Autodesk Inc.
GDSII is a trademark of Calma, Valid, Cadence.
Linux is a registered trademark of Linus Torvalds.
Mac OS and OS X are registered trademarks of Apple Inc.
MATLAB is a registered trademark of The MathWorks Inc.
Ubuntu is a registered trademark of Canonical Ltd.
Windows is a registered trademark of Microsoft Corporation.
All other trademarks are the property of their respective owners.

InductEx 8.0.4 — Date: 13 November 2026

Credits

InductEx is based upon work supported financially by the South African National Research Foundation, grant numbers 69006, 78789 and 93586, with additions developed under funding agreements W911NF14-C-0089 and W911NF-17-1-0120 with the Intelligence Advanced Research Projects Activity (IARPA).

Contents

I Using InductEx Studio

1 Introduction . . . 10

1.1 InductEx 8 and Studio 10

1.2 A Note on Modelling 10

1.3 History 10

1.4 Modelling and Extraction Process Flow 10

2 Installing *InductEx* . . . 12

2.1 Installation for Studio 12

2.2 Overview 12

2.3 Download the software 12

2.4 Installing the Universal Licence Daemon 12

2.4.1 Upgrading an existing daemon . . . 13

2.4.2 Microsoft Windows . . . 13

2.4.3 Linux . . . 13

2.4.4 macOS . . . 13

2.4.5 Custom daemon and licence locations . . . 13

2.5 Installing InductEx 14

2.5.1 Microsoft Windows . . . 14

2.5.2 Linux . . . 14

2.5.3 macOS . . . 14

2.6 Generating the System Identifier 14

2.7 Test the Software 15

2.8 Third party software 15

3 Getting Started with Studio . . . 16

3.1 Start Studio 16

3.2 The workspace 16

3.3 First extraction: a microstrip 16

3.4 Inspect the model before solving 16

3.5 Other first examples 17

4 Projects and Extraction in Studio . . . 19
4.1 Create a project 19
4.2 The Basic workflow 19
4.2.1 Extraction goal . . . 19
4.2.2 Input files . . . 19
4.2.3 Electrical setup . . . 19
4.2.4 Review . . . 19
4.2.5 Check and run . . . 19
4.3 Raw IXI and source editing 19
4.4 Staged calculations 21
4.5 Artifacts and dependencies 22
5 Layouts and Process Files in Studio . . . 24
5.1 Open and navigate a GDS layout 24
5.1.1 Layers . . . 24
5.1.2 Hierarchy and inspection . . . 24
5.1.3 Select a GDS cell for extraction . . . 24
5.1.4 InductEx information on the layout . . . 24
5.2 Modelling changes and GDS copies 24
5.2.1 Viewer and Editor modes . . . 24
5.2.2 Geometry and selection . . . 24
5.2.3 Bookmarks and reports . . . 25
5.3 The process editor and cross section 25
5.4 A layout cross section 26
5.5 Export a technology description 26
5.6 Use of external layout tools 26
6 Meshes, Fields and Results in Studio . . . 27
6.1 Read the run result 27
6.2 The mesh viewer 27
6.2.1 Probe, measure and clip . . . 27
6.2.2 Snapshots and display preferences . . . 27
6.3 Field plots 27
6.3.1 Interpretation of a field view . . . 28
6.4 S-Parameter plots 28
6.5 Keep a calculation record 29
7 Workspace, Preferences and Help . . . 31
7.1 Files and folders 31
7.2 Find text, commands and help 31

7.3 Diagnostics 31
7.4 Workspace Dashboard 31
7.5 Import / Export Center 31
7.6 Preferences 31
7.7 Documentation and examples 34
7.8 About and licence status 34
7.9 Diagnostic log and support bundle 34
7.10 Common problems 34

II Input and Modelling Reference

8 Geometry Entry . . . 38
8.1 Drawing volume 38

9 Process File . . . 39
9.1 Use of the Process File 39
9.2 Open and Protected Process Files 39
9.3 File structure 39
9.3.1 Comments . . . 39
9.3.2 File header . . . 39
9.3.3 Record blocks . . . 40
9.3.4 LDF parameter entry . . . 40
9.3.5 Process file global parameters . . . 40
9.3.6 Layer parameters . . . 43
9.3.7 Mesh mask layer entries . . . 45
9.3.8 Operators . . . 45

10 Layout Input . . . 46
10.1 The IXI file format 46
10.1.1 IXI description . . . 46
10.1.2 IXI units . . . 46
10.1.3 IXI commands . . . 46
10.1.4 IXI Controls . . . 49
10.1.5 IXI records and fields . . . 49
10.1.6 IXI expressions . . . 55
10.2 GDSII 56
10.2.1 Limitations of GDSII support . . . 56
10.3 Port label input file 56
10.4 Modelling aids defined on layouts 57
10.4.1 Extraction boundary . . . 57

10.4.2 Operator declarations . . . 57

11 Circuit Netlist . . . 58

11.1 Title line 58

11.2 Comments 58

11.3 Circuit elements 58

11.4 Ports 58

11.5 Control block 58

11.5.1 BACK-ANNOTATE command . . . 59

11.5.2 SET command . . . 59

11.5.3 SWEEP command . . . 59

12 Ports . . . 60

12.1 Ports for inductance extraction 60

12.1.1 Definition of ports and terminals . . . 60

12.1.2 Terminal geometries of inductance ports in integrated circuit models . . . 60

12.1.3 Ports drawn on layouts . . . 60

12.1.4 Port or terminal label declarations on layouts . . . 60

12.2 Conductors (Ports) for capacitance extraction 61

12.3 Ports for impedance extraction 62

13 Off-Chip Structures . . . 63

13.1 Arbitrary structures from Gmsh 63

13.1.1 Physical structures . . . 63

13.2 Package modelling 63

13.2.1 Wedge-type wire bonds . . . 63

13.2.2 Ball-and-stitch wire bonds . . . 64

13.2.3 Bump bonds . . . 64

14 Output Files . . . 65

14.1 Solution output 65

14.1.1 Solution result . . . 65

14.1.2 Mesh quality . . . 65

14.1.3 Quality of the singular value decomposition for inductive networks . . . 65

15 Field Plotting . . . 67

15.1 Viewing fields in Studio 67

15.2 Volumes 67

15.3 Planes 68

III Command Line Reference

16 Command Line Parameters 70

16.1 *InductEx* command line paramters 70

16.1.1 Earlier IXI transition description 70

17 Exit Codes 71

Bibliography 73

Index 74

Using InductEx Studio

1 Introduction 10

2 Installing *InductEx* 12

3 Getting Started with Studio 16

4 Projects and Extraction in Studio 19

5 Layouts and Process Files in Studio 24

6 Meshes, Fields and Results in Studio 27

7 Workspace, Preferences and Help 31

Introduction

1.1 InductEx 8 and Studio

InductEx 8 includes InductEx Studio for preparing projects, inspecting layout and process inputs, running extractions and viewing supported results. The first part of this manual describes that graphical workflow. Start with Chapter 3 after installation. The later parts retain the input formats, modelling rules and command-line reference for direct source entry, existing integrations and scripts.

For normal use, open an IXI project in Studio and follow its Basic workflow. Separate layout, mesh and plotting tools remain useful for tasks beyond Studio's supported controls; references to them in the modelling chapters describe those additional workflows.

1.2 A Note on Modelling

Before you read any further, or do any experiments, take note:
Numerical calculations are only as reliable as the models used to obtain them. No solution is absolutely correct – some are just better than others. You are strongly advised not to put too much trust in a calculation result until you are familiar with modelling and interpreting results and have verified one of your own extractions against a known measurement result for a similar structure.

If you have limited experience in modelling layouts for parameter extraction, please study the examples described in the accompanying design reference manual carefully before you attempt to do your own extractions. A simple modelling mistake (such as inverting the polarity of a port on a layout or making a connection mistake in a circuit netlist) can lead to wildly inaccurate solutions.

***InductEx* warns about possible modelling errors. It is helpful to know how to interpret those warnings to identify mistakes.**

1.3 History

InductEx [1–5] was developed to enable VLSI circuit designers to extract the inductance of complex 3D structures in superconducting integrated circuits, but also supports normal conductivity and the calculation of inductance in simpler structures such as SQUID loops or wires. Current distribution in, and magnetic fields in and around conductors can also be computed. Although early versions were focused on inductive substructures, *InductEx* is now aimed at full-gate extraction [5] and analyses that determine external magnetic field and trapped flux [6–8] coupling to circuit inductance.

InductEx also supports capacitance, impedance and S-parameter calculations for superconductor and normal conductor integrated circuit and printed circuit board layouts, with application ranging from filters in microwwave systems to resonators and couplers in qubit systems.

InductEx functions well as a stand-alone command line programme that accepts layout artwork from GDSII input files created by any layout software, but was carefully designed to be integrated with CAD software through its own, more powerful IXI interface.

InductEx is a powerful tool for geometry processing, modelling and segmentation of integrated circuit layout structures and the solution of inductive or capacitive networks. For inductance calculation, *InductEx* uses uses its own numerical solver, *TetraHenry* [9] – a purpose-built magneto-quasistatic, electroquasistic, EMQS and full-wave solver – to calculate current distribution in layout structures, from which inductance and coupling is determined.

Your use of *InductEx* is subject to the licence terms made available on the SUN Magnetics website (www.sun-magnetics.com) and included with every installation file.

Part of the design philosophy behind *InductEx* is easy integration into an EDA tool chain. *InductEx* thus slots well into the Synopsys design environment, for which it is actively maintained.

InductEx also interfaces well with LayoutEditor and Klayout, can be used with XIC, and with the correct script files it works directly from Cadence Virtuoso.

Finally, *InductEx* was developed to allow inductance extraction of complete circuit cells. The intention is to allow users to work directly on "ready-for-fabrication" cells without the need to alter layouts during modelling. To this end, *InductEx* supports auxiliary layers and operators, and cells that pass extraction may be sent for fabrication without the need to strip out modelling text or objects.

1.4 Modelling and Extraction Process Flow

InductEx is first and foremost an integrated circuit inductance calculator. The basic process flow for modelling and extraction is illustrated in Fig. 1.1.

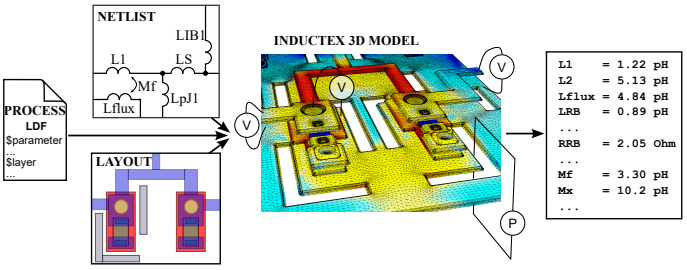

Figure 1.1: InductEx process flow for inductance extraction.

Basic extraction requires three inputs: An integrated circuit layout file, a fabrication process file that describes the fabrication steps and layer dimensions and material parameters

of the fabricated chip (or printed circuit board), and a netlist that describes the equivalent electrical circuit to which extracted electrical parameters must be assigned.

The process file is described in Section 9, the layout input artwork syntax and requirements in Section 10, and the netlist syntax and requirements in Section 11.

Installing *InductEx*

2.1 Installation for Studio

Install the Universal Licence Daemon and InductEx package for your operating system, then start the installed InductEx Studio application. Review its licence status and use the first Studio extraction in Chapter 3 to check the installation. The command-line checks below remain useful for scripts and troubleshooting. Studio includes its own supported layout, mesh, field and S-Parameter views; external viewers described below are optional for those workflows.

2.2 Overview

Starting with *InductEx* v8, a unified licensing and authentication mechanism based on the *SUN Magnetics Universal Licence Daemon* is used. This daemon is a common background service that manages licence authentication for all SUN Magnetics software products, including *InductEx*, *JoSIM Pro*, and future products. As a result, installation of the licence daemon is a mandatory prerequisite before InductEx can be executed.

The licence daemon needs to be installed only once per machine and may be shared by multiple SUN Magnetics products installed on the same system. Install the current supported daemon package, version 1.5.2 or newer. Version 1.5.2 introduced additive enrolment: a client can supply its configured licence file to an already running daemon without replacing licences used by other products. Core and Studio both request the `TTH` product entitlement for InductEx; a licence for another SUN Magnetics product alone does not authorize InductEx.

2.3 Download the software

All installation files are distributed via the SUN Magnetics website at www.sun-magnetics.com. Downloads are available exclusively to registered users and can be accessed after logging in to the user portal (select "Log In" in Fig. 2.1). Once authenticated, navigate to the `Downloads` page by clicking on the account icon and selecting "Downloads" (shown in Fig. 2.2), where platform-specific installers for the following components are provided:

- The SUN Magnetics Universal Licence Daemon,
- *InductEx* application binaries that include *TetraHenry*.
- Other SUN Magnetics install files, such as *JoSIM-Pro* and a stand-alone *TetraHenry*.

2.4 Installing the Universal Licence Daemon

Run the supported licence daemon installer for your operating system before installing InductEx. Follow the installer's instructions; administrative privileges may be required. InductEx packages do not install or upgrade the shared daemon.

Core and Studio launch the installed daemon with the configured licence file before requesting a signed grant. A daemon already serving another SUN Magnetics product receives this additional file through additive enrolment. Allow up to 60 seconds for the initial licence check to complete.



Figure 2.1: The main website at www.sun-magnetics.com.



Figure 2.2: Location of the Download section on www.sun-magnetics.com.

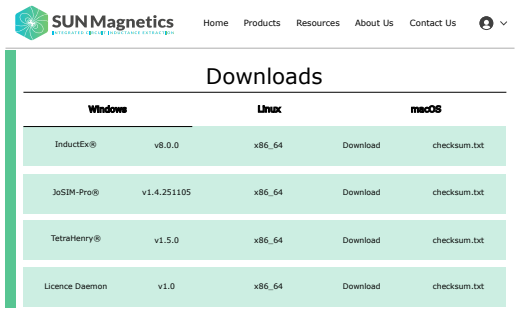

Figure 2.3: Example of downloads available for Windows.

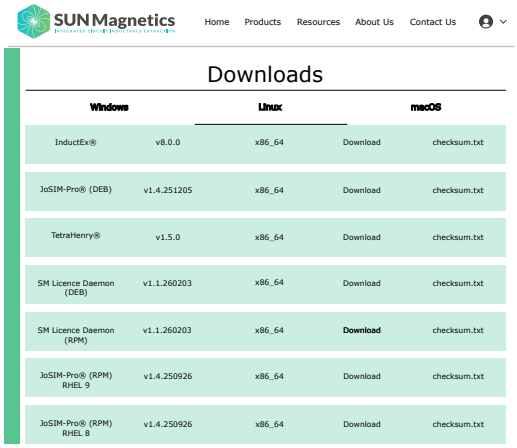

Figure 2.4: Example of downloads available for Linux.

2.4.1 Upgrading an existing daemon

If an older daemon is installed, obtain the current supported daemon installer from SUN Magnetics. Before upgrading, close SUN Magnetics applications and wait for all simulations to finish. Back up existing licence files and application licence settings, then follow the daemon installer's upgrade instructions. Restart the shared licence service as directed by the installer, or restart the computer before reopening applications. Installing new binaries while an old broker is still running does not update that broker.

Keep the existing licence files for other products. Do not overwrite them with an InductEx licence or manually replace individual daemon binaries. If the supported installer is unavailable, contact SUN Magnetics support for the upgrade package. Reopen Studio and use its licence Retry action, or rerun the Core example below, to check the upgraded service.

2.4.2 Microsoft Windows

The default installation directory is:

```
c:\Program Files\SUN Magnetics\Licenses\bin
```

The daemon executable is `smlicensed.exe`. Default installations require no daemon environment override.

The licence key file, once obtained from SUN Magnetics, should be placed in

```
c:\Program Files\SUN Magnetics\Licenses
```

The default licence filename is `sm_license.txt`. See the custom-location instructions below if your licence uses another filename.

2.4.3 Linux

Download the package for your Linux environment. Debian (Ubuntu and similar) and Fedora (Red Hat and similar) are provided. Unpack the tar.gz archive (double-click on the archive in a file explorer).

For Debian, install from the terminal with:

```
sudo dpkg -i package\_name.deb
```

For Fedora, install from the terminal with:

```
sudo rpm -i package\_name.rpm
```

The default installation directory is:

```
/usr/local/share/sunmagnetics/licenses/bin
```

The licence key file, once obtained from SUN Magnetics, should be placed in

```
/usr/local/share/sunmagnetics/licenses
```

The default licence filename is `sm_license.txt`. The daemon executable is `smlicensed` in the `bin` directory.

2.4.4 macOS

Download the package for macOS. Unpack the tar.gz archive (double-click on the archive in Finder). This will extract to a .pkg macOS installable file.

Double-click and install the extracted .pkg file by following the on-screen instructions.

The default installation directory is:

```
/Applications/SUN Magnetics/Licenses/bin
```

The licence key file, once obtained from SUN Magnetics, should be placed in

```
/Applications/SUN Magnetics/Licenses
```

The default licence filename is `sm_license.txt`. The daemon executable is `smlicensed` in the `bin` directory.

2.4.5 Custom daemon and licence locations

Use an absolute path to your InductEx licence file in Studio's General preferences (`LicenseFileName`). Core reads the same setting from `.sunmagnetics/ixcfg.ini` in your home directory. If another product already uses `sm_license.txt`, keep that file and configure your separate InductEx licence

instead.

For a daemon installed outside the default location, set `SMLICENSED_PATH` to the full executable path, including `smlicensed.exe` on Windows or `smlicensed` on Linux/macOS. Both Core and Studio support this override; it is not a directory path or a licence-file path. Core otherwise uses the `LicenseDaemon` setting in `ixcfg.ini`, then the platform default. Studio additionally recognizes `SMLICENSED_EXE` ahead of `SMLICENSED_PATH`, followed by its General preference and the platform default. Clear a stale Studio-only override when configuring a shared custom path.

2.5 Installing InductEx

After the licence daemon has been successfully installed, *InductEx* can be installed using the provided platform-specific installer or archive. The installation process follows standard conventions for the target operating system.

2.5.1 Microsoft Windows

Run the installer (a .MSI file), and accept the licence terms. You will be asked to select an install directory. The default is

```
c:\Program Files\SUN Magnetics\InductEx
```

The executable (binary) files are installed under the `bin` subdirectory, while documentation such as the user and design reference manuals, the licence terms and the change log are copied to the `doc` subdirectory.

The environment variable `Path` is appended with the value

```
c:\Program Files\SUN Magnetics\InductEx\bin
```

Without administrator privileges, you cannot (and should not!) execute *InductEx* from `Program Files` directory. Therefore, examples are copied upon installation to:

```
c:\Users\YourUserName\SM\examples
```

2.5.2 Linux

Download the package for your Linux environment. Debian (Ubuntu and similar) and Fedora (Red Hat and similar) are provided. Unpack the tar.gz archive (double-click on the archive in a file explorer).

For Debian, install from the terminal with:

```
sudo dpkg -i package\_name.deb
```

For Fedora, install from the terminal with:

```
sudo rpm -i package\_name.rpm
```

The default installation directory is:

```
/usr/local/share/sunmagnetics/inductex
```

The executable (binary) files are installed under the `bin` subdirectory, while documentation such as the user and design reference manuals, the licence terms and the change log are copied to the `doc` subdirectory.

Examples are in the `/inductex/examples` subdirectory–by default in

```
/usr/local/share/sunmagnetics/inductex/examples
```

Copy the entire directory to your home directory to run examples from there.

User and design reference manuals are in `/inductex/docs`

2.5.3 macOS

Download the package for macOS. Unpack the tar.gz archive (double-click on the archive in Finder). This will extract to a .pkg macOS installable file.

Double-click and install the extracted .pkg file by following the on-screen instructions.

The default installation directory is:

```
/Applications/SUN Magnetics/InductEx
```

The executable (binary) files are installed under the `bin` subdirectory, while documentation such as the user and design reference manuals, the licence terms and the change log are copied to the `doc` subdirectory. The executables are also symbolically linked to `/usr/local/bin` for easy path discovery.

Examples are in the `/InductEx/examples` subdirectory–by default in

```
/Applications/SUN Magnetics/InductEx/examples
```

Copy the entire directory to your home directory to run examples from there.

User and design reference manuals are in `/InductEx/docs`

2.6 Generating the System Identifier

After installing the daemon and InductEx, open a terminal or console in a directory you can write to. Execute *InductEx* with the single parameter `-id` to ask the daemon to generate the system identifier. If *InductEx* is on your executable search path, use:

```
inductex -id
```

The system identifier, `SM_SysID.txt`, will be written to the directory from which *InductEx* was called. The identifier command does not require an activated InductEx licence. If the file is not generated, check the daemon installation or custom executable path.

Email `SM_SysID.txt` to support@sun-magnetics.com, including your customer code or latest SUN Magnetics invoice number. Request the InductEx licence for your purchased tier. Save the returned licence file in the platform's default licence directory as `sm_license.txt`, or select its actual absolute filename in General preferences as described above.

The system identifier is an ASCII text file, structured as demonstrated below.

```
# SUN Magnetics software system identifier
# Generated from InductEx v8.0.0, 2026-01-22 15:23:13, from Windows
# ----------------------------------------------------------------
↪ ------------------------------------------------------------
# For key activation, email this identifier file to
↪ support@sun-magnetics.com and supply your customer code and/or
↪ last
# SUN Magnetics invoice number as reference in the subject line.
# ----------------------------------------------------------------
↪ ------------------------------------------------------------
HvZFcRBOgpQoS;ERVrEJomQE>p;[>ORQnB<UNTVoqra<NyKdN;fmDbvC]Gyehy;Qj:R:
↪ ::xQ^k<LZVttSi[UhhwRSmg_]vTbHQQGC@LOuk;qCIRHGLJtFa\bKjjKQ[nd
DxRsWfZZLkMWNb@hxxPQLy_\i`Y^;\pDkpSJadmWwkcGNFTuVl=mceXPqAWtlEV^aUZg
↪ sq@daI\UfYtKZ>jsbYbUnAX;FoGMXSWYXmfrrienwEOpE[]wlgru<WMe]oB`
...
```

2.7 Test the Software

Once the licence file is installed or configured, test InductEx and daemon authentication from a writable copy of the shipped examples. Both Core and Studio require a signed, supported-tier `TTH` grant; a reachable daemon alone does not establish that the licence is usable.

Navigate to:

```
/examples/inductance/microstrip/
```

Then execute:

```
inductex microstrip.ixi
```

If the installation is correct and the licence key is accepted, *InductEx* will complete the calculation and return results in a subdirectory:

```
/examples/inductance/microstrip/output
```

One of the files written to the output directory is `sol.txt`. It contains an overview of the simulation process and results.

If `PRODUCT_NOT_LICENSED` remains after the licence check completes, verify that the selected file includes the `TTH` entitlement and that the running daemon is version 1.5.2 or newer. Upgrade an older broker using the supported installer procedure above. Repeated runs are not a substitute for correcting the daemon or licence configuration. If a current daemon still rejects the configured InductEx licence, contact support with the daemon version and displayed error; do not send licence keys in diagnostic logs.

2.8 Third party software

Starting with *InductEx* 8, the dependence on third-party software was removed. Previously, Gmsh [10] was used to segment layout models as triangles and tetrahedra. That functionality is now provided by *InductEx* . For viewing meshes, current distribution and field plots, the new tool *MeshViewer* was developed.

Getting Started with Studio

InductEx Studio is the graphical interface to InductEx 8. It brings the input files, layout view, process view, mesh and calculation results into one workspace. This chapter describes a first extraction with an installed example. The command line remains available for scripts and integration with other design software; its parameters are described in the last part of this manual.

The layout, process description and electrical model still determine the calculation. Studio helps you enter and inspect these inputs, but the modelling advice at the start of this manual applies equally to a calculation started from the graphical interface.

3.1 Start Studio

Install InductEx and the SUN Magnetics Universal Licence Daemon as described in the installation chapter. Start InductEx Studio from the installed application entry. On Windows the application executable is `InductExStudio.exe`. On Linux and macOS use the installed application launcher so that the required libraries can be found.

Studio checks the licence during startup. If the licence is missing or invalid, use the reported details to correct the licence file or daemon location. The **General** page in **Tools** → **Preferences** contains these paths. **Help** → **About** provides **Retry Licence Check** and **Generate System ID**. The latter depends on the daemon and produces the identifier needed when obtaining a licence. Opening files for inspection does not establish that the product is licensed for extraction.

3.2 The workspace

The file browser is on the left and the open documents occupy tabs in the centre. An IXI project has its own pages within its document tab. **Basic** is the normal starting point; **Raw IXI** contains the complete project text. **Artifacts** lists referenced files and outputs, and **Results** shows the latest extraction summary. **S-Parameters** appears when an extraction supplies a suitable response.

View → **File Browser** and **View** → **Output / Results** show or hide the two outer panels. Drag a splitter to change its size. **View** → **Reset Layout** restores the standard panel arrangement. Project geometry and process views can appear beside the source when the window is wide enough; at smaller widths use the corresponding page. A collapsed panel can be restored with its splitter.

The status bar reports the active document, file kind, saved state, diagnostics, run state and licence state. A document with unsaved changes must be saved before a calculation can use those changes. Menu commands follow the active document: selecting a text file or a process file does not make it an extraction project.

3.3 First extraction: a microstrip

Use **Help** → **Examples / Templates** to find the supplied examples. Select the microstrip inductance example and use **Copy to Folder** to make a working copy in a directory where you can write. Keep the copied files together. Alternatively, copy the complete installed `examples/inductance/microstrip` directory before opening it.

1. Use **File** → **Open Folder** to show the working directory in the file browser. Open `microstrip.ixi` with **File** → **Open** or from the browser.
2. On **Basic**, inspect **Extraction goal**. This example uses `MQS` for inductance extraction. The selected mode determines the electrical setup shown later.
3. Select **Input files**. The example refers to `process.ldf`, `microstrip.cir` and `microstrip.gds`. Paths are interpreted relative to the IXI project. Resolve any missing-file indication before proceeding.
4. Select **Electrical setup**. Inspect the netlist and layout ports. The netlist describes the circuit to which extracted values are assigned. For this example, keep the supplied netlist and port declarations.
5. Select **Review** and then **Check and run**. Read the items requiring attention. Use **Review issues** to reach an affected setting or source location.
6. Save the project. Select **Run extraction**, or **Run** → **Run full extraction**. The output panel records the calculation messages. Wait for the run to finish before using its results.
7. Open **Results** to inspect completion, elapsed time, the final Core stage and warning/error counts. Open the solution output from **Artifacts**; for the numerical entries and units see the Output Files chapter.

Run → **Open Output Folder** opens the directory associated with the project. It is useful when comparing calculations or collecting output files. The availability of a file in that directory alone does not prove that the latest run completed successfully; always read the run summary and output log.

3.4 Inspect the model before solving

For a new model, use the staged commands described in Chapter 4. **Premesh** lets you inspect the processed geometry and **Mesh** lets you inspect the segmentation before running the solver. Check the process layers, port locations and electrical connectivity. A completed file check does not establish that these are physically correct.

To stop an active operation, select **Run** → **Stop**. Cancellation is reported in the run state. A cancelled or failed extraction must not be treated as a successful calculation.

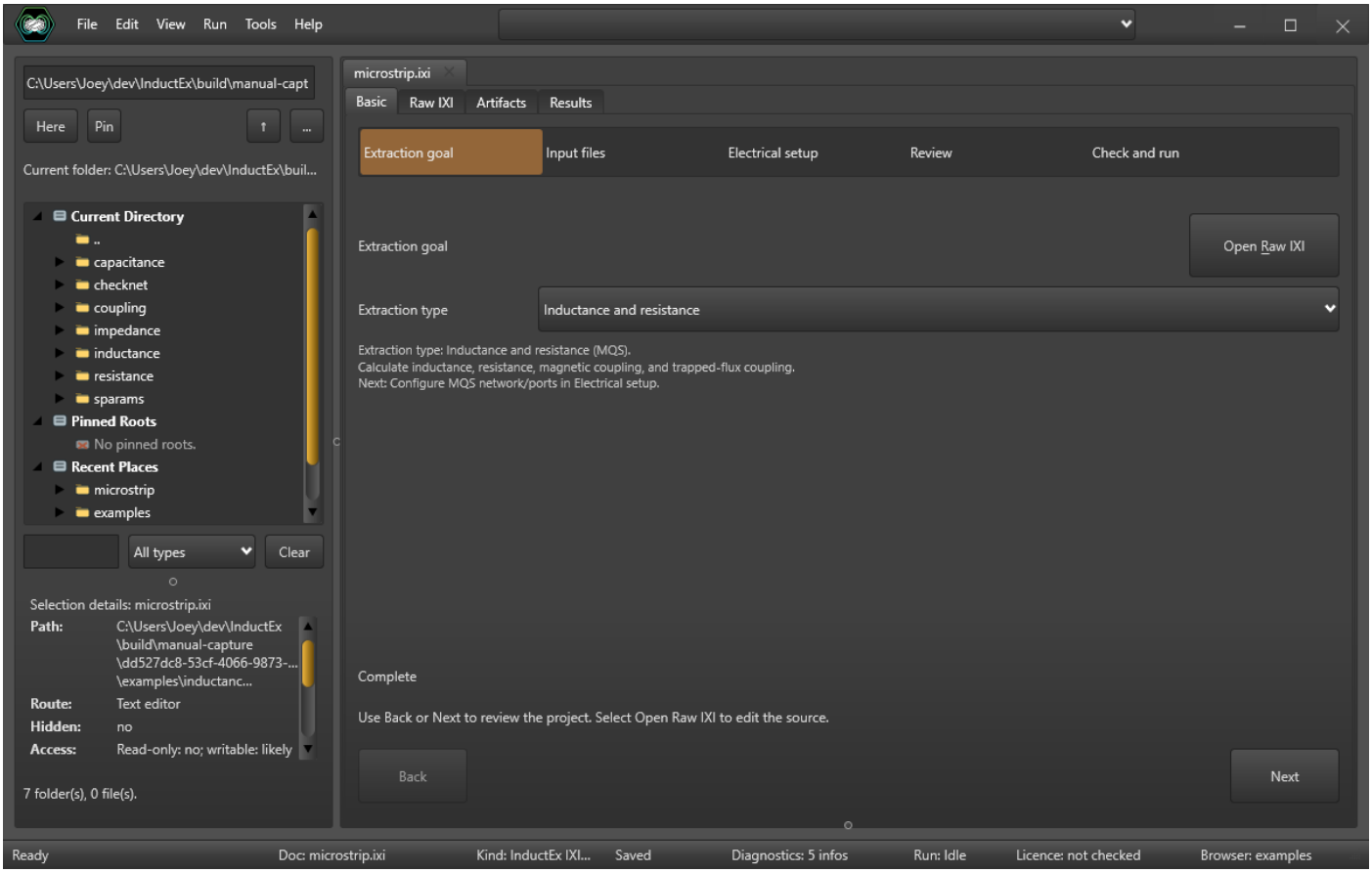

Figure 3.1: The Studio workspace with the microstrip IXI example open.

3.5 Other first examples

The supplied `capacitance/parallel_plates/parallelplates.ixi` example uses `EQS` and inline IXI geometry. Its conductor labels are `C1M0` and `C2M1`; an external GDS file is not required for this example. Open it in Studio and review the conductor summary in **Electrical setup** before running it.

The supplied `sparams/hairpinline_bandpassfilter/hairpinline_bandpassfilter.ixi` example gives a filter workflow. Copy its whole directory, inspect its frequency and port settings, and run the extraction. A suitable successful response is displayed on **S-Parameters**; Chapter 6 describes that view.

Use the resistance, impedance, coupling and field examples for the corresponding calculations. Their inputs and modelling requirements differ. Do not change the mode of an example and assume that its original port or conductor definitions are sufficient for the new calculation.

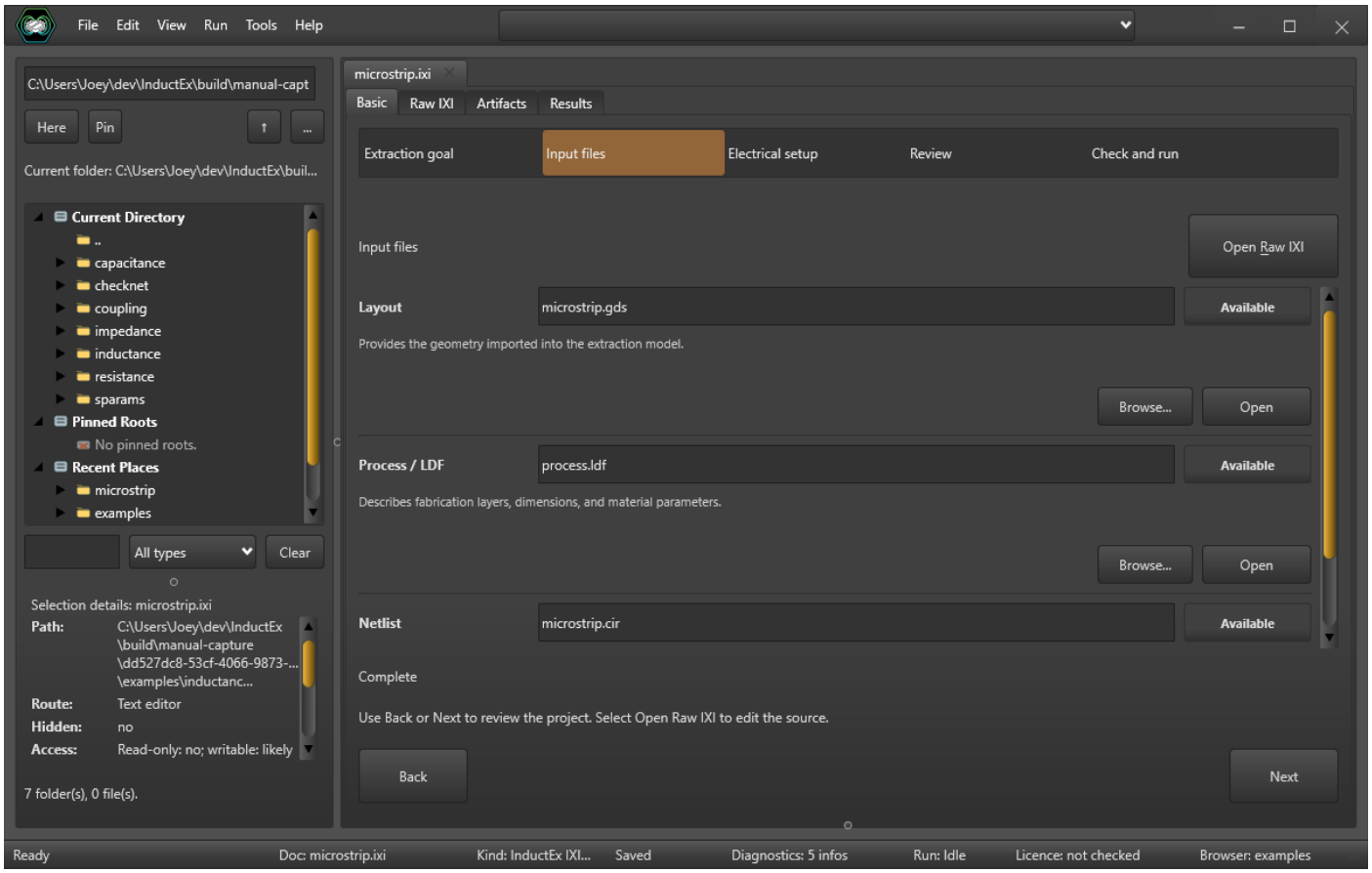

Figure 3.2: Input file review for the microstrip example. The process filename in this example is process.ldf.

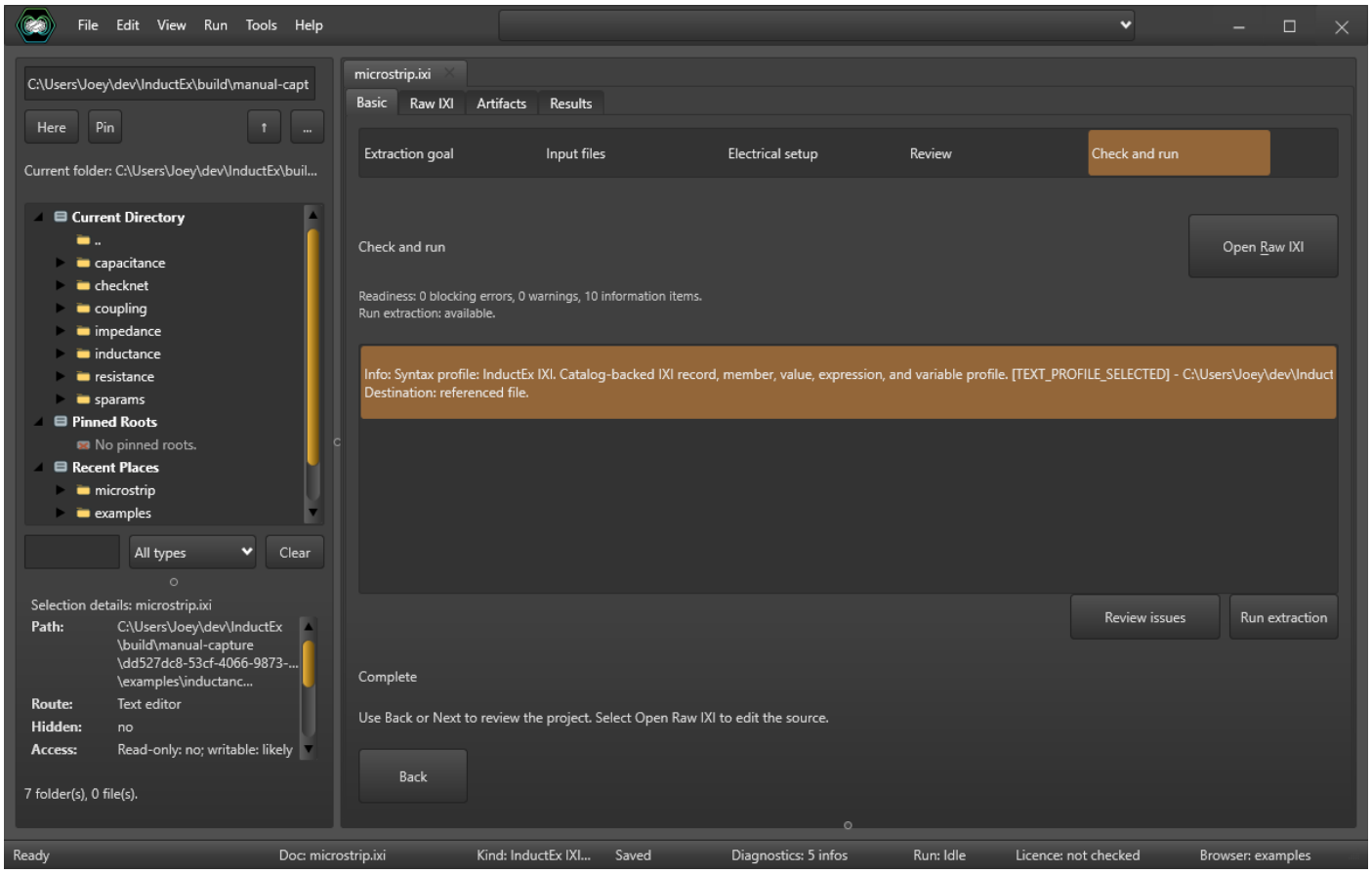

Figure 3.3: Check and run presents the current project checks before extraction.

Projects and Extraction in Studio

An IXI file is the project description used by Studio. It records the calculation commands, file references and any geometry entered directly in IXI. Studio's structured controls change this text; there is no separate set of hidden calculation settings that replaces the IXI file. See the Layout Input chapter for the complete IXI syntax.

4.1 Create a project

Use **File** → **New** and select **IXI Project**. The next page accepts an extraction type, layout input and process/LDF file. Expand the optional circuit-netlist input when a netlist is required. These entries can be supplied later. Use **Back** to revise the file type or **Cancel** to leave without creating a file.

Save the new IXI file in its working directory before running it. Relative paths are easiest to maintain when the project and its inputs are kept together. Moving only the IXI file can break its references.

The same wizard creates an **InductEx Netlist**, **Layer Definition File** or **Other Text File**. Creating a template does not provide the fabrication process or circuit model for your device; enter or obtain those descriptions separately.

4.2 The Basic workflow

Basic divides common project preparation into five steps. Use the step list or **Back** and **Next** to move between them. The steps report incomplete settings, items requiring attention and source errors.

4.2.1 Extraction goal

Select the intended extraction type. The mode is recorded in the IXI `Mode` command. Studio uses the mode to show the appropriate electrical setup. Changing it does not automatically convert geometry labels, a circuit netlist or frequency settings from one problem to another.

4.2.2 Input files

Select the fabrication process/LDF and, where required, the circuit netlist and layout. Browse for a file or edit its path. Open a referenced input to inspect its contents. The path status distinguishes a selected file from a missing file and settings that need attention.

Inline `$STRUCT`, `$PATH` and `$POLY` geometry is also supported. A project with inline geometry need not have a primary GDS input. Multiple layout imports, transformations and more involved structure definitions are described by the complete IXI source. Follow the source-reveal action when a setting cannot be represented unambiguously by a Basic control.

4.2.3 Electrical setup

For an MQS network calculation, inspect the netlist and the layout-port diagnostics. The option to run without a netlist is intended for current/field inspection; it does not create a circuit network. Use **Open netlist** to edit the circuit description and the source-reveal actions to inspect the relevant IXI declarations.

For EQS capacitance calculations, review the conductor-label summary. Labels identify the conductors whose capacitance is to be calculated. Their layer names and positions must agree with the model. The Ports chapter explains their syntax.

For S-Parameters, inspect the S-Parameter block and its ports. For characteristic impedance, inspect the impedance ports and terminal labels. A visible setup panel describes the required declarations; it is not a replacement for the physical port definitions.

4.2.4 Review

The review gathers the extraction goal, input files, electrical and mesh setup, use of more advanced IXI entries and items requiring attention. Read it before running a new project or after making a substantial change.

Settings outside the simple workflow remain in the IXI text. Studio provides source navigation rather than silently discarding those settings. Repeated declarations or unsupported structured edits can require work in **Raw IXI**. Correct syntax errors there before returning to Basic editing.

4.2.5 Check and run

This step combines current project checks with **Review issues** and **Run extraction**. After a run, use its **Results**, **Artifacts** and **Logs** actions. The run button is available only when the current project and application state permit extraction. Saving a file or clearing editor errors alone does not guarantee a valid physical model or licence.

4.3 Raw IXI and source editing

Raw IXI exposes the complete project. Use it for commands, expressions, imports and geometry that need direct source entry. Changes made through the Basic controls are reflected here. Preserve the `$END` boundaries and nesting described in the IXI reference.

IXI, LDF and netlist editors provide language assistance, completion suggestions and diagnostics for their respective formats. Completion and diagnostic preferences are on **Tools** → **Preferences** → **Editor**. Suggestions help with accepted syntax; they do not verify that a chosen value describes the intended device.

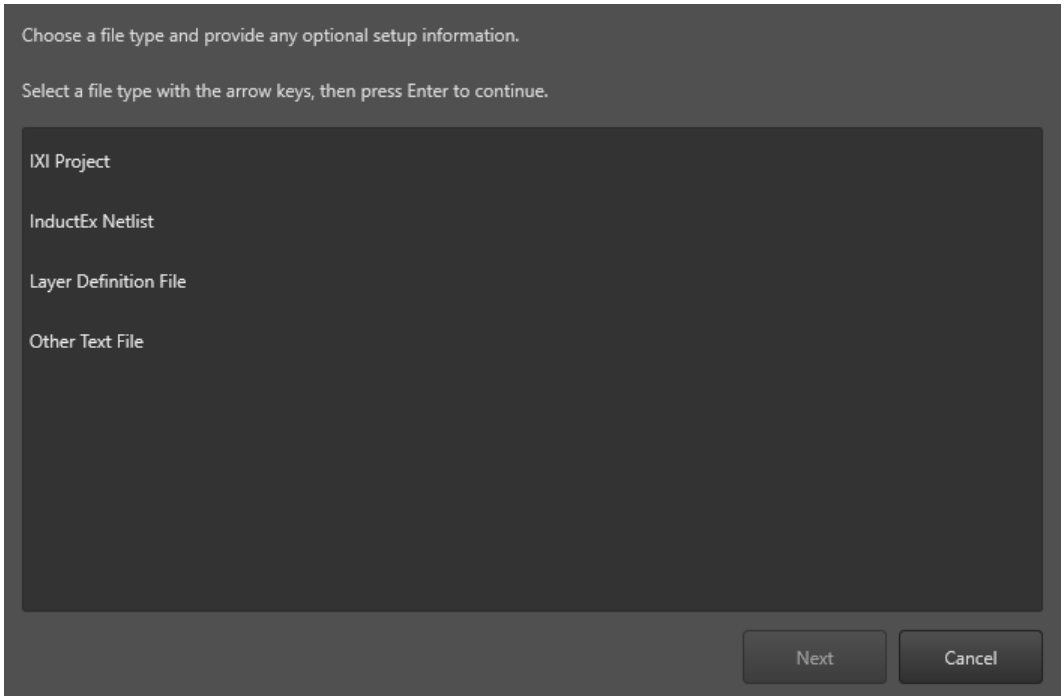

Figure 4.1: The new-file wizard creates an IXI project, netlist, layer definition file or other text file.

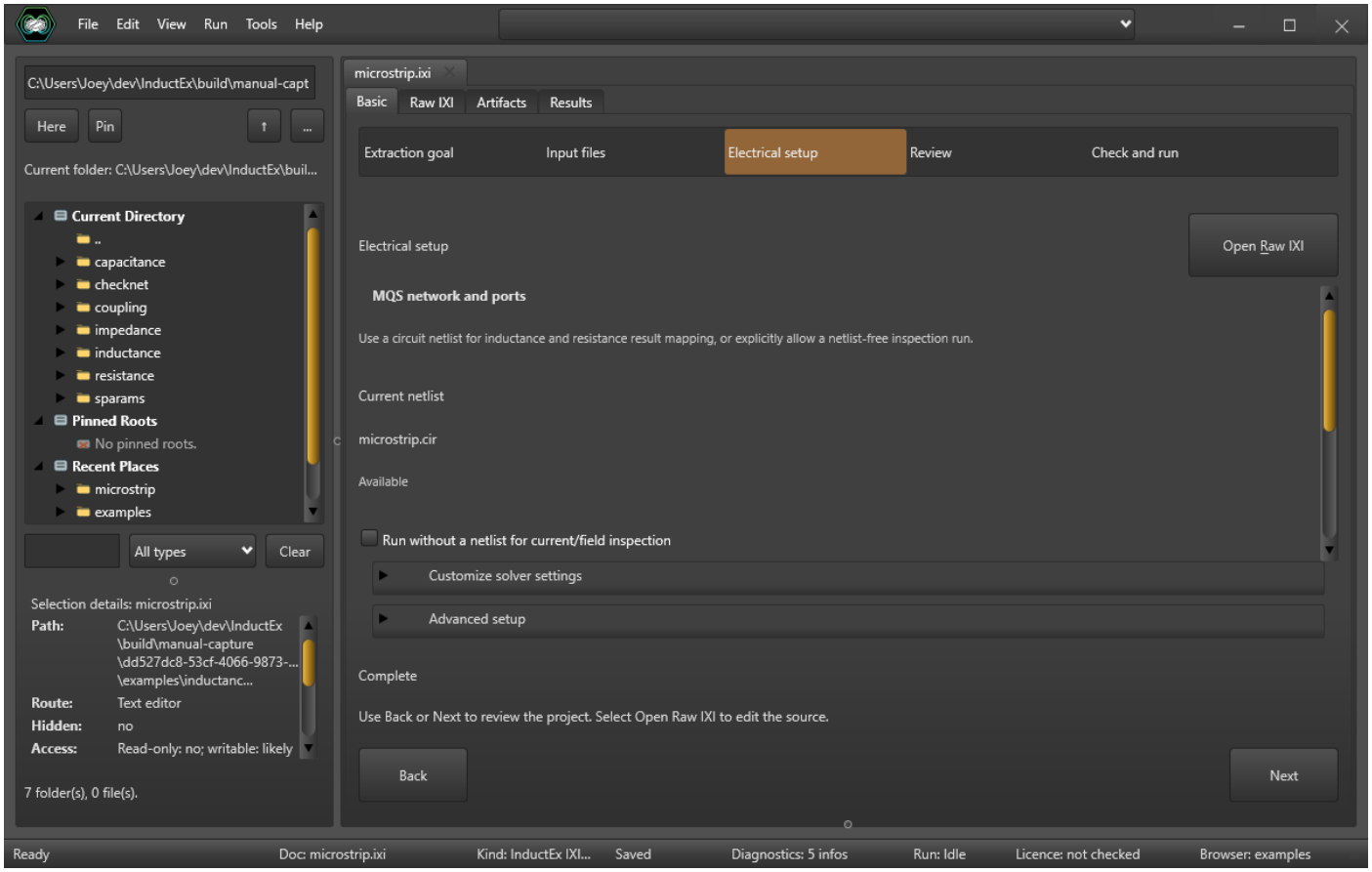

Figure 4.2: Electrical setup for an MQS project with a circuit netlist.

Use the **Edit** menu for undo, redo, cut, copy, paste, select all, find and replace. The editor's context menu provides the related commands. Find supports forward and backward navigation; replacement changes the source and therefore requires saving before another run. Use workspace text search when the text may be in another project input. Chapter 7 distinguishes this from the global search for commands and help.

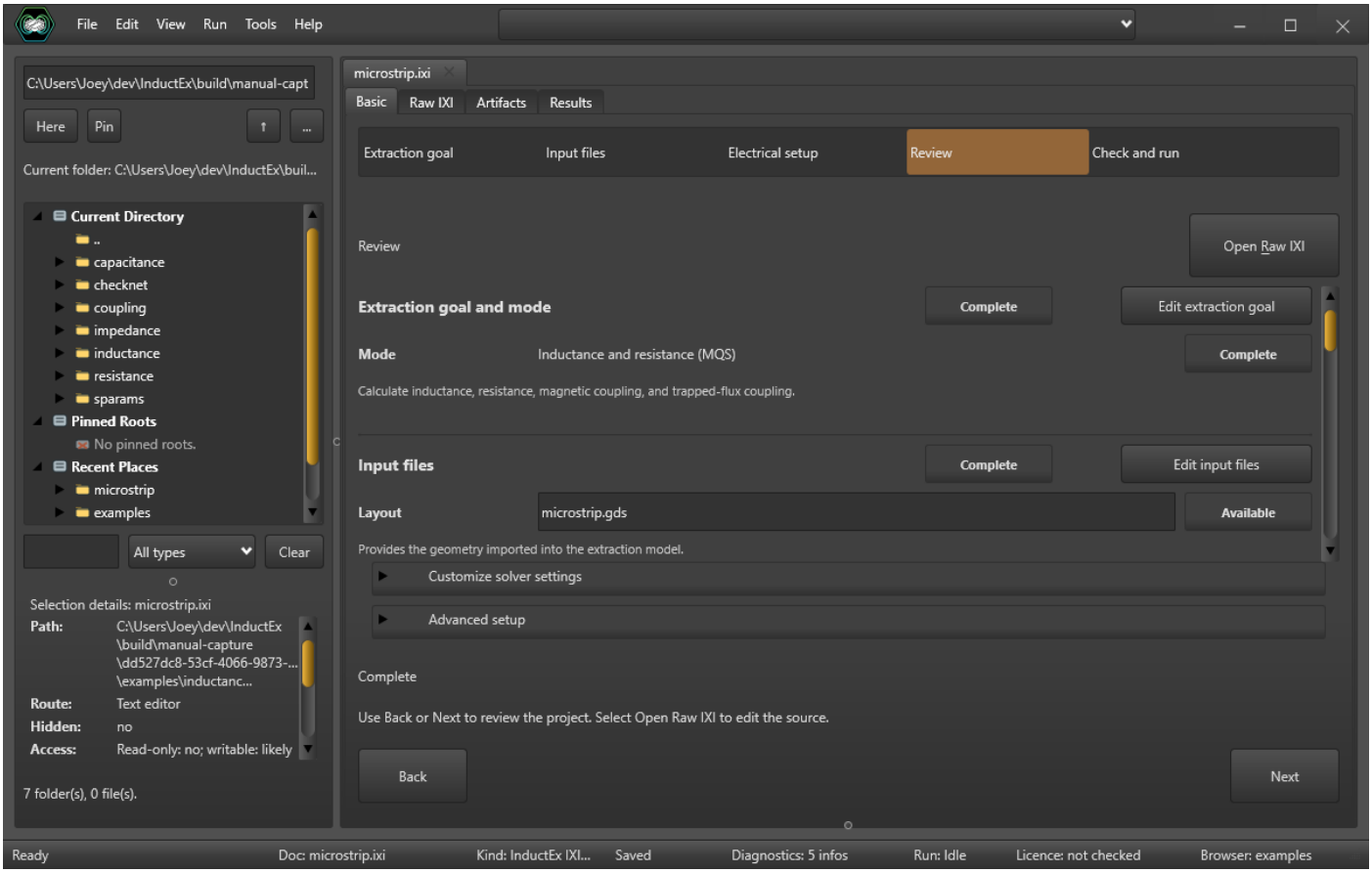

Figure 4.3: The project review gathers the settings and items requiring attention.

Go to Line reaches a known source position. **Go to Symbol / Record** helps locate a declaration in the current source. **Toggle Bookmark** marks the current editor position; **Next Bookmark** and **Previous Bookmark** return to these positions. **Toggle Fold / Unfold Current Block** collapses or expands the block at the caret without changing its text.

Add Next Occurrence adds the next matching occurrence to multiple cursors. Check all selected occurrences before typing a common replacement, then use **Clear Multiple Cursors** to return to ordinary editing. The available editor actions follow the document's editable state and feature settings. **Project Search** opens Global Workspace Search from the active IXI project.

File → **Save**, **Save As** and **Save All** operate on editable documents. Save As changes a document's identity and can alter the base directory for relative references. Keep required files with the new project and recheck their status. Closing a changed document offers the normal save decision; cancelling the close retains the document.

4.4 Staged calculations

The **Run** menu provides the following operations for the active IXI project:

Initialise / Reset
Initialises or resets the Core session for the project. Use it when starting a fresh sequence of stages.

Premesh
Processes the model to produce geometry for inspection before final segmentation.

Mesh
Builds the calculation mesh from the prepared model.

Run solver
Runs the solver on the prepared calculation state.

Recalculate
Recalculates using the available staged state. Its availability depends on the completed stages and current project.

Run full extraction
Runs the complete extraction workflow for the project.

Stop
Requests cancellation of the active operation.

Open Output Folder
Opens the project's output directory.

Inspect geometry and segmentation between stages. At wide window sizes Studio presents the staged geometry beside the project pages. At narrow sizes use **Live Mesh**. The same mesh controls are described in Chapter 6.

Changing the source can make an earlier model or result stale. A new run, a reset or another project using the Core can also invalidate staged state. Follow the enabled commands and reported state; do not assume that a mesh still displayed on screen belongs to the current source.

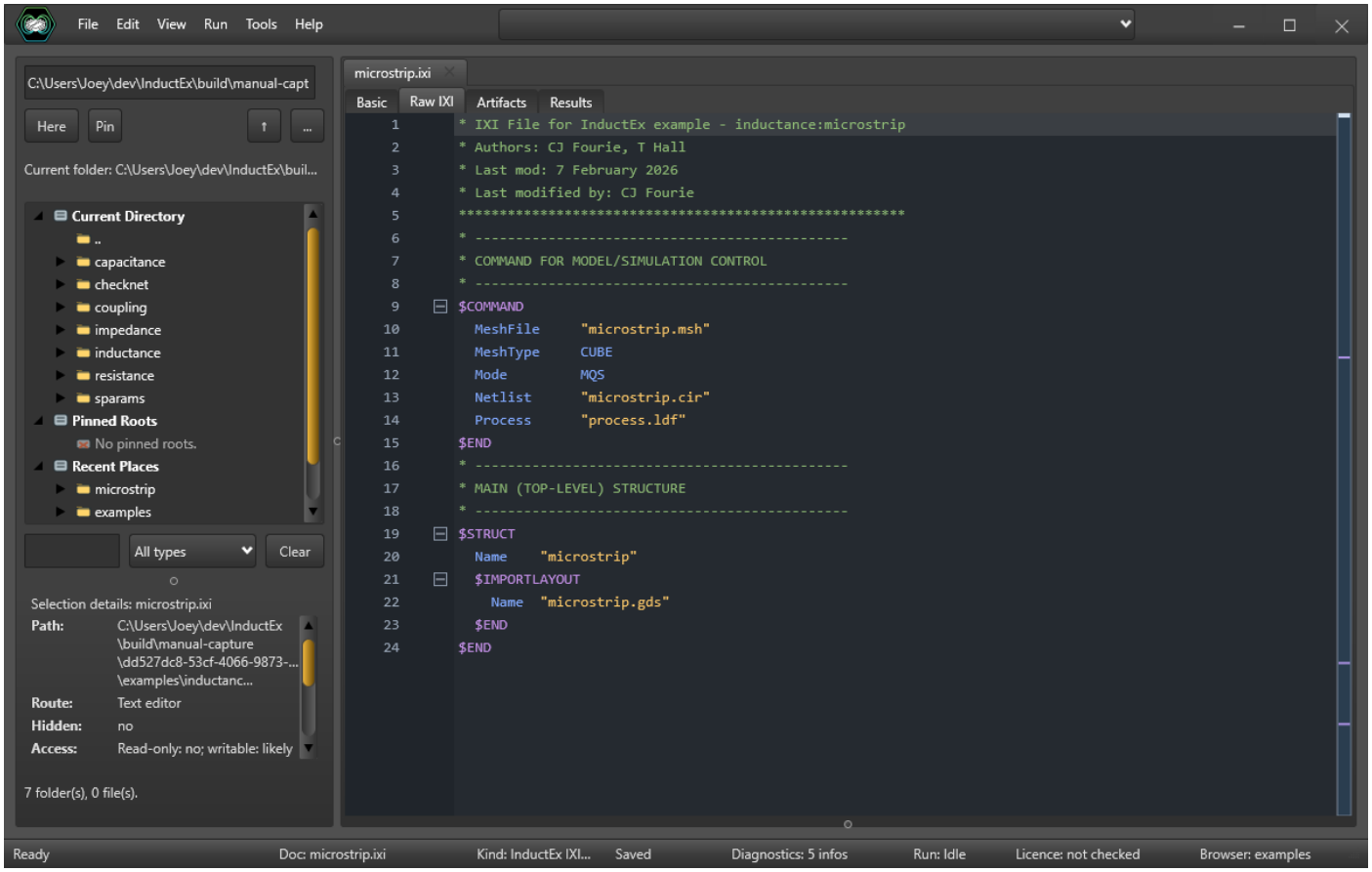

Figure 4.4: Raw IXI contains the complete source used by the project.

4.5 Artifacts and dependencies

Artifacts lists files referenced by the project and result files associated with its output directory. Use **Open** to inspect an input or result, and the location action to reach it on disk. The list reports the file's role and status. **Refresh Results** rescans the available outputs.

For an imported mesh, the project offers **Open Referenced Mesh**, **Open All Meshes** and **Preview Transform**. Use the preview to inspect the declared placement. It does not rewrite the imported source mesh.

Keep the output log with any numerical result you intend to use. Warnings about singular networks, mesh quality and missing connections can explain unexpected extracted values. The later reference chapters retain the details needed to interpret them.

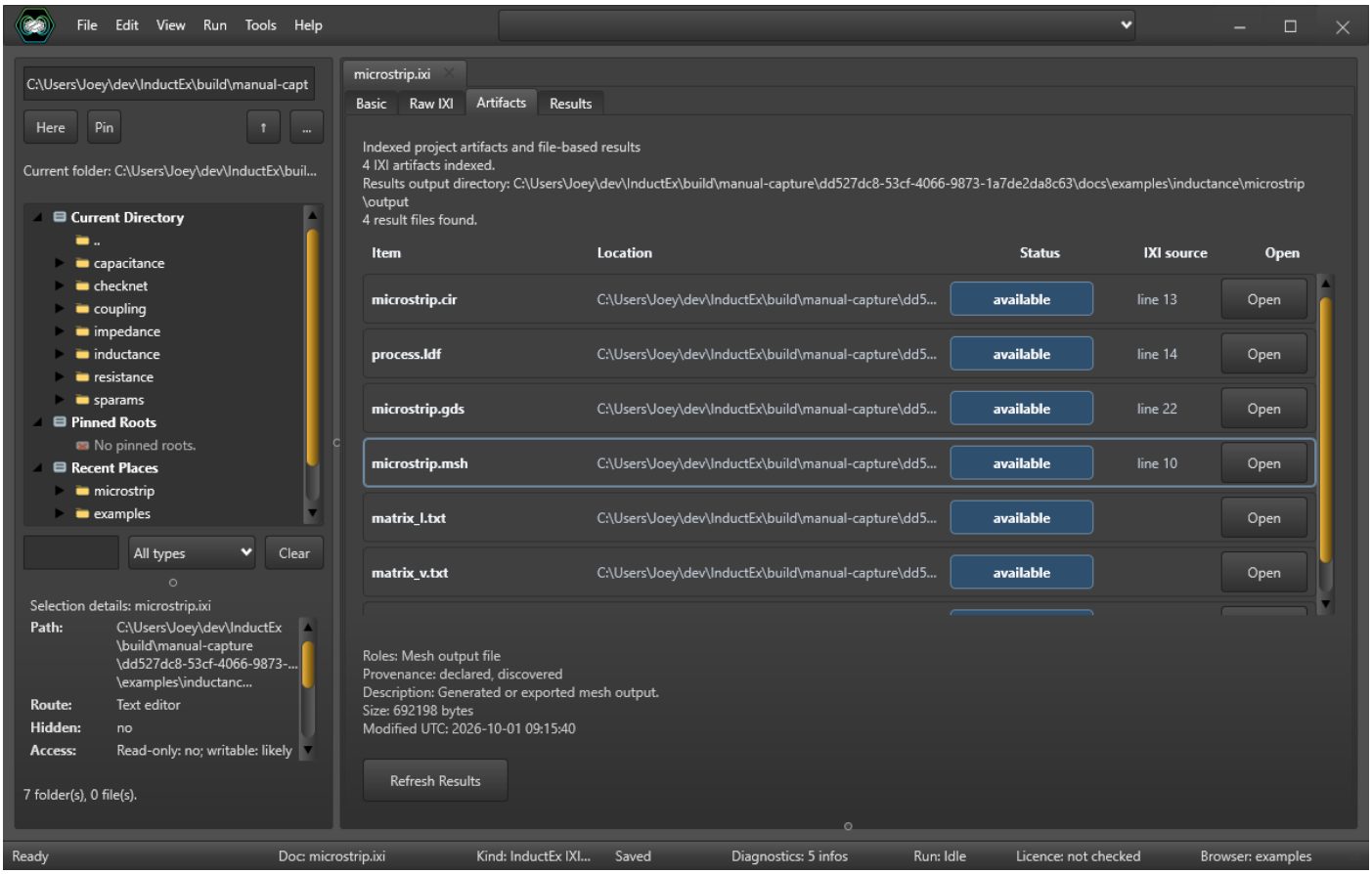

Figure 4.5: Artifacts connects the project source to its referenced input and output files.

Layouts and Process Files in Studio

Studio opens GDSII layouts and LDF process files directly. Use these views with the active IXI project to check how layout layers become physical structures. The Geometry Entry, Process File, Layout Input and Ports chapters retain the complete definitions and modelling rules.

5.1 Open and navigate a GDS layout

Open a `.gds` file from the browser, with **File** → **Open**, or from the project's Artifacts page. Loading is reported in the document. The viewer, layer list, hierarchy and inspector share the loaded layout.

Use the toolbar to fit the layout, zoom, reset the view or zoom to the selected object. View history provides backward and forward navigation. Enable the grid and crosshair when positioning or inspecting geometry. The grid spacing and minimum on-screen spacing can be set under **Preferences** → **GDS Viewer**.

5.1.1 Layers

The Layers panel lists layer/datatype rows. Show or hide individual rows, use **Show all** or **Hide all**, and isolate a row or layer. Filters distinguish text and geometry. Use the row controls to change display properties and the reset actions to restore selected or all display settings. **Save** and **Load** retain and restore the layer display setup.

Isolation can dim inactive layers or hide them, as selected in the GDS preferences. Visibility and colour changes are display settings; they do not change the fabrication process or remove geometry from the source layout.

5.1.2 Hierarchy and inspection

The Hierarchy panel exposes the structures and their references. Select the required structure and use **Fit selected** to inspect it. The Inspector reports the selected object's properties. **Copy details** and **Copy snippet** help record an object or use its information in the project source.

Use **Overview**, **Layers**, **Hierarchy** and **Inspector** to expose the corresponding panels. If the side panels take too much space, resize or collapse them with the splitter.

5.1.3 Select a GDS cell for extraction

To extract a particular cell from a GDS file, set `Top "cell_name"` inside the project's `$IMPORTLAYOUT` record. Use the name shown in the GDS Hierarchy panel. The Layout Input chapter gives a complete example under `ImportLayout`.

In the IXI project's **Basic** page, expand **Advanced setup**, then **Layout/imports** and **Structures and layout references**. Select the parent `$STRUCT` and its `$IMPORTLAYOUT` reference, enter the GDS cell name in **$IMPORTLAYOUT Top**, and select **Apply**. Alternatively, edit the record directly in **Raw IXI**; the editor recognises `Top` and provides completion and field help.

Selecting a cell in the GDS viewer changes the displayed layout. Set the project's `Top` import field to use that cell for extraction. The **Make Top** action for an IXI structure sets `$COMMAND Top`, which selects the main IXI structure rather than a cell within the imported GDS file.

5.1.4 InductEx information on the layout

The **InductEx overlay** adds the available process and project interpretation to the GDS view. This depends on a matching LDF/project context. A GDS file opened by itself cannot establish which fabrication process or extraction project should be used.

Inspect port text, extraction boundaries, operators and auxiliary layers with their process context. Studio's interpretation assists navigation and diagnostics; the source declarations and the physical model remain decisive. The Ports chapter explains positive and negative terminals, virtual cuts and conductor labels.

5.2 Modelling changes and GDS copies

Studio provides modelling-object and geometry editing actions in the GDS view. Use the available add-object commands and the editor's properties to define the required object. **Delete** and **Restore** operate on the selected working objects. Inspect the selection and pending changes before saving.

5.2.1 Viewer and Editor modes

The mode button changes between **Viewer** and **Editor**. Viewer mode is used for navigation and inspection. Editor mode enables the applicable drawing, selection and editing commands. The status text identifies the current mode. A context-menu command can be disabled when the selection, hierarchy level or read-only state does not permit that edit.

In Editor mode, right-click empty space and use **Add InductEx Object Here** to create a **Port Label**, **Split Terminal +**, **Split Terminal -**, **Virtual Cut**, **TERM Line**, **Extraction Boundary** or **Operator Marker**. Enter the name, layer and coordinate fields in the corresponding dialog and inspect its validation/preview before accepting. These objects use the port, terminal and operator conventions retained in the modelling reference.

5.2.2 Geometry and selection

For an editable boundary or path, **Edit Selected** offers **Insert Vertex Here**, **Delete Vertex Here** and **Properties**. Use **Edit**

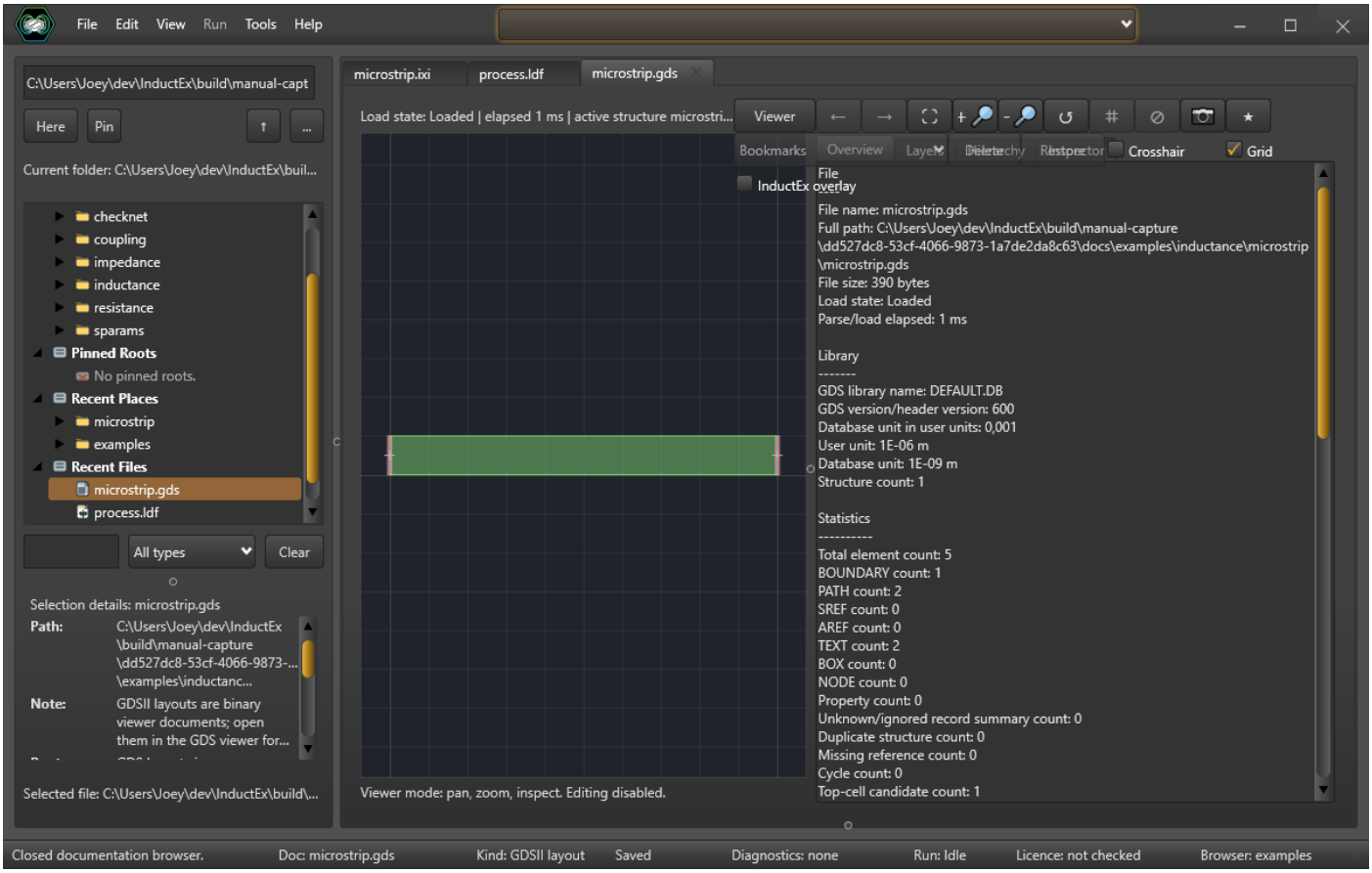

Figure 5.1: The microstrip layout opened in Studio's GDS viewer.

Port/Label for an applicable modelling label. The **Transform** group provides move, copy, delete, layer/datatype/texttype remap, rotate, mirror X, mirror Y and scale for the supported selection. Selection commands include zoom, copy summary, clear, select all visible and select active layer. Inspect what is selected before applying a transform.

When the selected object is an SREF or AREF, use **Edit Reference Placement** for its placement. **Edit Referenced Structure** enters the referenced structure and **Return to Parent Structure** returns to its parent. Changing a structure definition can affect other references to that structure; changing a placement concerns that reference. Geometry reached through a child reference must be edited at an appropriate hierarchy level.

Undo and redo apply to the GDS edit session. Keep the original layout and inspect the saved derived file after editing; the visible working overlays alone are not the saved GDS output.

5.2.3 Bookmarks and reports

Bookmarks retain useful layout views. Use the available add, restore and delete bookmark actions to revisit a location. Snapshot/report export records the displayed layout and its supporting information. It does not replace an exported GDS file.

Use the document's save/export workflow to create the derived GDS file. Check the destination and the reported summary. Keep an original copy when preparing a model from fabrication artwork, and reopen the derived file to check it before referring to it in an IXI project. An edited GDS copy does not automatically update every IXI file that referred to the original.

The Import / Export Center offers applicable GDS export tasks, including layout copies and snapshots. A snapshot records the current view for a report; it is not layout data that can replace a GDS input.

5.3 The process editor and cross section

Open the project's LDF from Artifacts or the browser. The source editor is accompanied by the process cross section. With a wide window these appear beside each other; at smaller widths use **Source** and **Cross Section**. **Fit** brings the stack into view.

The view identifies superconducting, isolation, resistive and normal material layers, markers, auxiliary layers and invalid entries. Select a layer in the cross section to inspect it or navigate to its source. A source selection can identify the related process element in the view. The context menu provides fit, zoom, source navigation, clear selection and material-legend controls.

The **Scale** setting changes how the layer stack is displayed.

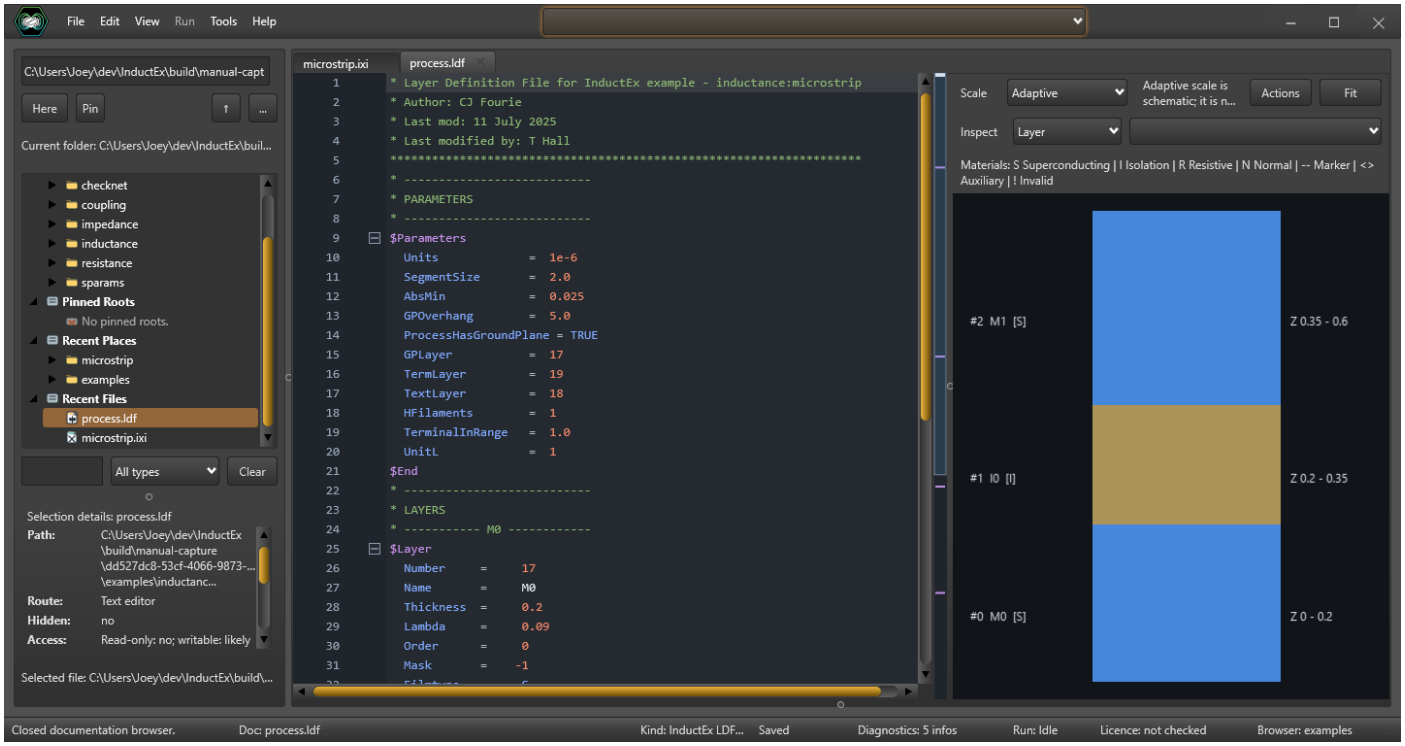

Figure 5.2: The process source and cross section for the microstrip example.

Adaptive scale is schematic; it is not a physical scale. Read the layer dimensions in the LDF when comparing thicknesses or separations. Do not measure a schematic stack image as if its pixel spacing were a device dimension.

The **Inspect** controls also expose the available conditional connections and junction inspections. Select the layer, connection scenario or junction to read its explanation and see the related elements. These views derive from validated process declarations; unavailable relationships are not invented.

If an edit makes the LDF invalid, Studio can retain the last valid process snapshot and explicitly report that the current source has errors. Correct the source and wait for the updated view before using it to interpret the model.

5.4 A layout cross section

The GDS cross-section probe relates a line through the layout to the process stack. Supply the appropriate LDF/project context and use the probe to select the section line. The resulting section helps inspect the layers and geometry along that line. This is an inspection of the specified model, not an independent reconstruction of the manufactured device.

5.5 Export a technology description

Tools → **Export InductEx-LVS Technology JSON** converts supported LDF process information into a technology description for InductEx-LVS. Open a suitable LDF or a project that supplies it, inspect the conversion diagnostics and select an output destination. Resolve unsupported or invalid process entries before relying on the export. The export does not perform layout-versus-schematic verification inside Studio.

5.6 Use of external layout tools

Existing layout-editor integrations and command-line tools remain described in the reference part of this manual. Use them when preparing fabrication artwork or performing a task outside Studio's implemented editing and inspection controls. An ordinary Studio extraction can open the IXI project, inspect its GDS/LDF inputs, run the calculation and inspect supported outputs without opening a separate mesh or field viewer.

Meshes, Fields and Results in Studio

Studio opens supported mesh and field output in the workspace. Numerical results remain in the output files described later in this manual. A view of a model is useful for checking geometry and segmentation, but it cannot establish numerical convergence on its own.

6.1 Read the run result

The IXI **Results** page reports completion status, elapsed time, final Core stage, warning/error counts and output directory. Its **Raw output** action gives access to the calculation output and **Open output folder** reaches the corresponding files. The outer Output / Results panel contains **Output Log**, **Result Files / Summary**, **Diagnostics Center** and **Plots / Tables**.

Use the run log to understand what the solver actually did. During a solver operation Studio reports activity, elapsed time and peak memory; activity is not a percentage of calculation completed. A failed or cancelled operation is reported as such. Results from an earlier run may be marked stale when a new run begins or the source changes.

Read the units and headings in `sol.txt`. Design values, extracted values, absolute differences and percentage differences have the meaning given in the Output Files chapter. An empty or unavailable design value must not be interpreted as zero. Check mesh-quality and singular-value information before accepting the result.

6.2 The mesh viewer

Open a generated `.msh` file through Artifacts, the result list or **File** → **Open**. The same viewer is used for staged project geometry. Loading has an explicit state and can be cancelled. An unsupported or invalid file produces a diagnostic rather than a completed model.

The toolbar provides fit-all, reset-view, surface, surface-and-edge and edge-only display, solid geometry colour, orientation marker, navigation, probe, measure, clipping, copy-probe and snapshot actions. Use the toolbar hints to identify the icons.

Rotate, pan and zoom the model in navigation mode. Use fit-all to restore an overview after inspecting a small region. The orientation marker identifies the axes and provides standard view directions. Display mode changes help reveal internal segmentation and overlapping surfaces.

The **Groups** panel controls visibility of mesh groups. Use **Hide** and **Show** to inspect one part without losing the rest of the model. The **Inspector** exposes selected mesh information. **Provenance** reports the available relationship between displayed geometry and its source; generated field-display geometry does not necessarily have the original mesh-element identity.

Each group row provides **Copy** for its summary, **Isolate**, **Fade** and **Ghost**. The row's display-mode choice can inherit the main mode or use its own surface, surface-and-edge or wire display. Use these controls to distinguish a selected conductor from surrounding geometry; they change the view and do not modify the solver input.

6.2.1 Probe, measure and clip

Select probe mode and a position on the model to inspect the available point or element information. **Copy probe** copies the current probe details. Measure mode is useful for distances between selected positions; read the reported units and coordinates.

Enable Z+ clipping exposes geometry below the clipping plane. **Reset clipping** restores the unclipped view. Clipping and group visibility change the displayed model; they do not change the mesh used by the solver.

6.2.2 Snapshots and display preferences

Export snapshot writes the current canvas view for a report. Choose a useful view and make the required groups visible before exporting. A snapshot is an image, not a mesh input.

Preferences → **Mesh Viewer** contains renderer and display choices, background, axes, lighting, shadows and related settings. The availability of rendering features depends on the selected path and graphics support. The experimental shader heatmap option is separate from the ordinary field displays; leave experimental settings disabled for a normal workflow unless you intend to inspect that feature.

6.3 Field plots

Request field calculation in the IXI project as described in the Field Plotting chapter. Open the resulting VTK output in Studio. The **Fields** panel lists the arrays for the supplied excitation, frequency and real or imaginary part. Select the required array and use **Apply field view**, or select its field row, to activate it.

The field-visualisation controls provide display, component, density, arrow size, range and model-context choices. Once a field view is active these choices update the canvas. An invalid manual range retains the last valid view and reports the problem. **Solid** returns to ordinary geometry.

Heatmap
: Colours a scalar, vector magnitude or selected X, Y or Z component. Use automatic limits initially, then a manual range when comparing views on the same scale.

Directional arrows
: Shows the full vector direction with equal arrow lengths. The selected quantity controls the colour. Arrow size and density control the presentation.

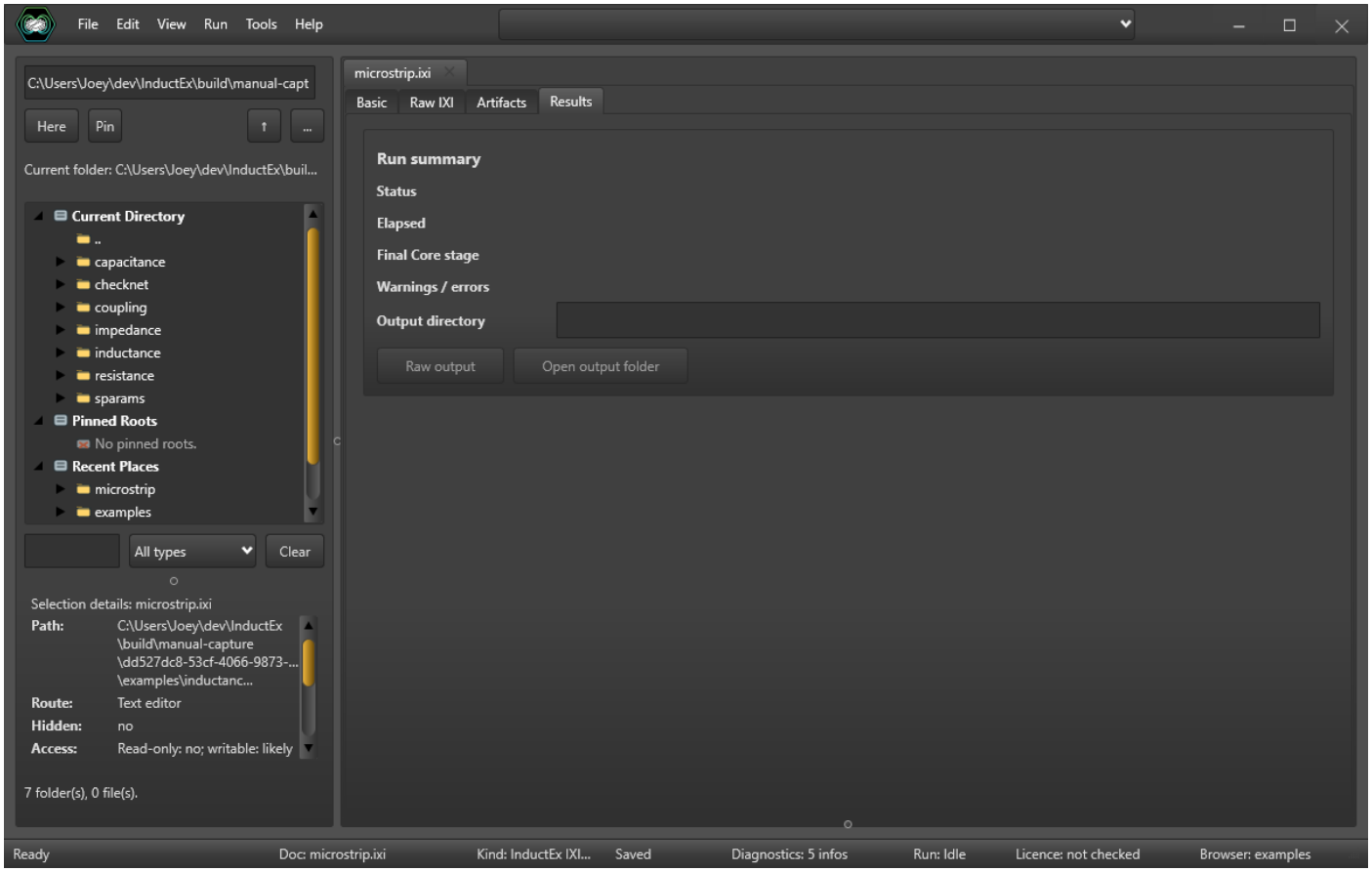

Figure 6.1: The Results page before extraction. A completed run supplies the summary and available result sections.

Streamlines
Traces the selected sampled vector field in both directions. Density controls the seeds used for these traces.

Combined displays
Heatmap with arrows or streamlines retains the field colour while adding direction information.

Show model mesh adds geometric context when the matching model mesh is available beside the output. Rotate the view to check the relation between the sampled field and the model. Snapshot export records the current canvas.

B, H, E, D and J identify magnetic flux density, magnetic field strength, electric field, electric displacement and current density. Studio retains the solver's complete array name. Charge output is identified as charge; do not infer a unit solely from a Q filename.

6.3.1 Interpretation of a field view

Each real or imaginary AC array is displayed separately. Its vector magnitude is not the combined complex amplitude. On a field plane, streamlines follow the tangential projection: a vector normal to the plane can have arrows but no in-plane streamline. Streamlines are spatial traces of sampled data, not moving particles or a new solver calculation.

A volume heatmap initially shows the exterior. Request an IXI `FPlane` when a specific interior section is needed. Directional displays use supported three-component legacy VTK arrays. VTKHDF supports the available scalar heatmap path but does not provide the same directional displays. XML VTK inputs are not interchangeable with legacy ASCII VTK; follow the diagnostic if conversion is required.

6.4 S-Parameter plots

After a successful extraction supplies a recognised S response, Studio adds **S-Parameters** beside **Results**. A recognised S-Parameter JSON file can also be opened directly as a chart document. Other JSON files open as text.

Enter the required traces, for example `S11S21` for a two-port response. Port indices are one-based. Use the colon form, such as `12:3`, when port numbers have multiple digits. Up to 64 traces can be selected. Choose dB, linear magnitude, wrapped phase in degrees, real part or imaginary part. Frequency is displayed in GHz.

The mouse wheel zooms the frequency range around the pointer. Drag to pan the axes. Hovering shows cross-lines, response markers and a tooltip with frequency and up to six trace values. **Reset**, or a double-click, restores the saved band and fits the vertical range. **Fit Y** fits the visible response without changing the frequency range.

Studio can plot a supplied rational model or the sampled response written by Core. A rational model is reevaluated over the visible frequency band when the view changes. A

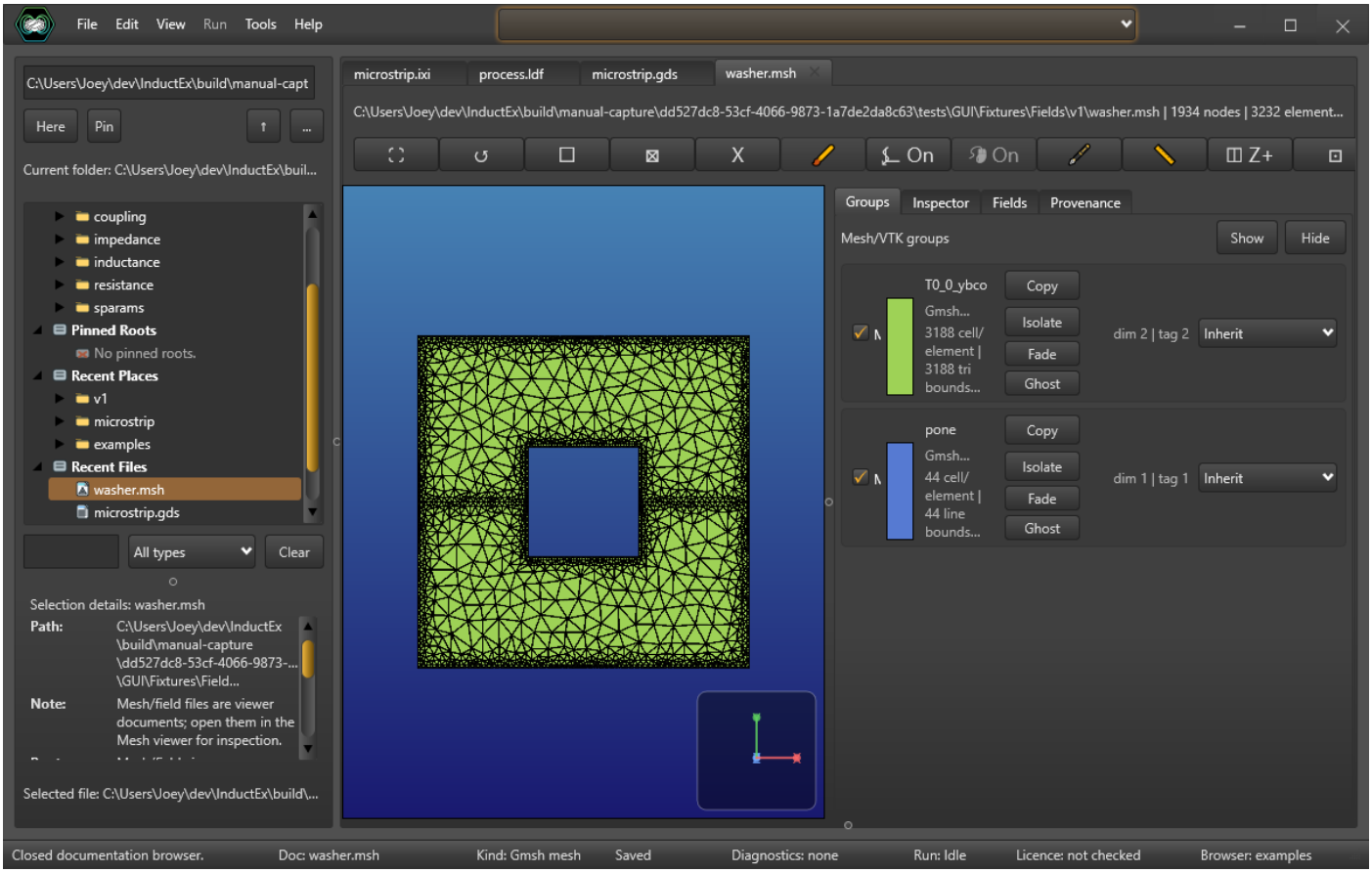

Figure 6.2: Studio's mesh viewer, shown with the mesh from the washer example.

sampled response uses the solved points and complex interpolation; zooming cannot recover an unsolved narrow feature between those frequencies. Read the model/sampled status when interpreting a detailed view. Increasing plot density does not improve the accuracy of the underlying calculation or fit.

Only current output associated with the successful run is loaded automatically. Old, failed, cancelled or ambiguous output is not silently assigned to a new extraction. Open the intended JSON directly when there is more than one applicable model.

Export PNG saves the visible chart and legend without the temporary hover overlay. **Export CSV** saves the selected traces over the visible band as `trace,f_Hz,re,im,valid`. Frequency in that file is in Hz and the complex parts are exported directly. Adaptive traces can have different frequency grids. Invalid samples are identified; wrapped phase jumps are shown as gaps in the plot.

6.5 Keep a calculation record

Keep the IXI file, process and layout inputs, netlist where used, output log and numerical files together. Add a mesh or field snapshot when it helps explain the model. The Support Bundle tool can collect selected saved documents and outputs, but a useful engineering record also needs the assumptions and checks used to judge the numerical result.

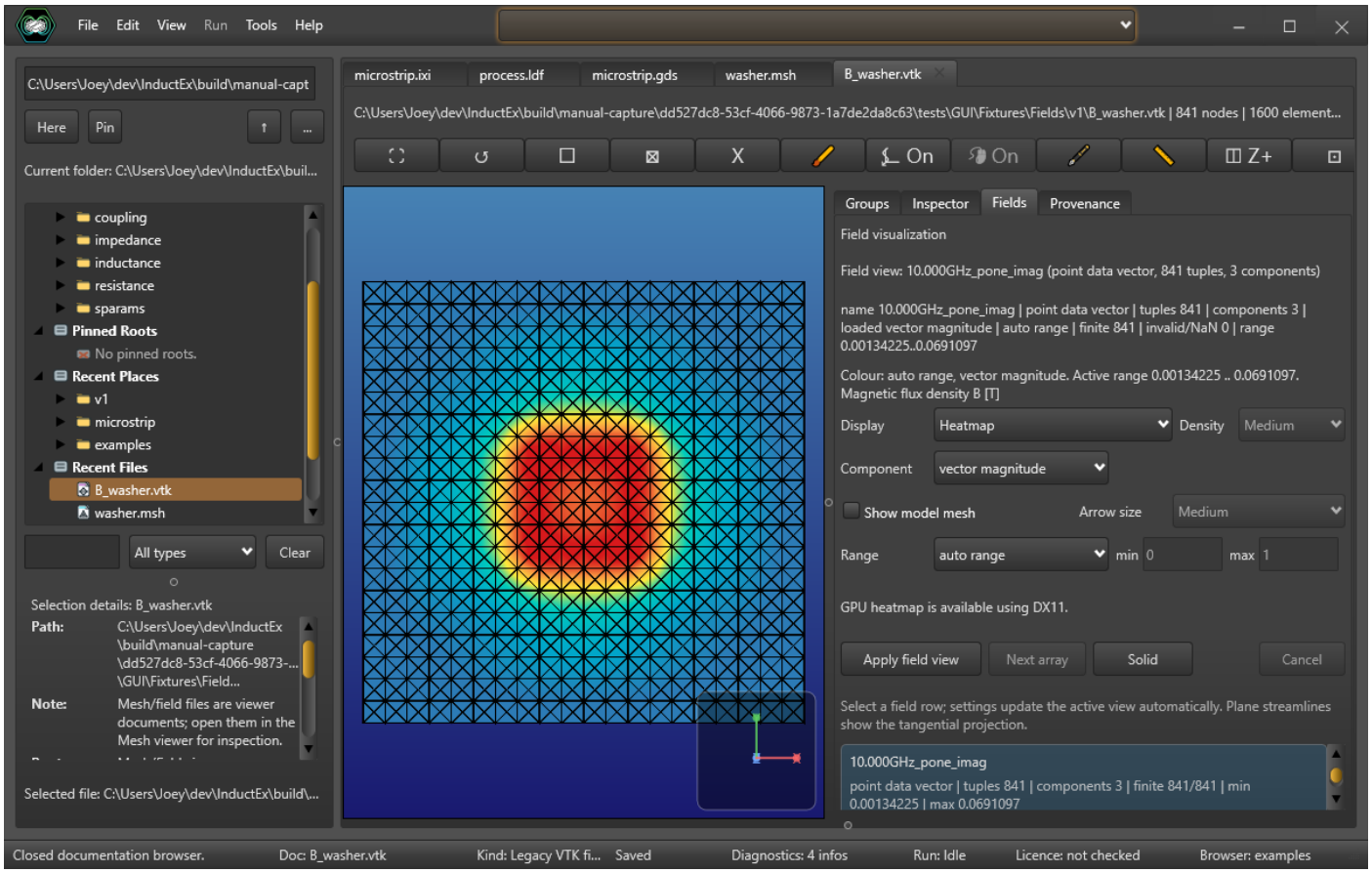

Figure 6.3: Magnetic flux density from the washer example, displayed as a heatmap in Studio.

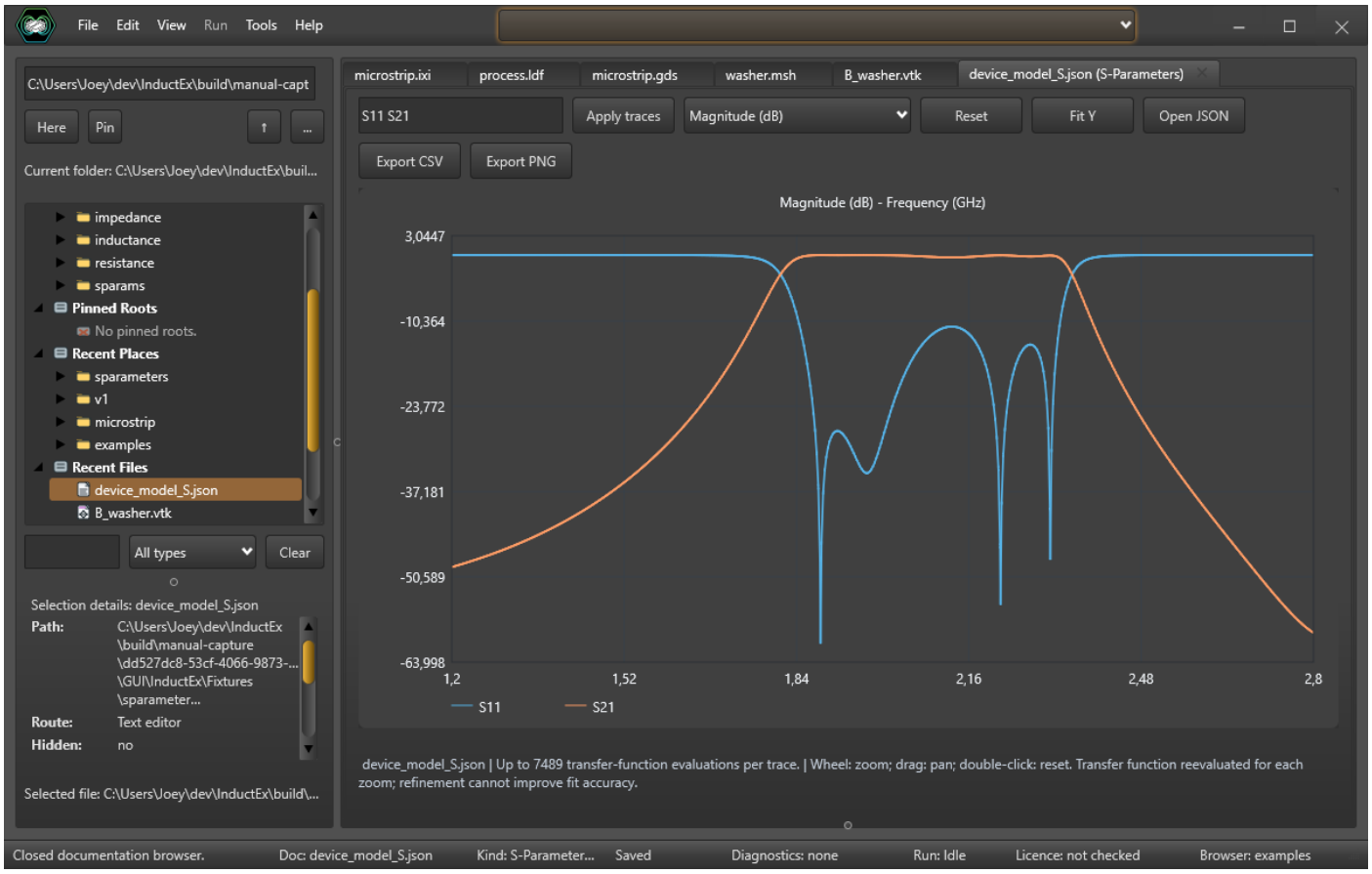

Figure 6.4: The native S-Parameter chart opened from a supplied two-port reference model.

Workspace, Preferences and Help

7.1 Files and folders

File → **Open Folder** changes the file-browser root. Use **View** → **Refresh Browser** after changes made outside Studio and **Show Hidden Files** when necessary. Recent files and places are available under **File**; their clear actions remove those histories.

The file-browser context menu provides the applicable opening, navigation, refresh, path-copy and pinning actions. Pinned folders help return to working directories. Opening a file that is already in the workspace focuses its existing document tab.

Use **Close Tab** or **Close All** to close documents. Save decisions apply to changed editable documents. **Preferences** → **General** controls restoration of the previous workspace/session at startup. Restoration of a view does not rerun an extraction or verify that an old output is current.

7.2 Find text, commands and help

Edit → **Find** and **Replace** operate on the active editor. **Workspace Text Search** searches document text in the workspace. Use this for a parameter, layer name or port declaration that may be in another input.

The workspace text-search dialog accepts a text query, filename filter, file-type filter and case-sensitive matching. Select open documents and, when needed, supply filesystem roots. The search runs in the background and can be cancelled with **Cancel Search**. Select a hit and use **Open Result** to navigate to it. **Replace preview** shows the proposed replacement in the result; it is a preview and does not perform a replacement across all files.

Tools → **Global Workspace Search** searches open tabs, filenames, commands, help pages and diagnostics. Select a result to read its details, then use **Open**. The **Files**, **Commands**, **Help** and **Diagnostics** controls restrict the kind of result. **Open Documentation** reaches the local help browser.

Tools → **Command Palette** finds and runs available commands by name. Its availability follows the current document and application state, just as a menu command does. Use the command centre in the window header for the related command-search workflow.

7.3 Diagnostics

Open **View** → **Output / Results** and select **Diagnostics Center**. Filter by current document and severity using **Info**, **Warn**, **Error** and **Fatal**. Select a diagnostic to inspect its message and use its source navigation where available. **Copy row** and **Copy summary** collect diagnostic text for a report.

Editor diagnostics and calculation messages serve different purposes. A valid IXI record can still describe an incorrect model, and a solver warning may only appear after modelling or extraction. Read both the current source diagnostics and the complete output log.

7.4 Workspace Dashboard

Tools → **Workspace Dashboard** gathers workspace, diagnostics, licence and path status. Use **Refresh** after correcting a reported problem and **Copy Summary** to retain the checklist. Its checks assist preparation and recovery; they do not replace inspection of the physical model or numerical result.

7.5 Import / Export Center

Tools → **Import / Export Center** lists tasks applicable to the active document. Select a task to read its detail and whether it is ready or unavailable. **Run Selected** starts the chosen action; choose a destination in its dialog.

The centre provides GDS and mesh snapshot/report export, validated GDS-copy export, support bundles, settings import/export, keymap import/export and export of selected editor text/log lines. Snapshot export produces a PNG and support report. Selected-text export writes a linear selection as UTF-8; rectangular and multiple-cursor selections are not supported by that task. Invalid or conflicting keymap entries are skipped and reported, while accepted entries are saved to preferences.

Export tasks are document-dependent. A PNG export records a view, while an exported data file retains the values or geometry described by its format. Read the selected task's explanation before using the output in another application. The LDF technology converter has its own Tools command; S-Parameter data export is provided on the chart.

7.6 Preferences

Use **Tools** → **Preferences**. **OK** accepts the preferences and **Cancel** leaves the dialog. The pages provide the following groups of settings:

General
: Light/dark theme, custom window chrome, previous-session restoration, licence file and daemon paths. Custom-chrome changes require a restart.

Health
: Configuration and session paths and the available information about the current setup. Use this when checking where Studio stores its state.

Editor
: Font and size, whitespace and line-ending markers,

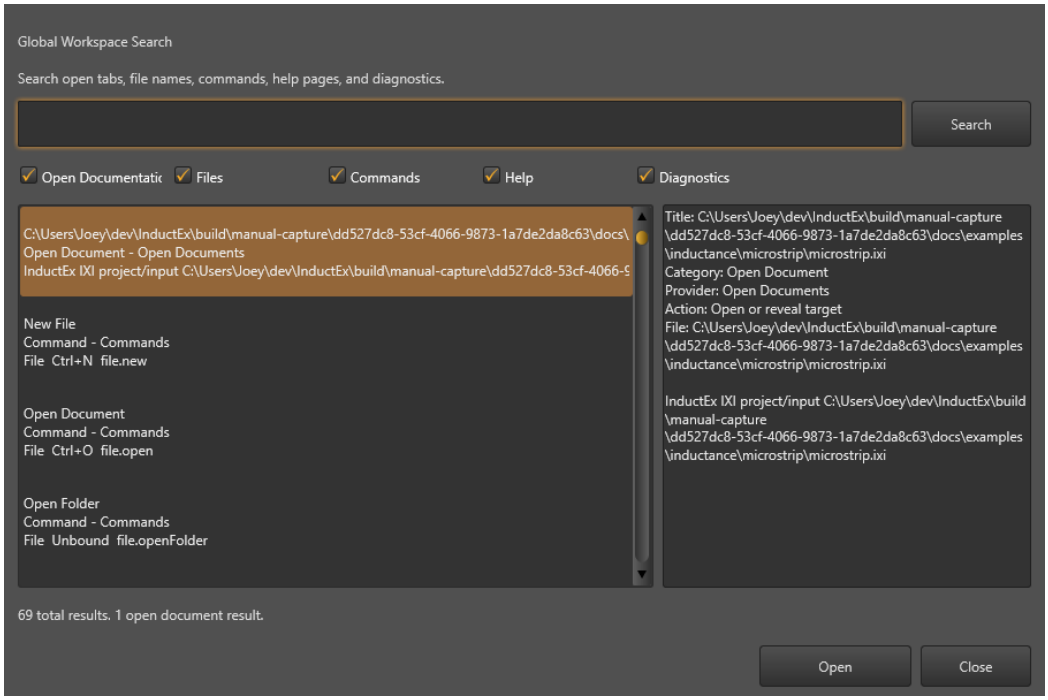

Figure 7.1: Global Workspace Search provides files, commands, help and diagnostic destinations.

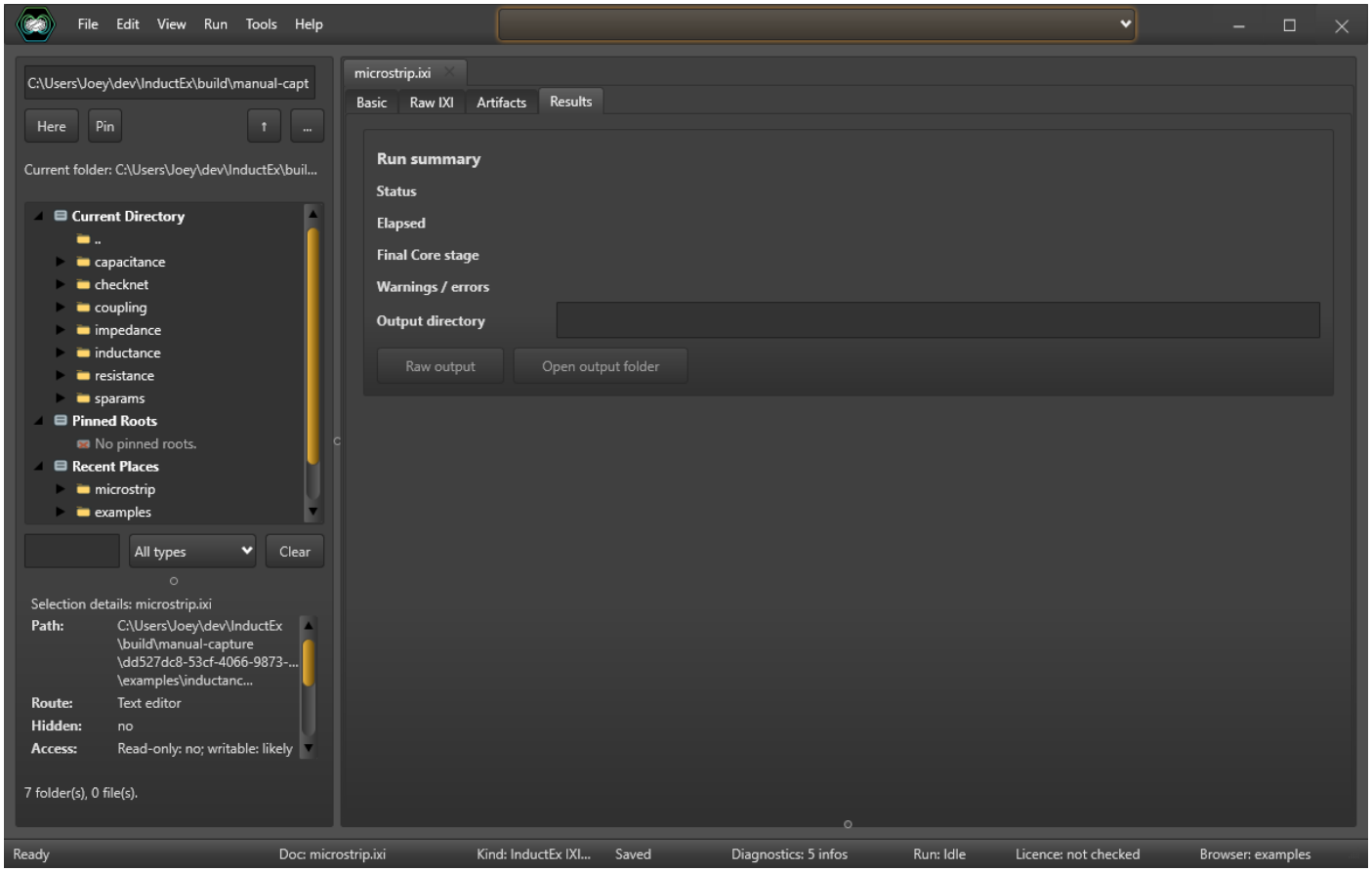

Figure 7.2: The Command Palette action focuses command search in the custom window header.

overview ruler, indentation and tab width, diagnostics, completion and supported wrapping. Wrapping is applied where supported by the editor.

File Browser

Browser root, hidden-file visibility and the recent-item limit.

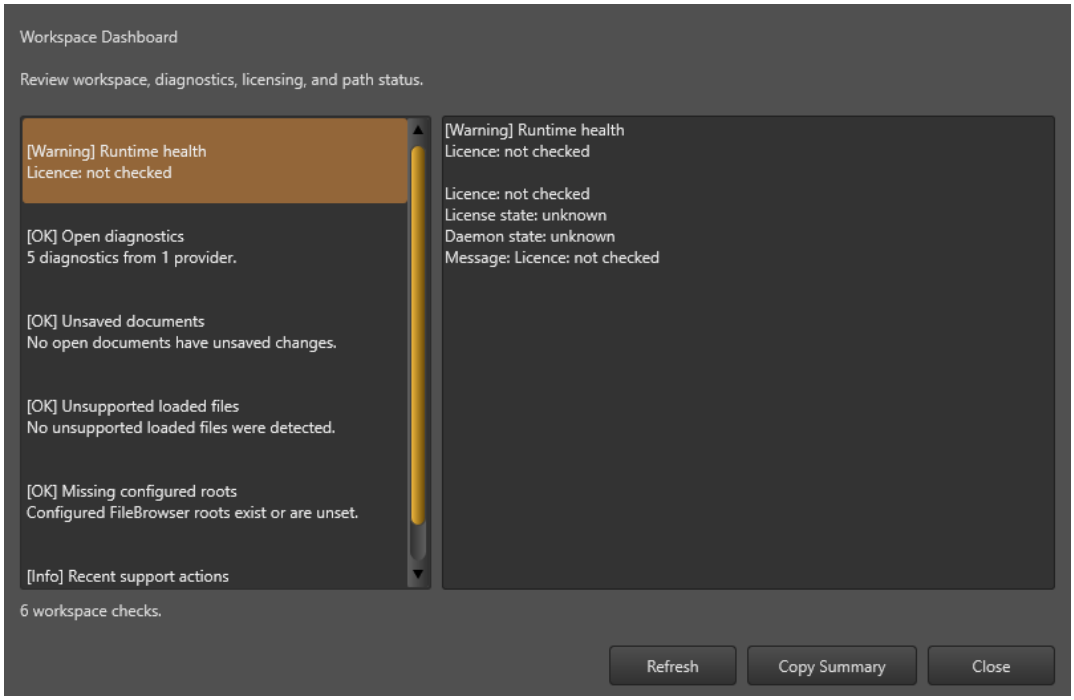

Figure 7.3: The Workspace Dashboard gathers the current preparation and path checks.

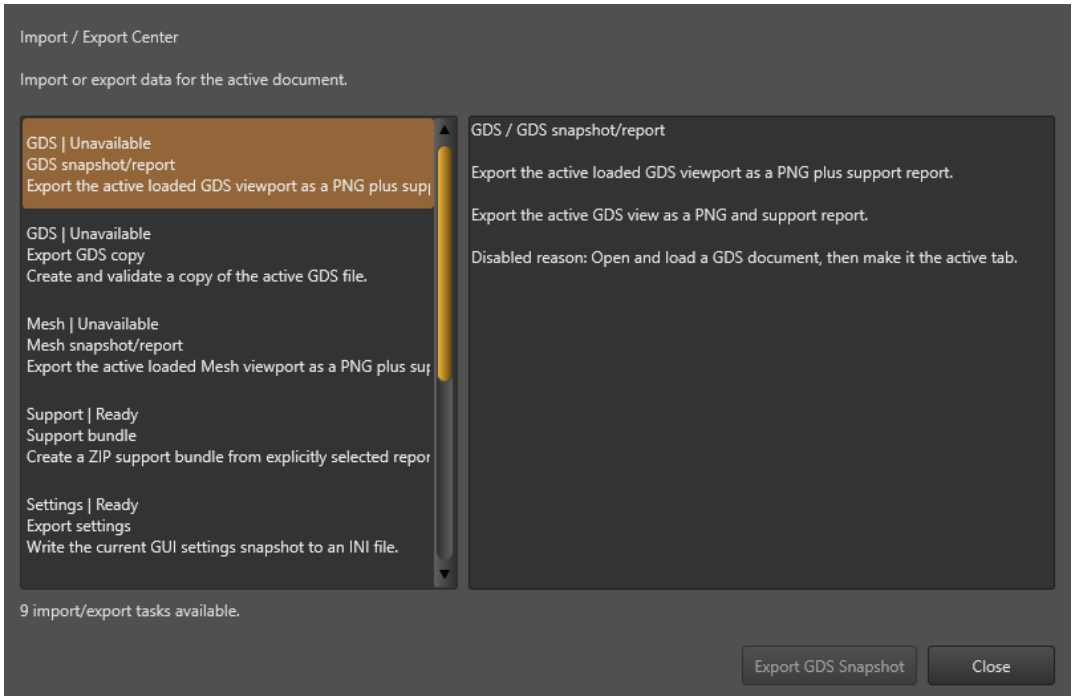

Figure 7.4: Import / Export Center reports the tasks available for the active document.

GDS Viewer
Grid and crosshair defaults, grid spacing, GPU rendering choice, layer isolation and text display settings.

Mesh Viewer
Renderer and display choices, axes, backgrounds, lighting, shadows and other supported mesh-view options.

Keybindings
Command shortcuts, reset by selection/category or all, and keymap import/export. Select a command and use **Apply** after editing its shortcut.

Repair
Path validation, configuration backup/restore, reset settings, reset session, clear recents and disable custom chrome. Read the pending operations before accepting a repair.

Use the light and dark theme commands under **View** → **Theme** for direct theme changes. Plot, editor and view controls follow the selected theme. Keybindings may be customised; when a shortcut differs from an instruction,

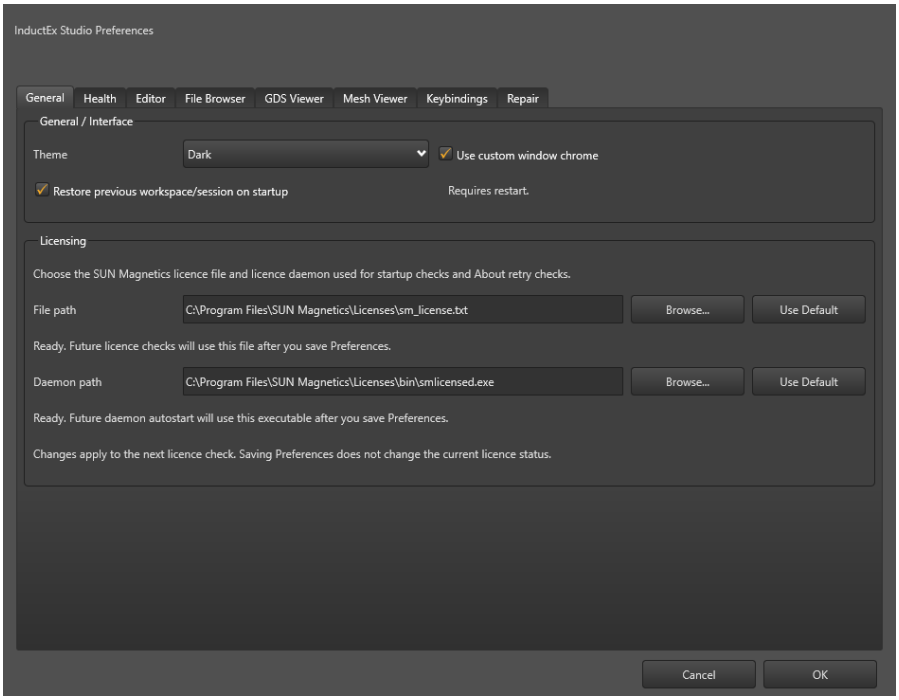

Figure 7.5: Studio preferences group interface, editor, browser, viewer and recovery settings.

use the named menu command or inspect the Keybindings page.

Resetting settings or the session changes saved Studio state. Export settings before a reset when you want to retain them. A reset is not a substitute for saving document edits, and settings import should be followed by checking the paths on the current computer.

7.7 Documentation and examples

Help → **Documentation Browser** opens the local documentation. Search for a topic and follow its entries; **Open User Manual PDF** opens the installed PDF. This browser uses the documentation shipped with Studio, so it does not require an online manual for normal use.

Help → **Examples / Templates** lists the supplied example projects. Inspect the selected entry and copy it to a working directory with **Copy to Folder**. Copy the associated files as well as the IXI project. An example that refers to a process file or layout cannot be made self-contained by copying only its project text.

7.8 About and licence status

Help → **About** shows Studio and Core build information, company/copyright details and the available licence/daemon status. **Retry Licence Check** repeats the check after correcting the setup. **Generate System ID** assists licence requests and requires the installed daemon. Preserve the complete identifier when providing it to SUN Magnetics.

7.9 Diagnostic log and support bundle

Help → **Diagnostic Log** opens the application diagnostic log. This is useful for startup, configuration and view problems. Retain the extraction Output Log separately when the question concerns a calculation.

Help → **Support Bundle** collects selected saved open-document files and optionally output/log files into a ZIP archive. Review the selection, add the required output/log files and choose **Export ZIP**. Unsaved edits are not a saved input file; save a working copy first when those edits are needed to reproduce a problem. Creating the archive does not send it to support.

7.10 Common problems

Run is unavailable
Select an IXI project, read Check and run, resolve the reported source/input problems and check the licence. A GDS, LDF, result or ordinary text tab is not itself an extraction project.

An input cannot be found
Check its path relative to the saved IXI file. Copying the whole example directory normally preserves its references.

The process view has not changed
Read the stale-snapshot indication and fix current LDF errors. A last-valid view is explicitly identified.

A field view is empty
Check the selected array and component, manual range, group visibility and whether the requested directional display is supported by that file. A normal vector on a plane need not create streamlines.

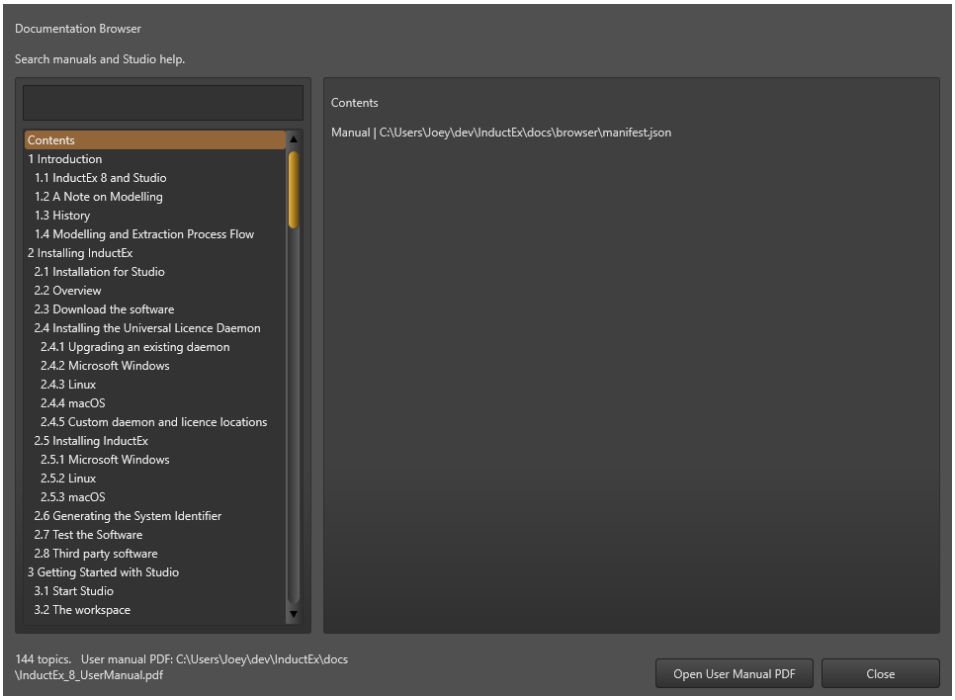

Figure 7.6: Documentation Browser provides the local manual topics and the PDF action.

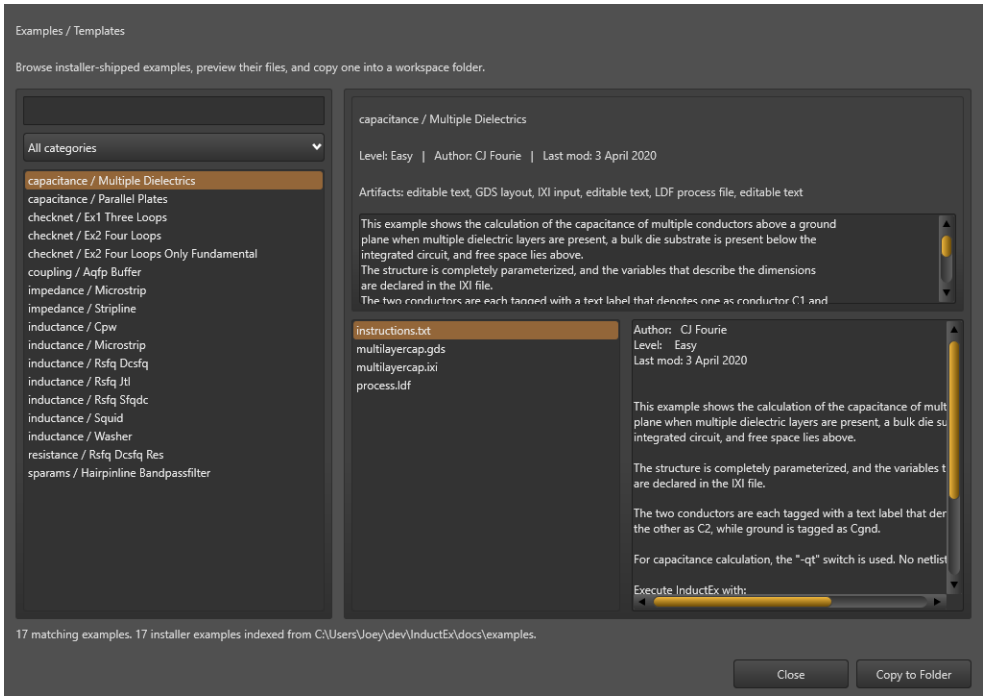

Figure 7.7: Examples / Templates lets you select a supplied project and copy it to a working folder.

No S-Parameter page appears
Check successful completion and whether a current supported S response was written. Open the intended JSON directly if automatic selection is ambiguous.

A panel is too small or hidden
Restore it from View or its splitter, or use Reset Layout. At narrow widths inspect the alternative project/source pages.

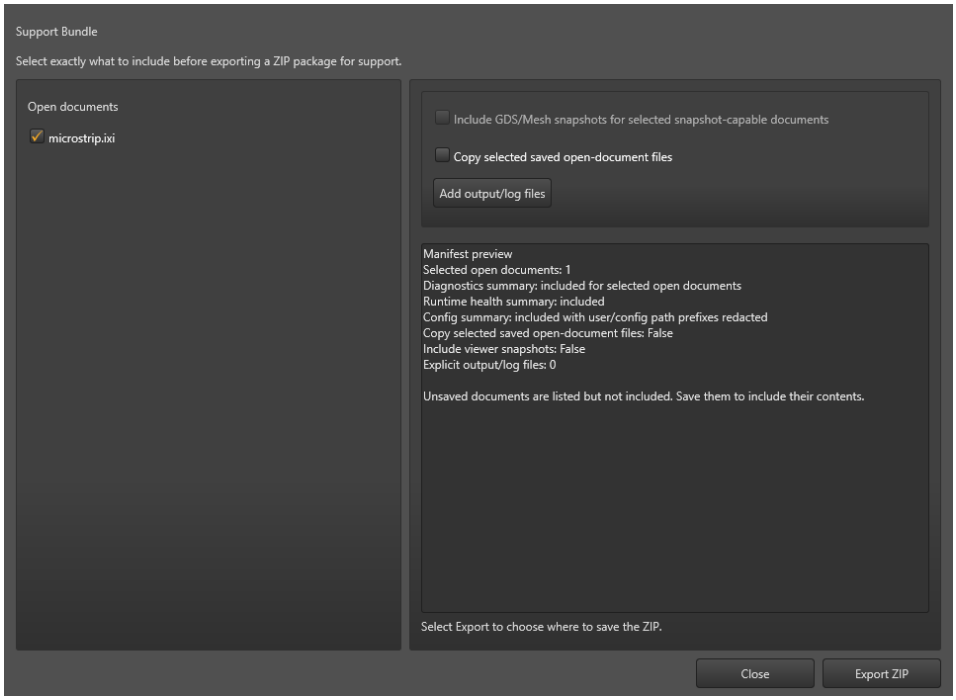

Figure 7.8: Support Bundle lets you select saved documents and output/log files before writing a ZIP archive.

II Input and Modelling Reference

8 Geometry Entry 38

9 Process File 39

10 Layout Input 46

11 Circuit Netlist 58

12 Ports 60

13 Off-Chip Structures 63

14 Output Files 65

15 Field Plotting 67

Geometry Entry

8.1 Drawing volume

InductEx accepts object geometry entry with precision down to the drawing unit, which is $1\,\mathrm{nm}$. Any coordinate value smaller than the drawing unit is rounded to the nearest integer.

The maximum drawing volume is a cube, with each side limited to $1\,000\,000\,000$ drawing units, centred around the origin (0, 0, 0). This limits the drawing volume to a cube with a volume of $1\,\mathrm{m}^3$.

If a GDS file is used to enter layout geometry, and the drawing unit is not $1\,\mathrm{nm}$, *InductEx* will translate the geometry correctly, but will internally rescale all coordinates to map to $1\,\mathrm{nm}$ drawing units. Smaller dimensions cannot be entered in this version of *InductEx* , due to the way that polygon triangulation works.

Process File

9.1 Use of the Process File

InductEx is primarily aimed at layer-based devices such as integrated circuit dies or printed circuit board structures (or a combination of these), which are usually defined as a set of layer mask layouts.

The process file describes to *InductEx* how a layout drawing (which is essentially a set of disconnected two-dimensional objects) should be modelled in three dimensions with the correct layer sequence and connections. As such, it describes the fabrication process, with the important caveat that the **description is focused on creating numerical models that have the same geometry as fabricated circuit structures** but may differ (substantially) from the actual fabrication sequence.

The **process file** is thus also referred to as the **layer definition file** or the **technology file**. It uses the file extension "`.ldf`".

Without the process file, *InductEx* has no information with which to build a three-dimensional calculation model from a layer-based two-dimensional layout.

The process file controls or configures *InductEx* by defining both the modelling and process technology parameters used to build extraction models.

Modelling parameters are not derived from the actual fabrication process. These parameters merely control segment size, ground plane modelling, optional segment blanking, etc.

Process technology parameters describe the actual layers in terms of model construction order (which is not necessarily the same as the fabrication order), dimensions, mask-to-wafer bias and physical parameters such as penetration depth.

During model construction, every layout input file must be assigned a corresponding layer definition file.

Standard layer definition files for superconductor integrated circuit processes of which the design rules are in the public domain are available on the SUN Magnetics website.

Layer definition files for some other processes are available to registered users.

9.2 Open and Protected Process Files

The standard process file is text-based, so that it can be edited with ease as a process is developed or adjusted. Many users also prefer to edit process parameters for specific layouts, which is easily don in the text-based file. This file format is openly accessible, and all process parameters are exposed to the end user.

For cases where the fabrication service provider requires the details of the fabrication steps, calibration parameters and material properties to be obscured from the end user, the process file can be protected. *InductEx* 8 requires a password, provided during execution, to read a protected file. Even with a password, end users cannot make changes to the protected file's process parameters.

9.3 File structure

9.3.1 Comments

A comment line starts with the asterix character, "*", and *InductEx* will skip over such lines when processing a process file.

All text after a double frontslash "//" character combination is also discarded as comments and not processed.

9.3.2 File header

For legacy support, an open process file does not require a header. However, to future-proof process files with a long life expectancy, it is best to add a header. All protected files require a header.

Header lines must start with the semi-colon character, ";", and the first line for a protected process file must be:

```
;SUNMAGNETICS-LDF-PROTECTED=:1
```

Other header lines can be provided in any order, provided that all header lines are included for a protected file. *InductEx* stops processing the header at the first line that does not start with a semi-colon.

All headers are provided as

`;field=`*value*

The complete list of header fields is:

container_version
(*Integer*)
The version of the process file format.

format
(*String; default = ldf-open*
Options are **ldf-open** and **ldf-protected**.

issuer
(*String*)
This holds the name of the entity that packed and issued the protected file. This can be the fabrication house/entity, or SUN Magnetics.

issuer_id
(*String*)
An ID code for the issuer, provided by SUN Magnetics.

process_owner
(*String*)
The fabrication entity.

process_id
(*String*)
The short name or ID for the fabrication process.

process_rev
(*String*)
The revision of the process described in this process file.

process_family
(*String*)
Some information about the process family; usually the junction technology (such as Nb/AlOx/Nb).

issue_date
(*Date String*; *format yyyy/mm/dd*)
The date on which the process file was created.

expiry_date
(*Date String*; *format yyyy/mm/dd*)
The date after which the process file becomes invalid. If provided for an open process file, *InductEx* will issue a warning but continue. The expiry date is mandatory for a protected file, and *InductEx* will not open an expired protected file.

calibration_id
(*String*; default = N/A)
An ID code for the calibration procedure, if applicable.

calibration_summary
(*String*; default = N/A)
A single line that summarises main calibration information.

calibration_dataset
(*String*; default = N/A)
A description of the calibration dataset used, if applicable.

min_inductex_version
(*String*; default = 8.0.0)
The minimum version of *InductEx* required to open this file if it is protected.

cipher_nonce
(*Hexadecimal String*)
The initialisation vector for decryption of a protected file.

9.3.3 Record blocks

The process file () contains a parameter record, layer records to define every fabrication step or layer, and optional mask and operator records.

Every record starts with

```
$Recordname
```

and ends with

```
$End
```

Each record contains a number of fields and values, which are entered as:

`field =` *value*

The field/value entry is discussed in section 9.3.4.

The list of records are:

Parameters
Global parameters, of which the fields are described in section 9.3.4.

Layer
Mandatory layer definition, of which the fields are described in section 9.3.6. Any layout structures defined on a layer that is not defined with a layer record are ignored by *InductEx* .

MeshMaskLayer
Mesh masks, to control segment size by position rather than by layer.

Operators
Specific operators that may be added to layouts to control modelling. These operators, if used effectively, can reduce segment count and solution time, but need to be used with care. Operator parameters are listed in section 9.3.8.

9.3.4 LDF parameter entry

Global or layer parameters, or parameters for operators, are entered as field/value pairs in the format

```
fieldname = value
```

The `value` can be an integer or floating point number, a string (with no spaces), a boolean value (`TRUE` or `FALSE`), or an array of entries. When `value` is an array, the array entries are separated by spaces and the array is enclosed in square brackets, for example:

```
LayerADD = [10 15 22]
```

```
EdgeSlice = [9E-2 2E-1 0.5]
```

Real values can be specified conditionally with regards to the mesh type. The syntax is:

`fieldname =` *defaultvalue* { MeshType = *value* ; MeshType = *value* }

The mesh type can be `CUB`, `TRI`, `TET` or `HYBRID`. This is useful for calibration across mesh types. Examples are:

```
Lambda = 0.09 { TET = 0.08 ; TRI = 0.93 }
```

```
Sigma = 5 { CUB = 5.5 }
```

9.3.5 Process file global parameters

AbsMin (*Floating point*; *Default = 0.01*)
Sets the radius of a test sphere around a current density

node within which it is mapped to a geometry node when filaments (cuboid meshes) are used. The value must **not** be zero).

BlankAllCutsGP (*Boolean; Default* = False)
If True, BlankAllLayer also blanks the ground plane.

BlankAllLayer (*Integer*)
Sets the optional global blanking layer number. No objects or parts of objects falling within blanking layer objects are segmented, which allows user-defined model trimming.

BlankXLayer (*Integer*)
Sets the optional *x*-direction blanking layer number. Any objects or parts of objects falling within *x*-blanking layer objects are only segmented in the *y* direction. Applicable to cuboid filament meshes only.

BlankYLayer (*Integer*)
Sets the optional *y*-direction blanking layer number. Any objects or parts of objects falling within *y*-blanking layer objects are only segmented in the *x* direction. Applicable to cuboid filament meshes only.

ChipEdgeLayer (*Integer*)
Sets the optional chip edge layer number. A chip edge is determined from the outer edges of the union of all objects on this layer, and is used to delimit dielectric boundary layers and build chip edges for chip-scale extraction.

CornerRoundRadius (*Floating point; Default = 0*)
Sets the global radius in Units of circular rounding at all layout object corners. If the value is zero, no rounding is applied.

CornerRoundSteps (*Integer; Default = 2*)
The number of segments created at a layout object corner when rounding is applied.

CropGP (*Boolean; Default* = True)
Boolean value to indicate that negative mask ground planes should be cropped to within GPOverHang of the union of all non-ground plane structures (this reduces unnecessary segments). It does not affect positive mask ground planes.

DataTypeNotZero (*Boolean; Default* = False)
If True, GDSII objects with DataType <> 0 are also accepted for model building. Note: this will be discontinued in a future release, as DataType is earmarked for support functionality and not for geometry.

DecimationDistance (*Floating point; Default = Units/100*)
Sets the Hausdorff distance (or epsilon), in Units, used by the Ramer-Douglas-Peucker polygon boundary decimation algorithm if PolyDecimation is True.

DielectricOverhang (*Floating point; Default = GPOverHang*)
Sets the distance in Units between the edge of any object and the edge of a rectangular plane that describes the outline of a meshed dielectric interface plane (for EQS and EMQS solutions). For EMQS solutions, if any ground boundary is specified, DielectricOverhang is set to 0.

EdgeSeg (*Floating point; Default = 0*)
Optional parameter that defines the width in Units of a slice inside the boundary of every conducting object to allow fine segmentation near conductor edges. This parameter is disregarded when a cuboid mesh is used.

EdgeThickFraction (*Floating point; Default = 1*)
Sets the top edge rounding any layer object through a reduction in the thickness at the edge. The value is range limited between 0.1 and 2.0.

EpsilonR (*Floating point; Default = 1*)
Optional parameter that defines the relative dielectric constant, or relative permittivity, of isolation layers.

EpsilonRAbove (*Floating point; Default = 1*)
Optional parameter that defines the relative dielectric constant, or relative permittivity, of the global environment above the last manufactured layer on the wafer and outside of the chip edge. The default is 1 for free space.

EpsilonRBelow (*Floating point; Default = 11.7*)
Optional parameter that defines the relative dielectric constant, or relative permittivity, of the global environment below the lowest layer on the wafer (essentially the substrate). The default is 11.7 for silicon at 300 K.

ExtractBoundLayer (*Integer*)
Sets the optional extraction boundary layer (according to its GDS layer number). If a polygon exists on this layer, all other layout objects will be clipped to its boundaries to limit the size of an extraction model.

FieldCageScale (*Floating point; Default = 2.0*)
Sets the scaling of the cuboid cage around a meshed model in which electric or magnetic field is calculated for visualisation purposes. The longest dimension of the meshed model in any Cartesian axis is designated unity, and the minimum scaling value that *InductEx* accepts is 1.1.

FieldCageZScale (*Floating point; Default = 1.0*)
Sets the *z*-axis scaling of the cuboid cage around a meshed model in which electric or magnetic field is calculated for visualisation purposes. FieldCageScale is designated unity. The minimum scaling value that *InductEx* accepts is 0.01.

FieldMeshSize (*Floating point; Default = 2.0*)
Sets the size of mesh elements on the outside of the cubical field calculation cage in Units.

Frequency (*Floating point; Default = 10E9*)
The frequency of the port excitation sources. For purely superconductive circuits at frequencies below the gap frequency, this does not influence the results. With resistive components, the frequency dependent skin depth affects current distribution in the normal conductors.

GPLayer (*Integer*)
Indicates the GDS layer number of the ground plane, which is used to fill negative-mask ground planes to the defined boundary.

GPOverHang (*Floating point; Default = 2.5*)
Sets the distance in Units between the edge of any object and the edge of the ground plane that is generated to cover the smallest necessary area beneath structures. A smaller value results in artificially higher calculation results. 2.5 is a good value for all processes, but 7.5 is recommended when lines crossing holes in the ground plane are investigated (the ground plane automatically wraps around holes). This parameter is disregarded for positive mask ground planes.

GridSnapOn (*Boolean; Default* = False)
If True, snaps all layout object vertices (after polygon union and bias adjustments) to a grid defined by SnapSize.

HFilaments (*Integer; Default = 1*)
The global value for the number of height filaments into which segments are divided. Can be overrided per layer. Applicable to cuboid filament meshes only.

Lambda (*Floating point; Default = 0.09*)
Global value for London penetration depth. Can be overrided per layer. If Lambda is defined for a layer, Sigma and MuR are ignored.

Lambda0 (*Floating point; Default = 0.09*)
Global value for the London penetration depth at 0 K if temperature is specified. Can be overrided per layer.

LambdaFit (*Floating point; Default = 4*)
Exponent for temperature dependent London penetration depth.

LossTangent (*Floating point; Default = 0.0*)
Global value for dielectric loss tangent. Can be overrided per layer.

MeshMaxSizeFactor (*Floating point; Default = 5*)
Increase in minimum side length for a "good" meshed triangle from the SegmentSize value applied to polygon edges. A larger value results in fewer (larger) mesh elements further away from polygon edges, which reduces segment count for large structures such as ground or skyplanes. The lower value is limited to 1.0.

MeshMinAngle (*Floating point; Default = 25*)
Minimum corner angle in degrees for a "good" meshed triangle. A larger value results in more refinement steps. The value is range limited between 10 and 40.

PolyDecimation (*Boolean; Default* = True)
If True, decimates the boundary of a polygon with the Ramer-Douglas-Peucker algorithm to find a similar polygon with fewer points. The dissimilar distance is defined with DecimationDistance.

ProcessHasGroundPlane (*Boolean; Default* = True)
Boolean value to indicate ground plane in process. Usually only False for monolayer HTS circuits.

Relief (*Boolean; Default* = True)
If False, *InductEx* does not adjust mesh elements for tetrahedral and triangular meshes to reflect relief of structures overlapping other structures in unplanarised layers below.

ResZero (*Floating point; Default = 1E-3*)
Sets the smallest calculated resistance value printed to the output – below this the resistance is considered zero and an element is considered to be purely inductive.

SegmentSize (*Floating point; Default = 1.0*)
Sets the maximum side length of segment in Units for a cuboid mesh, or the maximum side length of segments on the edge of a polygon for triangular and tetrahedron meshes. Larger values speed up calculations but cause higher inductance results. Always compare calculations against a known result after you adjust SegmentSize. (Replaces old parameter “GapMax”).

SheetCurrents (*Boolean; Default* = True)
Boolean value to set the use of sheet current models in conducting planar structures when *TetraHenry* is used. Set False to model plane structures as volumes rather than sheets.

Sigma (*Floating point; Default = 10*)
Bulk conductivity of resistive layers (global). Calculated as the inverse of resistance per square times layer thickness for thin films, and expressed as $\Omega^{-1}\text{Units}^{-1}$. For example, if unit size is in μm, and layer thickness is 80 nm, a conductivity of $10/(\Omega\,\mu\text{m})$ equals a sheet resistance of $1.25\,\Omega/\square$.

SnapDNewVertex (*Floating point; Default = 0.001*)
Sets the minimum perpendicular distance in Units between a vertex on a via and an edge on a conductor polygon below which the conductor polygon edge snaps to the via and becomes two edges.

SnapDPortToPolyEdge (*Floating point; Default = 0.01*)
Sets the minimum perpendicular distance in Units between a vertex on a terminal polygon and an edge on a conductor polygon below which the conductor polygon edge snaps to the terminal polygon and becomes two edges.

SnapSize (*Floating point; Default = AbsMin*)
Sets the grid size to which object vertices are snapped if GridSnapOn is True.

TC (*Floating point; Default = 9.2*)
Global critical temperature in kelvins for transition to superconductivity. Can be overrided per layer.

Temperature (*Floating point*)
Global temperature. The default is not set. If temperature is defined, all London penetration depth values are calculated from $\lambda(T) = \lambda(0)[1-(T/T_c)^P]^{(-1/2)}$, where P is the defined by LambdaFit.

TerminalInRange (*Floating point; Default = 1*)
Sets the distance (in Units) from the rectangular boundary around any terminal object within which a text label will be accepted as “linked” to that object.

TermLayer (*Integer*)
Indicates the GDS layer number of the layer on which terminals are drawn.

TextLayer (*Integer*)
Sets the text layer (according to its GDS layer number).

UnitC (*Floating point; Default = 1.0*)
Sets a unit size in Farad to which all capacitance results are normalized. Capacitance results are divided by the value of UnitC after calculation. If UnitC = 1E-15, then results are displayed in fF.

UnitI (*Floating point; Default = 1.0*)
Sets a unit size in Ampere to which all current results are normalized. Current results are divided by the value of UnitI after calculation.

UnitC (*Floating point; Default = 1.0*)
Sets a unit size in Farad to which all capacitance results are normalized. Capacitance results are divided by the value of UnitC after calculation. If UnitC = 1E-12, then results are displayed in pF.

UnitL (*Floating point; Default = 1.0*)
Sets a unit size in Henry to which all inductance (self and mutual) results are normalized. Inductance results are divided by the value of UnitL after calculation. As an example, inductance results can be normalized to PSCAN

dimensionless units by setting UnitL = 2.64E-12. If UnitL = 1E-6, then results are displayed in μH, while UnitL = 1E-12 leads to results in pH.

UnitR (*Floating point; Default = 1.0*)
Sets a unit size in Ohm to which all resistance results are normalized. Resistance results are divided by the value of UnitR after calculation.

Units (*Floating point; Default = 1E-6*)
The unit size in meters in the layer definition file. The default is 1 μm.

VirtualGap (*Floating point; Default = 0.2*SegmentSize*)
Size of the gap cut by a virtual cut terminal when cuboid filaments are used. The default is SegmentSize since version 5.06. It was fixed at AbsMin in earlier versions.

9.3.6 Layer parameters

Bias (*Floating point; Default = 0*)
Defines the mask-wafer offset for the layer. The value is added to the boundaries of any object on the layer (positive values "grow" the object, while negative values "shrink" the object). For processes with zero mask-wafer offset, Bias is still available as a calibration parameter to fit extractions to measurements. Bias is not supported when tetrahedral modelling and the *TetraHenry* engine is used.

CornerRoundRadius (*Floating point; Default = Global CornerRoundRadius*)
Sets the radius in Units of circular rounding on this layer at layout object corners. If the value is zero, no rounding is applied.

CornerRoundSteps (*Integer; Default = Global CornerRoundSteps*)
The number of segments created at a layout object corner on this layer when rounding is applied.

DecimationDistance (*Floating point; Default = Units/100*)
Sets the Hausdorff distance (or epsilon), in Units, used by the Ramer-Douglas-Peucker polygon boundary decimation algorithm for this layer if PolyDecimation is True.

EdgeSeg (*Floating point; Default = Global EdgeSeg or Layer Lambda*)
Optional parameter that defines the width in Units of a slice inside the boundary of every conducting object to allow fine segmentation near conductor edges. This parameter is disregarded when a cuboid mesh is used. If not defined for the layer, the global value is used if the layer is resistive, while the layer penetration depth is used if the layer is superconductive.

EdgeSlice
(*Array of Float*)] Optional parameter that sets the Width in Units of consecutive slices from the boundary of every object on a layer. It has the same function as EdgeSeg, but overrides the single-valued EdgeSeg when specified. This parameter is disregarded when a cuboid mesh is used. Primary application is for higher precision charge distribution when large capacitive structures are analysed.

EdgeThickFraction (*Floating point; Default = Global EdgeThickFraction*)
Sets the top edge rounding any object on this layer through a reduction in the thickness at the edge. The value is range limited between 0.1 and 2.0.

EpsilonR (*Floating point; Default = 4.6*)
Defines the relative dielectric constant, or relative permittivity.

Filmtype (*Character; Default = A*)
Sets the film type of the layer. The options are:

A
Auxiliary layer.

C
Cut layer (everything underneath ablated away).

I Isolation layer.

N
Normal (resistive) layer that is not segmented (used for instance for superconductive circuits where only inductance is extracted, and resistive components should be ignored during calculation).

R
Resistive layer that is segmented.

S
Superconducting layer.

HFilaments (*Integer; Default = Global HFilaments*)
Sets the number of height filaments into which segments on this layer will be divided when cuboid filament meshing is used. This is only applied to superconductive layers. More filaments result in exponentially longer calculation times, but it is suggested that this value is set to 2 if a layer is thicker than 1.5 times the penetration depth, and to 3 if a layer is thicker than 2.5 times the penetration depth. For known superconductor thin-film processes it is of no benefit to exceed 3.

HSlice
(*Array of Float*)] Optional parameter that sets the height fraction of consecutive slices in the thickness of every object on a layer. It overrides HFilaments, and sets the thickness of every slice as the thickness of the layer multiplied by a fraction calculated from the corresponding value in the array divided by the sum of all values in the array. Primary application is for higher precision charge distribution when large capacitive structures are analysed.

IDensity (*Floating point*)
Optional parameter that sets the current density of a superconductive Josephson junction layer. This is multiplied by the area of objects in the layer if it exists between the terminals of a port to give junction critical current. This parameter can be used in other processes to yield any linear relationship to the area of a layout object of interest.

IsGP (*Boolean; Default* = False)
Used for auxiliary ground planes in processes with multiple ground plane layers. When set to True, this layer is cropped similarly to the main ground plane if the global parameter CropGP is True.

Lambda (*Floating point; Default = Global Lambda*)
Optional parameter that defines the London penetration depth of the layer. If undefined, the global value is used.

Lambda0 (*Floating point; Default = Global Lambda0*)
Optional parameter that defines the London penetration depth of the layer at 0 K. If undefined, the global value is used.

LambdaFit (*Floating point; Default = Global LambdaFit*)
Optional parameter that defines the London penetration depth of the layer at 0 K. If undefined, the global value is used.

LayerOperations (*String*)
Layer operation expression. LayerOperations provides a powerful way to describe polygon operations, and replaces the deprecated (and limited) LayerAdd and LayerSub from earlier versions of *InductEx* .
Expressions are written with variables (for layers), operators and optional parentheses to determine operator order. Layers can be identified by name or number. Operators, in order of precedence, are:

DEFLATE, SHRINK
Shrink the previous collection of polygons by the succeeding number in Units.

INFLATE, GROW
Grow the previous collection of polygons by the succeeding number in Units.

NOT, -, !
Subtract the next group of polygons from the previous group.

AND, *, &
Find the intersection of the two groups of polygons.

XOR, ^
XOR the two groups of polygons.

OR, +, |
Union the two groups of polygons.

LayerOperations uses the original polygons for each layer (before any operations on that layer). If LayerOperations is defined for a specific layer, that layer's polygons are replaced by the result of LayerOperations for modelling. To retain the layer's polygons, it must form part of the LayerOperations expression.
As an example, to subtract a cut layer CUT from a metal layer M1, the LayerOperations expression for layer M1 will be:

```
LayerOperations = (M1 - CUT)
```

It is thus possible to define intermediate (or scaffolding) layers for which no layout polygons exist, but that can be used to create intricate fabrication steps from polygon operations on other layers.

Mask (*Default = 0*)
Sets the mask polarity of the layer. The options are:

1
Positive layer (deposited where objects are defined). Normally all superconductive and resistive layers.

0
Layer does not contribute to the vertical height of the wafer – typically used to define auxiliary or operational layers.

-1
Negative layer (etched away where objects are defined). Normally all isolation and anodization layers, and only effective on metal layers designated as ground planes.

LossTangent (*Floating point; Default = Global LossTangent*)
Dielectric loss tangent of an isolation (non-metal) layer.

MuR (*Default = 1*)
Optional parameter that defines the relative permeability of a normal metal layer (ignored for superconducting layers). The default is 1 for free space.

Name (*String*)
This parameter is required and sets the name of the layer for terminal label definitions. The name is also applied to output files for visualization.

Number (*Integer*)
This parameter is required, and sets the layer number associated with this layer (which coincides with the GDSII layer number if GDSII input is used). Since version 4.26, this number can range from 0 - 255.

Order (*Integer*)
Required parameter with integer value 0 or higher that sets the construction order of the wafer. The lowest order layer is processed first and the rest in sequence of increasing order. The order numbers do not have to be sequential. Different layers cannot share the same order.

PEC (*Boolean; Default* = False)
If True, sets the layer material as a perfect electrical conductor.

PlanarModel (*Integer; Default = 0*)
Sets the planarization model applied to this layer. The options are:

0
No planarization.

1
Full planarization up to last layer before this one. If set to 1 for an isolation layer directly below a metal layer, it creates depressions at vias through this isolation layer. This implements the complemented caldera planarization method used by AIST for the ADP process [11]. If set to 1 for a metal layer, vias to lower layers will be of the stud type.

2
(and higher) Reserved for future models.

SegmentSize (*Floating point; Default = Global SegmentSize*)
Optional parameter that sets the maximum size of a cuboid filament in Units (see the description under global parameters in section 9.3.5 for more detail). If undefined, the global value is used. For triangular and tetrahedron meshes, only the global SegmentSize is used.

SheetCount (*Integer; Default = 1*)
Sets the number of slices used to subdivide current flow in sheet layers (triangular meshes).

SheetCurrents (*Boolean; Default = Global SheetCurrent*)
Optional boolean parameter to set the use of sheet current models in the layer when *TetraHenry* is used.

SheetModel (*String; Default = 'Thick'*)
Optional string parameter to set the layer sheet current model for *TetraHenry* with sheet layers (triangular meshes). Options are 'Thick', where current flow is projected to the top and bottom surfaces of the film, and 'Thin', where current flows in the centre of the film.

Sigma (*Floating point; Default = Global Sigma*)
Optional parameter that defines the bulk conductivity of a

layer (see the description under global parameters in section 9.3.5 for more detail). If undefined, the global value is used.

TC (*Floating point; Default = Global TC*)
Optional parameter to define critical temperature in kelvins for transition to superconductivity. See the description under global parameters in section 9.3.5 for more detail. If undefined, the global value is used.

Thickness
Required parameter that defines layer thickness in Units.

ViaBypass (*Boolean; Default* = False)
When set to true for a conductive layer of mask type -1 or +1, via etching through an isolation layer directly above this layer will continue to the isolation layer directly below this layer if no conductive object is in the way.

9.3.7 Mesh mask layer entries

Number (*Integer*)
The layer number on which mask polygons are drawn. This should not be a number used in any `Layer` record.

SegmentSize (*Floating point; Default = 1*)
Segment size in Units for all layout structures inside any polygons defined on this layer.

9.3.8 Operators

LayersConnect (*Array of Integer*)
Creates a list of GDS layer numbers (separated by space or TAB characters) of layers on which nodes should be electrically connected by the operator.

LayersRemove (*Array of Integer*)
Creates a list of GDS layer numbers (separated by space or TAB characters) of layers on which objects should be removed by the operator.

LayersTransform (*Array of Integer*)
Creates a list of GDS layer numbers (separated by space or TAB characters) of layers on which objects should be transformed by the operator.

Name
This parameter is required, and sets the name of the operator as it is used in text labels on layouts.

Type (*Default = UD*)
Sets the operator type. The options are:

EC
Electrical connection – Connects nodes on the layers defined in LayersConnect electrically (without segments), and removes objects in layers defined by LayersRemove within which operator text labels fall. Electrical connections are only made between nodes falling inside the polygon with smallest area in any of the layers defined by LayersRemove.

LD
Layer delete – Removes all objects in layers defined by LayersRemove. Useful to get rid of excess structures, such as multiple ground planes when these do not influence extraction results.

MR
Make rectangle – Changes all polygons in the layers defined by LayersRem and within which the operator text label falls to rectangles.

OD
Object delete – Removes objects in layers defined by LayersRemove within which operator text labels fall. Useful to get rid of excess structures, such as multiple ground planes when these do not influence extraction results.

UD
Undefined – does nothing

Layout Input

InductEx was originally developed to processes structures in the GDSII stream file format (more commonly referred to just by the file extension GDS). This is an industry standard binary format, and most layout tools can convert layouts to GDS. This format is still useful for layout entry, although more functionality is available with the *InductEx*-specific IXI input file format.

The IXI file is also used as the primary input file that sets all execution parameters and commands. A single IXI file can describe a complete model setup, with the full layout described by parameterised coordinates in a number of structures, or by a GDS file referenced in the IXI file.

10.1 The IXI file format

10.1.1 IXI description

The IXI input file format is text based and human readable. It was developed to allow an easy interface from Cadence Virtuoso while maintaining readability, but has evolved to allow more complicated geometry read-in that what is possible with GDS files.

The legacy IXI format uses a flat hierarchy with a limited object set, and was deprecated with the release of *InductEx* v6. Legacy files can still be processed if *InductEx* is called with the `-leg` switch.

The latest IXI file syntax now allows it to represent more than just a die layout, but also act as a project file that combines multiple layouts, external geometries, materials, wire bonds and bump bonds to model a complex multi-chip or chip-and-package environment.

Record identifiers in an IXI file are preceded by the `$` character, and record declarations are terminated with an `$End` command.

The IXI input file is read in partially sequentially. A variable has to be declared before it is used, but materials and bond properties can be specified anywhere. The limitation arises from the ability of variables to be used in or referenced from other variables.

The typical structure of the IXI file is shown below. The blocks and records are described in the following sections.

IXI File 10.1.1 — outline.ixi.

```
* A comment line starts with '*'
// A comment can be added after any position in a line with '//'
$COMMAND
  // The optional command block lists modelling and control
  ↪  commands
  // These commands are processed before any files are read in.
$END

$CONTROL
  // The optional control block sets modelling paramater overrides
  // which are applied after the main process file has been read
  ↪  in.
$END

$SPARAM
  // Optional definition of S-parameter setup; required for
  ↪  S-parameter
  // analysis
$END

$VARIABLE
  // Optional variable
$END

$STRUCT
  Name    "main"
  // One STRUCT is required. All layout objects defined here.
  // Nested structures and external GDS files can be included.
$END

$NETLIST
  // Optional. If specified all lines between $NETLIST and $END
  // will be imported as the netlist. Good for single-file setup.
$END
```

10.1.2 IXI units

The IXI input file uses $1\,\mathrm{m}$ as the unit for any coordinate or dimension entry outside of a `STRUCT` record.

All coordinates inside `STRUCT` records are in Unit linked to the associated process technology file. The default unit is $1\,\mu\mathrm{m}$.

Rotation angles in the IXI input file are always specified in degrees.

10.1.3 IXI commands

Simulation and modelling commands used to control a simulation are provided in the `COMMAND` block. These commands are processed right at the start, before any files are read in. Commands are entered as:

key value

or, if *value* is a string that contains whitespace:

key "value"

The available commands are:

AbsTol
(*Floating point; default = 1E-4*)
The absolute error tolerance for GMRES conversion.

Analysis
(*String array; Default = [ ]*)
Request an optional analysis. Options are:

C | CritField
Perform a critical field analysis.

G | Gradient
Perform a gradient field analysis.

U | Uniform
Perform a uniform field analysis (all three Cartesian axes).

Cores
(*Integer; default = system maximum*)
The maximum number of processor cores used by *InductEx* and *TetraHenry*.

Debug
(*Boolean String; Default = False*)
If **True**, *InductEx* writes detailed debug information and mesh analysis files.

EqualSegments
(*Boolean String; Default = False*)
If **True**, and only for a cuboid mesh, *InductEx* equalises all segments to the same size. This must be used with caution, and is only intended for non-Manhattan structures (such as spirals) with high polygon vertex counts. For narrow structures, this method creates a risk of conductor pinch-off and thus open circuits.

Excitation
(*String; Default depends on the numerical analysis method specified with the MODE command*)
Specifies the source/port excitation method. To solve inductance, capacitance and critical field of moats and vortex pinning sites, do not change excitation default. This command is used to control a simulation so as to view the current distribution and magnetic fields for a defined current or phase excitation.

AC
Voltage excitation at the defined frequency. Required to calculate inductance a generalised network that could contain resistance. It is the default in MQS, EMQS and Full-wave modes.

DC
Voltage excitation at 0 Hz. It is the default in EQS mode.

PHASE
Phase excitation at 0 Hz. It is the default when critical field analysis is done.

ExportFile
(*String*)
Directory/Path and name of the optional post-processed layout output file (useful for model debugging purposes). If the text contains spaces, enclose in double quotation marks (“ ").
If this command is provided, *InductEx* will regurgitate the processed main layout to both a GDS and SVG file.

Fidelity
(*String; Default = High*)
Provides two quick fidelity settings. Options are:

High
Edge slices for superconducting layers by default (at the width of penetration depth), with `AbsTol = Tol` = 1E-6. Important when mutual inductance is extracted.

Low
No edge slices, `AbsToL = Tol`=1E-6.

Frequency
(*Floating point; default = 1E10*)
Sets the analysis frequency.

HideTerminal
(*Boolean String; Default = False*)
Sets the terminal visibility for *TetraHenry*. If **True**, *TetraHenry* is executed in a hidden terminal.

HoleBridge
(*String; Default = XY*)
Sets the polygon simplification algorithm that bridges holes. The default is to slice polygons open between existing vertices.

EV
Slice between existing vertices.

XY
Slice polygons through hole centres along the *x* and *y* axes (alternating). This creates more vertices where.

Iterations
(*Integer; default = 400*)
The number of GMRES iterations that *TetraHenry* will be limited to for each solution. GMRES completes when the specified error tolerance (`ABSTOL` and `TOL`) is reached, or when the iteration limit is reached. For capacitance extraction, a limit of 800 is recommended. For S-parameter calculation, inspect the error tolerance that *TetraHenry* reports. A limit of as high as 2000 could be required for convergence.

License | License
(*String; default is the previous path and name as read from the INI file.*)
Directory/Path and name of the licence file. If the text contains spaces, enclose in double quotation marks (“ "). It overrides the previous path and name of the licence file as read from the INI file, and is saved to the INI file on exit to become the new default.

Log
(*Boolean String; Default = False*)
Instructs *InductEx* to write a detailed log file; useful for debugging mesh issues.

MeshFile
(*String*)
Directory/Path and name of the output mesh file. If the text contains spaces, enclose in double quotation marks (“ ").

MeshType
(*String; Default = Auto*)
MQS/EMQS Mesh built according to option:

Auto | Hybrid
Triangles (sheets) and tetras (volumes) as per process (LDF) file for conductors. Vias are tetras (volumes).

Cube | Cub | Cuboid | Fil | Filaments
Cuboid volume segments.

Tri | Triangle | Triangles
Triangles (sheet models) for all conductors, tetras (volumes) for vias.

Tet | Tets | Tetra | Tetras | Tetrahedron
Tetrahedra everywhere.

EQS meshes are always triangles wrapped over the surface of volumes.

Mode
(*String; Default = MQS*)
Numerical analysis method. Options are:

MQS
Magnetoquasistatic method to calculate inductance and resistance, magnetic coupling and coupling from trapped flux.

EQS
Electroquasistatic method to calculate capacitance.

EMQS
Electro-magnetoquasistatice method to calculate S-parameters.

FULL
Full-wave method to calculate S-parameters.

Z0
Simultaneous MQS and EQS solutions to calculated characteristic impedance.

Netlist
(*String*)
Directory/Path and name of the circuit netlist file for inductance and resistance calculation. If the text contains spaces, enclose in double quotation marks (“ ").

NoNetlist
(*Boolean String; Default = False*)
If **True**, *InductEx* runs a solution without a netlist. Inductance and resistance calculation not possible, but current distribution can be seen. Set **True** for S-parameter calculations.

Plot
(*String array; Default = [ ]*)
Array between square brackets, with characters separated by space, comma or semi-colon, to select output plots. Array options are:

B
Plot magnetic flux density (T) in MQS, EMQS or FULL mode.

D
Plot electric displacement field ($\mathrm{C/m^2}$) in EQS, MQS, EMQS or Full mode.

E
Plot electric field strength ($\mathrm{V/m}$) in EQS, MQS, EMQS or Full model.

H
Plot magnetic field strength ($\mathrm{A/m}$) in MQS, EMQS or FULL.

J
Plot current density ($\mathrm{A/m^2}$) in MQS, EMQS or Full mode.

Q
Plot charge density ($\mathrm{C/m^2}$) in EQS mode.

To plot current density and magnetic flux density in MQS mode, for instance, use for example:
`PLOT [ J B ]`, or
`PLOT [B,J]`.

Prefix
(*String*)
Optional prefix to be added to name of all output files.

Process
(*String*)
Directory/Path and name of the process file (LDF format) that describes the fabrication steps and layer/material parameters and properties for the main layout. Additional dies imported through a `Die` record each have their own designated process file.

ProcessKey
(*String*)
The passkey to the main process (LDF) file, if it is protected. This string should contain alphanumerical characters and can use most special characters. Whitespace characters and quotation marks are not allowed. The string can be enclosed in double quotation marks (not part of the passkey).
The process passkey is generated by the originating foundry or by SUN Magnetics when the protected process file is packed.

Resistance
(*Boolean String; Default = True*)
If **False**, *InductEx* changes all resistive layers in the process file to have `FilmType = N`. This effectively removes resistive structures from a mesh and allows simplified inductance-only netlists (for instance in a logic cells when a quick inductance extraction without shunt resistance is required).

Sparsity
(*String; Default = Default*)
Sparsity pattern of the preconditioner matrix in *TetraHenry*.

CUBE
Preconditioner matrix contains near-field interactions between elements.

DEFAULT
Use Diagonal in MQS mode, and **Cube** in EQS, EMQS and FULL modes.

DIAG | DIAGONAL
Preconditioner matrix contains diagonal entries.

TetraHenry | TTH
(*String; default is the previous path and name as read from the INI file.*)
Directory/Path and name of the *TetraHenry* executable. If the text contains spaces, enclose in double quotation marks (“ "). It overrides the previous path and name of the *TetraHenry* executable as read from the INI file, and is saved to the INI file on exit to become the new default.

Tol
(*Floating point; default = 1E-4*)
The relative error tolerance for GMRES conversion.

Top
(*String*)
Name of the top structure in the IXI file. To select a cell within an imported GDS layout, use the `Top` field inside its `$IMPORTLAYOUT` record, as described under `ImportLayout` below.

TopLevelTextOnly
(*Boolean String; Default = True*)
If **True**, *InductEx* processes only text labels at the top hierarchical level of a layout, otherwise all text from nested structures are also processed.

Verbosity
(*Integer; default = 0*)
Sets the verbosity level of output reporting. Highest level is 5.

10.1.4 IXI Controls

Controls used to override modelling parameters are proved in the optional `CONTROL` block. These controls are processed after the process file has been read in. Available controls are :

Set
Set (override) a designated parameter value.

10.1.5 IXI records and fields

Records contain fields with values, and some records can also contain other records. The records are:

Ball
The definition of the ball geometry parameters for a ball-and-stitch wire bond.
Supported fields:

BallDiameter
real value or expression; default = 120E-6
Diameter in metres of the ball bond at its centre (half of its height).

BallHeight
real value or expression; default = 120E-6
Height in metres of the ball from the contact pad and the wire joint.

Name
string
The name of the ball to reference from a bond setup.

PadDiameter
real value or expression; default = 70E-6
The diameter of the ball at the contact pad and at the wire joint.

Supported records: None.
The field values represent the dimensions in metres, and are real-valued. The fields define the ball geometry as shown in Fig. 13.4.

Bump
The declaration of a bump bond for a flip chip interconnect.
Supported fields:

BondSetup
String
The name of a bump bond setup defined in a separate `BumpSetup` record.

EndPin
String

StartPin
String

Supported records: None.
`StartPin` and `EndPin` identify the start and end pins for the bump bond. The pins must be on different dice, each imported with a separate `Die` record and with one die flipped 180 degrees around the *x* or *y* axis. If pin locations are not aligned in the *z*-direction, the bump centre is placed at the midpoint of the *x* and *y* coordinates of the start and end pins. The pin names are specified in the field value as *diename.pinname*.

BumpSetup
The definition of the bump bond parameters for flip chip interconnects.
Supported fields:

CircumferenceSections
integer value or expression; default = 6
Optional definition of the minimum number of sections around the circumference of a wire bond. The floor is the maximum of 6 and the circumference divided by the mesh size, and explicit definition of `CircumferenceSections` is only necessary if a smoother wire is required than what is obtained with the selected mesh size.

Diameter
real value or expression; default = 30E-6
Diameter in metres of the bump at each contact pad.

Material
String
Name of a defined material for the bump bond. It must match the name of a material defined in the IXI file.

MeshSize
real value or expression; default = 1E-3
Mesh size in metres of the bump bond.

MidDiameter
real value or expression; default = 38E-6
Diameter in metres of the bump center (midway between opposing contact pads). If `MidDiameter` is smaller than `Diameter`, the bump is concave. It is convex if `MidDiameter` is larger than `Diameter`.

Name
String
The name of the bump setup, to be referenced when a bump bond is defined between two contacts on different dies.

Supported records: None.

Component
An lumped component, only processed inside an `Sparam` record, that is connected to a port on the layout (or in the mesh).
Supported fields:

Impedance
*real value or expression; default is the impedance of the **Sparam** record in which it is defined*
Characteristic impedance (real value only) of the port in the *n*-port network for S-parameter calculation.

Name
string
The name of the component.

Value
real value or expression
The element value of the component in SI units.

Port
string
The name of the port on the layout or in the mesh. It must match an existing port. The name is not case sensitive.

Type
string; *default = INDUCTOR*
The component type. Supported components are:

C | Cap | Capacitor
Capacitor (value in farad).

L | Ind | Inductor
Inductor (value in henry).

L | Res | Resistor
Resistor (value in ohm).

Supported records: None.

Die
A layout in GDSII or IXI format for importation as a separate die into the layout model. The die must have an associated layer definition file to define the process. Any number of dice can be imported, as long as they do not overlap during model building.
Supported fields:

Name
String
This is the name by which the `Die` record is referenced when a wire bond or bump bond is declared in a `Bump` record or `WireBond` record.

Layout
String
The path and name of the layout input file (in GDS or IXI format) for this die.

Process
String
The path and name of the process file (.LDF) that describes the layer stack and fabrication/material parameters for this die.

Rotate
triple of real values or expressions, *comma-separated*, *enclosed in parentheses*; *default = (0, 0, 0)*
Rotation in degrees of the die around the *x*, *y* and *z* axes respectively (in that order).
Note that the coordinate triple **must be on a new line**. If multiple coordinate triples are defined, only the last one is used.

Translate
triple of real values or expressions, *comma-separated*, *enclosed in parentheses*; *default = (0, 0, 0)*
Translation in metres in of the rotated die in the *x*, *y* and *z* directions respectively, starting from the model universe origin at (0, 0, 0).
Note that the coordinate triple **must be on a new line**. If multiple coordinate triples are defined, only the last one is used.

Supported records: None.
The structure is:

```
$DIE
  Layout "text"
  Name "text"
  Process "text"
  Rotate
  (Rx, Ry, Rz)
  Translate
  (Tx, Ty, Tz)
$END
```

FPlane
A 2D plane used for field calculations. The plane is defined as a rectangle between opposite corners (x_1, y_1) and (x_2, y_2) in the *xy* plane at *z* = 0 and then rotated and translated as specified. This record is called outside of a `Struct` record, so that all coordinates (position and translation) are in metres. Rotation angles are in degrees. *z*-axis coordinates in `Coords` are ignored.
Supported fields:

Coords
two triples of real values or expressions, *comma-separated*, *enclosed in parentheses*
Coordinates in metres of opposite corners in the *xy* plane.
Note that each coordinate triple **must be on a new line**. Exactly two coordinates are required. If more triples are provided, only the first two are used.

Rotate
triple of real values or expressions, *comma-separated*, *enclosed in parentheses*; *default = (0, 0, 0)*
Rotation in degrees of the die around the *x*, *y* and *z* axes respectively (in that order).
Note that the coordinate triple **must be on a new line**. If multiple coordinate triples are defined, only the last one is used.

Translate
triple of real values or expressions, *comma-separated*, *enclosed in parentheses*; *default = (0, 0, 0)*
Translation in metres in of the rotated die in the *x*, *y* and *z* directions respectively, starting from the model universe origin at (0, 0, 0).
Note that the coordinate triple **must be on a new line**. If multiple coordinate triples are defined, only the last one is used.

Supported records: None.
The structure is:

```
$FPLANE
  Coords
  (x1, y1, 0)
  (x2, y2, 0)
  Rotate
  (Rx, Ry, Rz)
  Translate
  (Tx, Ty, Tz)
```

```
    $END
```

Gds

A GDS layout file in GDSII format from which layout objects are imported into the main (primary) layout, ***and not as a separate die***. The `Gds` record can only be used inside the main `Struct` record before any other objects are defined, because it rewrites the structure array. Other structures and objects can be defined after this importation. If multiple GDS files are specified, the last one will be imported. In this version of *InductEx*, rotation and translation of the imported layout objects from the GDS file are not supported.

Note that this record is not used to import an entire chip layout as part of a multi-chip model (that is done with `Import`), but rather to add layout objects to single layout. It is retained for legacy purposes, but application is severely limited.

The `GDS` record only allows a GDS file to be imported into the layout. To import an IXI file, use the `ImportLayout` record.

Supported fields:

Name

string

This is the name of the GDS file to import structures from. All structures are imported.

Supported records: None.

ImportLayout

A layout file in GDSII or IXI format from which layout objects are imported into the main (primary) layout, ***and not as a separate die***. The `ImportLayout` record can only be used inside the main `Struct` record before any other objects are defined, because it rewrites the structure array. Other structures and objects can be defined after this importation. If multiple layout files are specified, the last one will be imported. In this version of *InductEx*, rotation and translation of the imported layout objects from the GDS file are not supported.

Note that this record is not used to import an entire chip layout as part of a multi-chip model (that is done with `Die`), but rather to add layout objects to single layout. It is retained for legacy purposes, but application is severely limited.

Where an IXI file is imported, it may not have a `COMMAND` or `CONTROL` block. The `ImportLayout` record supersedes the older `Gds` record.

Supported fields:

Name

string

This is the name of the GDS or IXI file to import structures from.

Top

string; optional

The name of the cell (structure) to use from the imported GDS file. Its geometry and referenced cells form the imported layout. Use the cell name stored in the GDS file, enclosed in double quotes; this need not match the enclosing `$STRUCT` name.

If `Top` is omitted, *InductEx* selects a cell automatically from the layout hierarchy and port labels. With several unreferenced cells and no port labels, it selects the last unreferenced cell in file order. Specify `Top` explicitly when a particular cell is required.

Legacy

boolean value; default = false

If `true`, the legacy (*InductEx* v5) IXI format is used.

Supported records: None.

For example, the following imports `Layout_Microstrip` and its referenced cells from `Test.gds`:

```
$STRUCT
    Name "microstrip"
    $IMPORTLAYOUT
        Top "Layout_Microstrip"
        Name "Test.gds"
    $END
$END
```

Here `microstrip` names the IXI structure, while `Layout_Microstrip` selects the GDS cell. This import field is separate from `Top` in the `$COMMAND` block, which selects the main IXI structure.

Import

A Gmsh mesh file for importation, with all dimensions in Unit metres (the default is 1 metre).

The `Name` field specifies the path and name of a .MSH file to import.

Supported fields:

Name

String

The path and name of a Gmsh mesh file (.MSH) to import as en external object.

Rotate

triple of real values or expressions, comma-separated, enclosed in parentheses; default = (0, 0, 0)

Rotation in degrees of the meshed object around the *x*, *y* and *z* axes respectively (in that order).

Note that the coordinate triple **must be on a new line**. If multiple coordinate triples are defined, only the last one is used.

Scale

real value or expression; default = 1

Scale factor multiplied into every coordinate.

Terminal

two space-separated strings

Each string can be delimited by double quotation marks, but no white space is allowed in a string. The first string specifies a port terminal (polarity included), such as `P1+`. The second string defines an “Physical" inside the .MSH file to which the port terminal is assigned.

Multiple **Terminal** fields are allowed.

Translate

triple of real values or expressions, comma-separated, enclosed in parentheses; default = (0, 0, 0)

Translation in metres in of the meshed object in the *x*, *y* and *z* directions respectively, starting from the model universe origin at (0, 0, 0).

Note that the coordinate triple **must be on a new line**. If multiple coordinate triples are defined, only the last one is used.

Unit
real value or expression; default = 1
The unit of the coordinates in the mesh file relative to 1 metre.

Supported records:

Physical
A record that defines the material of a specified "Physical" in the mesh file. Multiple **Physical** records are supported.

The structure is:

```
$IMPORT
  Name "text"
  Rotate
  (Rx, Ry, Rz)
  Scale real value or expression
  Terminal "text" "text"
  Translate
  (Tx, Ty, Tz)
  Unit real value or expression
  $PHYSICAL
    Material "text"
    Name "text"
  $END
$END
```

Mask
A mask object, defined outside of the main `Struct` record, that defines a mask area as a polygon. Masks are used to mask out the layer-linked segment size of objects and to enforce a specified mesh size over masked area. There is no limit on the number of masks defined. Where masks overlap, the last defined mask applies.
All dimensions in the `Mask` record are in Unit metres, as specified in the layer definition file of the process used for the layout.
`MeshSize` sets the mesh size for all nodes inside the mask. The `XY` field takes two-dimensional coordinate pairs as values. Coordinates are real values or expressions. Coordinate pairs are placed in parentheses and read from consecutive lines (one pair per line). There is no limit to the number of coordinate pairs. Supported fields:

MeshSize
real value or expression; default = 1
The size of meshed elements inside the mask polygon. This overrides the process file settings.

XY
doubles of real values or expressions, comma-separated, enclosed in parentheses
Coordinates in process file Units of the polygon that describes the mask object.
Note that each coordinate double **must be on a new line**.

Supported records: None.
The structure is:

```
$MASK
  MeshSize real value or expression
  XY
  (x1, y1)
  (x2, y2)
  ...
  (xN, yN)
  $END
```

Material
A material record that defines material parameters for a physical structure. Multiple physical structures can reference the same material.
Supported fields:

EpsRIn
real value or expression;
The dielectric constant or relative permittivity (ε_r of the material. If not specified, the material has no dielectric constant.

Lambda
real value or expression; default = 0
The London penetration depth in metres.

MuR
real value or expression; default = 0
The London penetration depth in metres.

Name
string
The name by which the `Material` record is referenced when material is assigned in a `Physical` record.

Sigma
real value or expression; default = 0
The bulk conductivity in $1/(\Omega\,\mathrm{m})$ or S/m.

Supported records: None.

Netlist
Optional definition of the circuit netlist, if an external file is not specified.
All lines between the record start line (`$NETLIST`) and the record end line (`$END`) are read as the netlist. This replaces a netlist file.
Supported fields: None.
Supported records: None.

Path
A path or wire. A `Path` record is only processed if it is defined inside a `Struct` record, because it defines a layout object linked to technology file.
Supported fields:

Layer
string
The name of the layer on which the path is defined. It must match a layer name defined in the layer definition file for the process linked to the layout. It is case insensitive.

Text
string
Optional text assigned to the path. Used to define port terminals on a terminal layer.

Width
real value or expression; default = 0
The width of the path in process file Units.

XY
doubles of real values or expressions, comma-separated, enclosed in parentheses

Coordinates in process file Units of the polygon that describes the mask object.
Note that each coordinate double **must be on a new line**.

Supported records: None.

Note: Paths in the IXI format conform to PathType 0 of the GDSII format, so that the there is no structure extending beyond the start and end nodes.
The structure is:

```
$PATH
  Layer "text"
  Text (Optional) "text"
  Width real value or expression
  XY
  (x1, y1)
  (x2, y2)
  ...
  (xN, yN)
$END
```

Physical
A physical record, only processed inside an `Import` record, that defines the physical object in a .MSH file to import.
Supported fields:

Material
string
The name of material linked to this physical, which must match that the name of a material defined anywhere in the IXI file.

Name
string
The name of this physical definition, and must match that of a physical object in the .MSH file specified in `Import`.

Supported records: None.

Poly
A polygon. A `Poly` record is only processed if it is defined inside a `Struct` record, because it defines a layout object linked to technology file.
Supported fields:

Layer
string
The name of the layer on which the polygon is defined. It must match a layer name defined in the layer definition file for the process linked to the layout. It is case insensitive.

Text
string
Optional text assigned to the path. Used to define port terminals on a terminal layer.

Width
real value or expression; default = 0
The width of the path in process file Units.

XY
doubles of real values or expressions, comma-separated, enclosed in parentheses
Coordinates in process file Units of the polygon that describes the mask object.
Note that each coordinate double **must be on a new line**.

Supported records: None.

The structure is:

```
$POLY
  Layer "text"
  Text (Optional) "text"
  XY
  (x1, y1)
  (x2, y2)
  ...
  (xN, yN)
  (x1, y1)
$END
```

Port
An S-parameter port, only processed inside an `Sparam` record, that links the S-parameter port to a port on the layout (or in the mesh).
Supported fields:

Impedance
*real value or expression; default is the impedance of the **Sparam** record in which it is defined*
Characteristic impedance (real value only) of the port in the *n*-port network for S-parameter calculation.

Name
string
The name of the port on the layout or in the mesh. It must match an existing port. The name is not case sensitive.

Supported records: None.

Sparam
The setup of an S-parameter analysis.
Supported fields:

Impedance
real value or expression; default = 50
Default characteristic impedance (real value only) of a port in ohm.

Mode
string; Defaut = DEC
The frequency stepping: linear (LIN) or decade (DEC).

StartFrequency
real value or expression; default = 1E9
Frequency in hertz where S-parameter analysis starts.

Steps
integer; default = 10
Number of simulation steps from `StartFrequency` to `StopFrequency`.

StopFrequency
real value or expression; default = 1E9
Frequency in hertz where S-parameter analysis ends.

Supported records:

Component
A record that defines a lumped component attached over a port that exists in the model structure.
Any number of components can be defined, and components may be in parallel (and thus over the same port).
Component polarity matches the port polarity.

Port
A record that defines an S-parameter port (one of the ports of an *n*-port network) and links it to a port in the model structure. The S-parameter port polarity matches that of the port in the model structure.

Sref
A reference to a structure inserted at the `XY` coordinates. The structure is reflected around the *x*-axis before angular rotation if `Reflect` is *true*.
An `Sref` record can only be called inside a `Struct` record.
Supported fields:

Angle
real value or expression; default = 0
Rotation angle, measured in degrees and applied counterclockwise, of the referenced structure around the insertion point.

Name
string
The name of the structure to insert. It must match a defined structure name.

Reflect
boolean value; default = false
If `true`, the structure is reflected (mirrored) around the *y* axis before rotation.

XY
doubles of real values or expressions, comma-separated, enclosed in parentheses
Coordinates in process file Units. Only the first entry is read as the insertion point of the referenced structure.
Note that the coordinate double **must be on a new line**.

Supported records: None.

Struct
A structure that contains layout objects such as polygons and paths, text and structure references.
Supported fields:

Name
string
The name of the structure, by which it is referenced through **Sref** in other structures.

Supported records:

Gds
A record that identifies a GDS file from which layouts objects are imported directly into this structure.

Path
A wire or path with defined width.

Poly
A polygon.

Sref
Reference to a nested structure.

Text
Text, mostly used for labels.

Text
A text label. It can only be defined inside a `Struct` record.
Supported fields:

Layer
string
The name of the layer on which this text is defined. It must match a layer name defined in the layer definition file for the process linked to the layout. It is case insensitive.

Text
string
The text string that is applied as the label text. If the text string contains spaces, as with a port definition that specifies name and layer(s), the string must be enclosed in double quotation mark

XY
doubles of real values or expressions, comma-separated, enclosed in parentheses
Coordinates in process file Units. Only the first entry is read as the insertion point of the text label.
Note that the coordinate double **must be on a new line**.

Supported records: None.

The structure is:

```
$TEXT
  Layer "text"
  Text "text"
  XY
  (x, y)
$END
```

Variable
A variable a name and a numerical or expression value:
Supported fields:

Name
string
The name of the variable to be used as reference later. Variable names are case-insensitive.

Value
expression
A numerical value, or an expression parsed to find a numerical value. For more detail on expression evaluation, see section 10.1.6.

Supported records: None.

Wedge
The definition of the wedge geometry parameters for a wedge-wedge wire bond, or the stitch geometry for a ball-and-stitch wire bond.
Supported fields:

Name
string
The name of the wedge/stitch by which it is referenced through the **EndBond** or **StartBond** fields in **WireBond-**

Setup.

SpreadRatio
real value or expression
The spreading ratio of the wedge (as a multiple of the wire bond diameter).

ThickRatio
real value or expression
The Thickness ratio of the wedge foot (as a fraction of the wire bond diameter).

Supported records: None.

WireBondSetup
The definition of wire bond parameters.
Supported fields:

Alpha
real value or expression; default = 85
Angle in degrees from horizontal to wire departing from ball bond (only used for ball-stitch configuration).

BendRadius
real value or expression; default = 10E-6
Radius in metres of all three bends in a bend-stitch wire bond (only used for ball-stitch configuration).

Beta
real value or expression; default = 20
Angle in degrees from horizontal to wire arriving at wedge/stitch bond in ball-stitch configuration (only used for ball-stitch configuration).

CircumferenceSections
integer value or expression; default = 6
Optional definition of the minimum number of sections around the circumference of a wire bond. The floor is the maximum of 6 and the circumference divided by the mesh size, and explicit definition of `CircumferenceSections` is only necessary if a smoother wire is required than what is obtained with the selected mesh size.

Diameter
real value or expression; default = 30E-6
Diameter in metres of bond wire (both wedge-wedge and ball-stitch configurations).

EdgeSlice
real value or expression; default = 0
Distance in metres from outer surface of a bond wire and its ball or wedge feet at which a skin-depth slice is added for better fidelity in current distribution modelling.

EndBond
String
Name of the bond foot definition for the end/termination of the wire bond.

EndType
String; default = wedge
Specification is ignored, as the only option is "wedge". The field is reserved for future use.

H1
real value or expression; default = 300E-6
Height in metres of ball-stitch bond wire above pad at the ball end (only used for ball-stitch configuration).

Material
String
Name of a defined material for the wire bond. It must match the name of a material defined in the IXI file.

MeshSize
real value or expression; default = 1E-3
Mesh size in metres of the wire bond and its ball/wedge feet.

StartBond
String
Name of the bond foot definition for the start/origion of the wire bond.

StartType
String; default = wedge
Bond foot type for the start/origin of the wire bond. Options are:

Ball
A ball bond.

Wedge
A wedge or stitch bond.

VertexHeight
real value or expression; default = 500E-6
Height in metres of the apex of a wedge-wedge bond wire above the highest bond pad (only used for wedge-wedge configuration).

Supported records: None.

WireBond
The declaration of a wire bond interconnect.
Supported fields:

BondSetup
String
The name of a wire bond setup defined in a separate `WireBondSetup` record.

EndPin
String

StartPin
String

Supported records: None.
`StartPin` and `EndPin` identify the start and end pins for the wire bond. The pins can be on the same die or on different dice, each imported with `Die`. The die surfaces must have the same normal. The pin names are specified in the field value as *diename.pinname*.

10.1.6 IXI expressions

Expressions are evaluated for variables, as well as for all record fields in the IXI file where real or integer values are expected.

Expressions may be enclosed by double quotation marks, but *InductEx* will also evaluate expressions that are not enclosed by double quotation marks. The decimal separator is a period (.), and real numbers use E notation (such as `2.07E-15`).

The following constants are supported:

Pi
: Value = 3.14159265359
e
: Value = 2.718281828
Phi0 | P0
: Value = 2.067833848E-15

Supported functions are:

Abs(x)
The absolute value of *x*.
Arccos(x)
The inverse cosine of *x* (where *x* is in radians).
Arcsin(x)
The inverse sine of *x* (where *x* is in radians).
Arctan(x)
The inverse tangent of *x* (where *x* is in radians).
Ceiling(x)
The ceiling (nearest lower integer value) of *x*.
Cos(x)
The cosine of *x* (where *x* is in radians).
Exp(x)
The exponent e^x.
Floor(x)
The floor (nearest higher integer value) of *x*.
Log(x)
The natural logarithm of *x*; $\ln x$.
Log10(x)
The base 10 logarithm of *x*.
Max(x, y)
The maximum of *x* or *y*.
Min(x, y)
The minimum of *x* or *y*.
Round(x)
The nearest integer value of *x*.
Pwr(x, y)
The power function; x^y.
Sin(x)
The sine of *x* (where *x* is in radians).
Sqr(x)
The value of *x* squared; x^2.
Sqr(x)
The square root of *x*; $\sqrt{x}$.
Tan(x)
The tangent of *x* (where *x* is in radians).

Parentheses are used to set precedence. Otherwise, operator precedence is:

- Unitary minus
- ^ (Raise to the power)
- *, /
- +, -

Some valid examples of the use of expressions are:

```
$VARIABLE
  Name x1
  Value -2 * Phi0
$END
$VARIABLE
  Name x2
  Value "x1 * 1E15"
$END
$VARIABLE
  Name x3
  Value ( Sqrt(2) * Max(x1, x2) )
$END
...
$PATH
  Layer MET1
  XY
  ( -length/2 , 0 )  // where 'length' is a defined variable
  (  length/2 , 0 )
$END
```

10.2 GDSII

For GDSII input files, *InductEx* only reads objects with DATATYPE = 0 and text labels with TEXTTYPE = 0.

If, for any reason, DATATYPE or TEXTTYPE values other than zero should be processed, you may set the DataTypeNotZero parameter in the layer definition file to True.

10.2.1 Limitations of GDSII support

InductEx does not read in Node Elements from GDSII files.

10.3 Port label input file

When a port label input file (in ASCII text format) is specified with the `-p` parameter, port labels, fluxon positions and bond points (both text and coordinates) are read in from this file. These text labels are added to any port, fluxon or bond position labels that may have been read in from the geometry input file. The file may have any suffix, although `.txt` is preferable.

Port declarations follow the rules set out in Section 12, and the file syntax is:

P*label*; (*x*, *y*)

where *Plabel* is the port label that contains the port name and positive and/or negative terminal layers as described in Section 12. The label coordinates within the layout are at

x and *y*, which is in `Units` as defined in the layer definition file.

Examples of valid definitions in a port label input file are:

`P1 M2 M0; (0, 25.5)` - Port P_1 between layers M2 and M0, at *x* = 0; *y* = 25.5.

`PIN+ M2; (10, 20)` - Port P_{IN}, with positive terminal on M2, at *x* = 10; *y* = 20.

`#BondPin1; (1500, -200)` - Bond point at *x* = 1500; *y* = -200.

10.4 Modelling aids defined on layouts

Some modelling aids are available to prune or simplify circuit structures during modelling, mostly to reduce the demand on computing resources.

10.4.1 Extraction boundary

InductEx allows you to specify an extraction boundary for a die, outside of which all structures are ignored. This is helpful in large circuit layouts where the structures to be extracted do not span the entire layout. With an extraction boundary, structures outside the area of interest are not discretized, which saves calculation time and memory.

The extraction boundary also makes it easy to separate isolated sections of a negative mask layer (mostly ground planes in current recycling circuits) during modelling.

The extraction boundary is defined on the layout in a special non-fabrication layer which is set with the `ExtractBoundLayer` global parameter in the layer definition file. Any combination of layout objects may be used – it is thus possible to define several isolated extraction zones.

10.4.2 Operator declarations

Operators are declared with text labels on any layer. Operators act at the coordinates of the text label. Syntax is:

`@`*Name*

with the first character always "@". The name of the operator must match that of an operator declared in the layer definition file, although the name is case-insensitive. If no match is found, the operator is ignored.

Circuit Netlist

For most inductance calculation problems with *InductEx* a circuit netlist file is needed. This netlist file is in part a subset of the SPICE electrical netlist. The netlist should contain all the inductors and mutual inductors in the circuit model, and the values assigned to each will be used as the "design value" during post-processing.

11.1 Title line

An *InductEx* netlist file does not start with a title line. If you use a title line, start it with a comment marker.

11.2 Comments

If a line in a netlist file starts with "*", *InductEx* discards the line as a comment.

In-line comments are marked with "//". All text after the marker until the end of the line is discarded as a comment. A comment cannot be closed.

11.3 Circuit elements

Inductors, resistors and mutual inductance (or coupling elements) are supported.

Elements are defined as:

L*name node+ node- designvalue* <[*back-annotate*]> <coupling=off>

R*name node+ node- designvalue* <[*back-annotate*]>

K*name Lname1 Lname2 designcoupling* <[*back-annotate*]>

M*name Lname1 Lname2 designvalue*

Node names may be any alphanumeric combinations excluding whitespace characters such as Space and TAB.

All design values are optional. If these are not specified, *InductEx* only prints the extracted values and does not list any comparison between design and extracted values. Design values are in the units specified in the layer definition file by parameters UnitI, UnitL or UnitR.

The back-annotate assignment, where name of the circuit element to which the extracted value must be assigned in the back-annotated file is enclosed in square brackets, is also optional. If it is not defined, the specific parameter value will not be written to a back-annotated file.

An optional `coupling=off` parameter excludes an inductor from coupling calculations when external magnetic fields are applied.

Coupling for mutual inductance is specified with `K` for compatibility with SPICE, although `M` can also be used if SPICE compatibility is not required. Back-annotation is only possible for a `K` element.

Holes are never defined in the circuit netlist file – *InductEx* adds components for holes automatically when these are specified with flux labels on the layout.

11.4 Ports

The netlist should contain port definitions for the extraction setup. Port definitions on layouts are discussed in Section 12. Some valid definitions in the netlist file are:

```
P1          1        0
Ptwo        N1       N2
PAny_name   posterm ground
J3          node1    node2    250
B1          node1    node2    100E-6
Ibias       node+    node-    source=i    amplitude=0.5E-3
```

Port identifiers are the symbols `P`, `J`, `B` and `I`. These can be used in any way, but it allows users to label junctions and bias current sources more clearly.

If a numerical value is listed after the node names, and the port is defined in a via, the value is used as the design value for the via area. For a Josephson junction, it is possible to extract the junction critical current from the layout size and the trilayer's `IDensity` parameter in the layer definition file.

The source type of a port is by default assumed to be a voltage source with amplitude of 1 V. This can be altered for specific analyses, such as finding current distribution in a circuit, by specifying the source type is current (`i`). The amplitude is also controllable.

11.5 Control block

A control block may be included in the netlist file, and can be used to assign values to certain variables in order to exert control over modelling and solution methods. An example of the use of the control block is:

```
.control
SET RCond = 1e-16
SET AbsTol = 1e-6
.endc
```

Control block statements contain a command (`SET`, `SWEEP`), parameter name, equal sign and a value or range of values.

All parameter values assigned in the control block override the values assigned in the layer definition file.

11.5.1 BACK-ANNOTATE command

For compact SPICE model extraction is back-annotation of calculated parameters to a SPICE-based simulation deck. The circuit netlist file used by *InductEx* is stripped of components such as Josephson junctions and current, voltage or phase sources, and is populated with excitation ports. The circuit netlist is often also stripped of resistive components. The extraction circuit netlist is thus practically never the same as the SPICE simulation deck.

With back-annotation, *InductEx* writes extracted parameter values directly to the SPICE simulation deck. The target deck file is nominated in the control block in the extraction circuit netlist file with the `BACK-ANNOTATE` command. The target deck file has to exist, and any parameters have to be identified in sqaure brackets in the extraction netlist, as shown in Section 11.3.

If the target simulation netlist is a subcircuit in a larger simulation deck, the subcircuit can be identified too. Syntax is:

`BACK-ANNOTATE` *targetfilename*

`BACK-ANNOTATE` *targetfilename* `SUBCKT` *targetsubcircuitname*

As a safety precaution, the target netlist file is never overwritten but a new file, with the string "`_annotated`" appended to the target file name, is created.

11.5.2 SET command

Inside the control block, global and layer parameters can be assigned new values with the `SET` command.

Syntax is:

`SET` *parametername = newvalue*

`SET layer` *layername parametername = newvalue*

Global parameters currently supported are: `frequency`, `gpoverhang`, `lambda0`, `lambdafit`, `rcond`, `segmentsize`, `tc`, `temperature`, `uniti`, `unitl` and `unitr`.

Layer parameters supported are: `bias`, `hfilaments`, `lambda`, `lambda0`, `lambdafit`, `segmentsize`, `sigma`, `mu_r`, `tc`, `temperature` and `thickness`.

Solution stability parameters supported are: `rCond` and `nonZeroSVThreshold`.

Examples are:

```
SET gpOverHang = 5.0
SET segmentsize = 1.5
SET layer M1 lambda = 0.085
```

11.5.3 SWEEP command

Inside the control block, parameter sweeps for global and layer parameters can be defined with the SWEEP command. Single and double parameter sweeps are supported, and results are written to a Matlab-compatible M-file for easy visualization.

Syntax examples are:

`SWEEP` *parametername startvalue incrementvalue stopvalue*

`SWEEP layer` *layername parametername startvalue incrementvalue stopvalue*

`SWEEP` *parameter1 start1 inc1 stop1 parameter2 start2 inc2 stop2*

`SWEEP variable` *variablename startvalue incrementvalue stopvalue*

`SWEEP variable` *var1 start1 inc1 stop1* `variable` *var2 start2 inc2 stop2*

Global parameters currently supported are: `frequency`, `gpoverhang`, `segmentsize` and `temperature`.

Layer parameters supported are: `bias`, `segmentsize`, `hfilaments`, `lambda`, `sigma`, `mur` and `thickness`.

Variables used in `SWEEP` commands must be defined in an .IXI input file as described in Section 10.1.

Examples are:

```
SWEEP frequency 1e10 1e10 1e11
SWEEP layer M1 segmentsize 2.0 -0.2 1.0
SWEEP layer M1 lambda 0.07 0.01 0.1 layer M1 bias -0.2 0.1 0.2
SWEEP variable d 10 1 20 variable w 5 5 10
```

Ports

12.1 Ports for inductance extraction

12.1.1 Definition of ports and terminals

For inductance extraction, port definition is a very important aspect of modelling. A good understanding of how ports are used in the extraction process and how port definitions propagate to extraction models in *InductEx* is thus necessary for accurate modelling.

For basic inductance calculation, voltage ports are defined in a model, as shown in Fig. 12.1. These ports are excited with a nominal voltage of 1 V at the specified frequency.

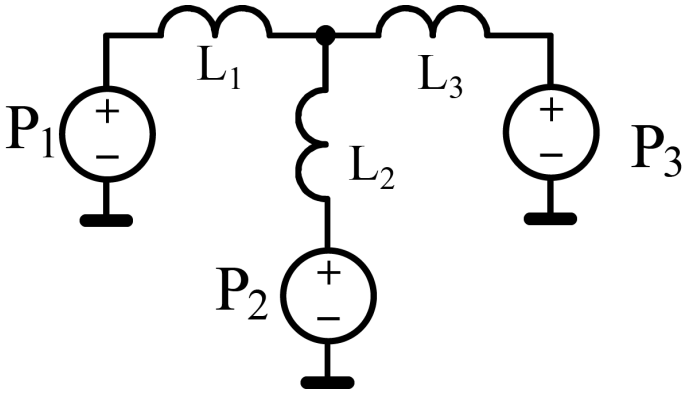

Figure 12.1: Example of how ports (nominally voltage sources) are connected in *InductEx* models for parameter extraction.

A port must have two terminals – positive and negative – but each terminal can be connected to more than one terminal object. Such terminal objects are defined on the TERM layer.

12.1.2 Terminal geometries of inductance ports in integrated circuit models

InductEx is primarily aimed at integrated circuit structures. The standard definition of a port is thus a single Term object that is applied to every layer to which the port is connected. Such a vertical stack may have multiple layers assigned to each terminal.

Fig. 12.2 shows an illustration of several types of inductance port definitions on layouts processed with *InductEx*.

Ports P_{sl1} and P_{sl2} are defined with a wire/path object on the Term layer that exactly coincides with the edge of stripline conductor in layer M1. The Term layer object should be a single line segment with the end vertex coordinates matching the vertices of the stripline conductor.

Port P_{via1} does not need a terminal object, and can be used for vias or similar structures such as planar Josephson junctions. The smallest via inside which the port label coordinates fall is elevated to the Term layer and used to define the port geometry on every layer linked to the port. All other vias inside the stack are deleted during modelling. In this example, the positive terminal is connected to layers M3, M2 and M0, while the negative terminal only connects to layer M1.

Port P_{cpw1} is defined with three labels; each connected to a line on the Term layer at the edge of the coplanar waveguide layout objects. The centre conductor is connected to the positive terminal, while the ground conductors on either side are connected to the negative terminal.

Finally, a polygon port is used for one of the negative terminal objects of Port P_{cpw2}. Since version 6.1, such a polygon may coincide with and edge of the conductor polygon, or exceed the boundaries of the conductor polygon.

For tetrahedral meshing, a polygon port terminal is always defined in the bottom surface of the conductor to which it is linked.

12.1.3 Ports drawn on layouts

In order to provide port/terminal information to *InductEx*, a Term layer is needed. Any layer can be used, as long as it is called TERM, and the layer number or index corresponds to the parameter `TermLayer` in the layer definition file.

Terminals are drawn as polygons, boxes or paths in the terminal layer. It is not necessary to draw terminals over vias or Josephson junctions, as *InductEx* will automatically use the via or junction boundaries to determine the terminal dimensions. Therefore, it is mostly only necessary to draw terminals as lines on the edges of input/output lines. Collapsed boxes (zero width) can be used to draw line terminals, but these do not conform to proper GDSII protocol and might not be exported correctly by most programmes. **When using GDSII files, it is better to draw line terminals as paths or 2-point polygons** on the TERM layer. The path width is immaterial, as InductEx only uses the centre line as the actual terminal (thereby forcing a zero width path). Path width is therefore only used as a visual aid during layout.

12.1.4 Port or terminal label declarations on layouts

Ports or terminals are declared in the Text layer, and their geometries defined in the Term layer. The coordinates of a text label is used to match the port text to a structure in the Term layer. The standard syntax is:

`P`*name* [*postermlayers*] [*negtermlayers*] *value*

`P`*name*`+` [*postermlayers*]

`P`*name*`-` [*negtermlayers*]

`P`*name*`/` *termlayer* *direction*

The `P` character is the port identifier. Other valid identi-

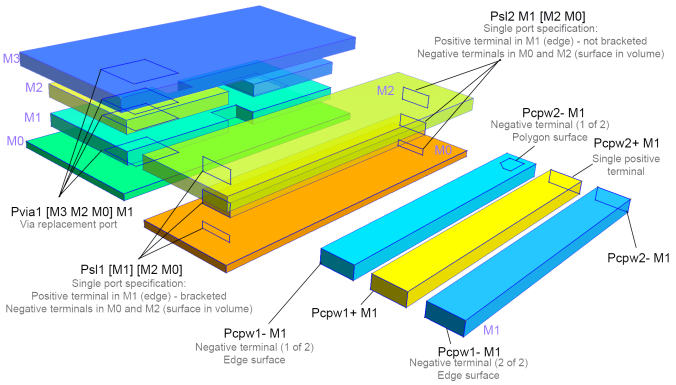

Figure 12.2: Illustration of different methods to define ports with *InductEx*.

fiers are `I` to identify current source and `B` and `J` to identify Josephson junction ports. Hole ports use the `F` identifier.

If the optional suffix characters `+` or `-` are added to the first terminal text, it indicates that this is only one terminal of the port and specifies the polarity (positive or negative). If the suffix is in {`\ | /`}, the port is a virtual cut. [*postermlayers*] is the space-separated list of names of the layers on which the terminal (defined by the polarity) is located, or the positive terminal if the polarity is not specified. If polarity is not specified, the list of layer names [*negtermlayers*] is required for the negative terminal. For a virtual cut port, *termlayer* is limited to a single layer, while *direction* defines the axis along which the port is connected to a conductor (*x* or *y*, which should coincide with the predominant axis of current flow) and the direction of the positive terminal. The virtual cut opens a slit in the conductive structure on which it is defined, and should run edge-to-edge. The sides of the slit are exactly `SegmentSize`/2 from the centre of the virtual cut line.

Examples of valid labels are:

`P1 M2 M0` - Port P_1, with positive terminal in layer M2 and negative in M0.

`PTWO [M1] M0` - Port P_{TWO}, with positive terminal in layer M1 and negative in M0.

`P3 M1 [M2 M0]` - Port P_3, with positive terminal in layer M1 and negative in M2 and M0.

`P4+ M2` - Positive terminal of Port P_4, in layer M2.

`P4- M0` - Negative terminal of Port P_4, in layer M0.

`P5/ YBCO +y` - Port P_5, a virtual cut through a conductor in layer Ybco, with current flow and port connection along the *y* axis. The positive terminal is in the positive direction along the *y* axis.)

`P6/ M3 -x` - Port P_6, a virtual cut through a conductor in layer M3, with current flow and port connection along the *x* axis. The positive terminal is in the negative direction along the *x* axis.)

`J7 M2 M1` - Port J_7, with positive terminal in layer M2 and negative in M1. Typically used for Josephson junctions.

`I8 M2 M0` - Port I_8, with positive terminal in layer M2 and negative in M0.

`Fnine` - Fluxon port F_{nine}, flux in positive *z* direction.

If the port text label coordinates fall inside or on the boundary of an object in the Term layer, this is used as the port definition.

When no Term layer object can be linked to the port label coordinates, and if the positive and negative terminals were defined in the same label, *InductEx* searches for a via object (with `MASK = -1`) that encloses the label coordinates and is ordered between the superconducting layers on which the terminals are declared. If more than one via object satisfies these conditions, the one with the smallest area is used. When such a via is used as a terminal object, it is moved to the Term layer (and should show up as such when a GDSII output file is generated with the `-d` option). For convenience, it is therefore not necessary to define a terminal layer object over a Josephson junction when it is to be used as a port.

Terminal geometries may be defined anywhere in on the edge or inside of a superconductive layer object.

Currently, *InductEx* also accepts text labels on layers other than the text layer when reading in terminal declarations. However, terminal text must be placed in the top cell of a layout, unless the `-X` switch is used to force text propagation up the cell hierarchy.

For instances where text labels are not supported by a layout tool (such as WaveMaker), or where it is required for convenience to alter port labels without the use of a layout tool, port terminals can be defined in a text file. This is discussed in Section 10.3.

12.2 Conductors (Ports) for capacitance extraction

For capacitance extraction, each conductor is identified with a label on the layout. The label must start with the character "C", followed by any name.

Conductive objects that are linked together in a layout through metal contacts or vias are linked the same capacitance conductor.

12.3 Ports for impedance extraction

Impedance extraction uses ports similar to those used for inductance extraction, but the requirements are more strict. Currently, only two impedance ports are allowed. The first two ports read in by InductEx are used, and the rest discarded. These ports should be on the outer edges of conductive objects. Line terminals are advised.

Off-Chip Structures

13.1 Arbitrary structures from Gmsh

For off-chip structures that cannot be defined as layout objects in a planar integrated circuit construction, *InductEx* supports structure read-in from meshed files (triangles and tetrahedra, linked to physicals) in the Gmsh .MSH mesh file format.

13.1.1 Physical structures

Physical Surfaces are used to expose surfaces to *InductEx* for the purpose of connecting port terminals.

Physical Volumes are used to define parameters for the Volumes, with only sigma, lambda and relative permeability that can be defined.

13.2 Package modelling

InductEx allows the modelling of packaging for integrated circuits for the purpose of parameter extraction.

Currently, wedge-type and ball-and-stitch wire bonds are supported as geometric models, so that an integrated circuit can be modelled complete with wire bonding to a printed circuit board or chip carrier module. Capacitance extraction is not supported yet when wire bonds are present.

An example of such a chip and carrier package is shown in Fig. 13.1.

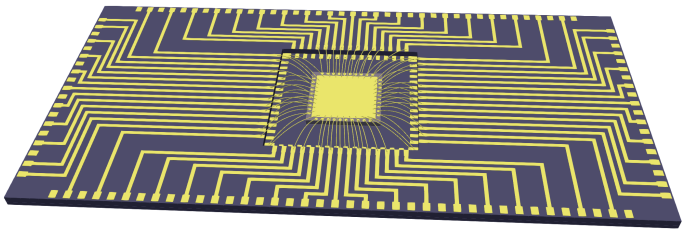

Figure 13.1: Example of a superconductor integrated circuit die mounted on a ceramic chip carrier and wire bonded to traces on multiple levels of the chip carrier.

Package modelling requires tetrahedral segments for the bonds, although chip layers can use triangles.

Package modelling requires setup in an IXI file, with every chip die or carrier printed circuit board imported from a GDS or IXI file with the `Die` record in an IXI file. Each chip of printed circuit board can have a different layer definition file to describe its process.

Bond structures are defined in the main IXI file. Bonds are connected between bond locations defined on die or printed circuit board layouts.

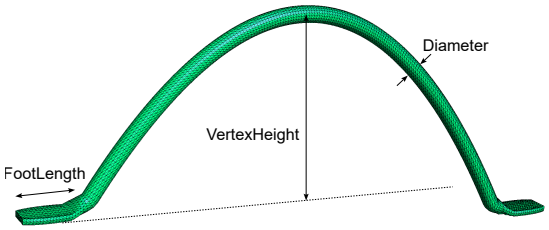

Figure 13.2: Wedge-type wire bond.

13.2.1 Wedge-type wire bonds

A wedge-type wire bond is shown in Fig. 13.2. The wire bond is modelled as a parabola, with the vertex placed at `VertexHeight` above the highest bond pad. Wedge-type wire bonds can be instantiated between any two bond locations on any dies, although the die surfaces must have the same normal. Currently, dies are required to lie flat (a normal of (0, 0, 1)) The die surfaces can have different z positions.

The wedge foot geometry is controlled by the foot length, spread ratio and thickness ratio of the foot setup, as well as the diameter of the wire bond setup. The dimensions are detailed in Fig. 13.3. A skin of outer segments is created to the depth of `EdgeSlice`, if that field is specified in the wire bond setup.

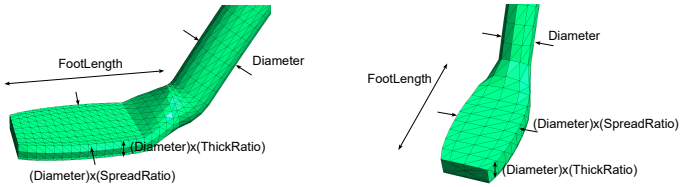

Figure 13.3: Wedge wire bond model showing wedge foot detail for “ThickRatio=0.3" on the left and “ThickRatio=0.5" on the right. Mesh size on the left is half of that on the right.

13.2.2 Ball-and-stitch wire bonds

A ball-and-stitch wire bond follows the EIA/JEDEC Standard No. 59, “Bond Wire Modeling Standard", published in June 1997. The geometry is shown in Fig. 13.4. The start foot must be of type `Ball` and the end foot must be of type `Wedge`. The stitch uses the same model shown in Fig. 13.3. The ball parameters are shown in Fig. 13.4. Currently, dies are required to lie flat (a normal of (0, 0, 1)) The die surfaces can have different *z* positions.

Dimensions coloured in black are directly accessible through the `Ball` and `WireBondSetup` records parameters, while dimensions coloured in grey are derived.

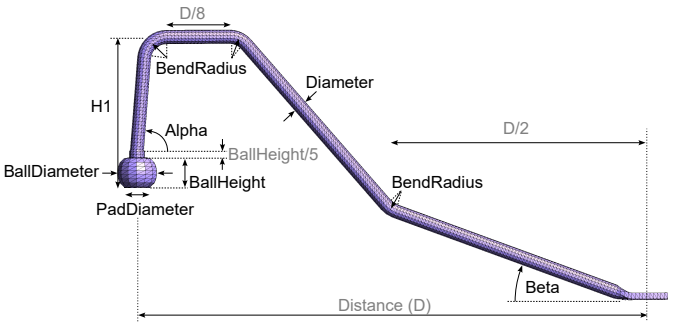

Figure 13.4: Ball-and-stitch wire bond model.

A skin of outer segments is created to the depth of `EdgeSlice`, if that field is specified in the wire bond setup.

13.2.3 Bump bonds

InductEx can model solder bump bonds as shown in Fig. 13.5, where the effects of the `MidDiameter` and `Diameter` parameters are shown. The bumps have the same diameter at the upper and lower connection. If `MidDiameter` is smaller than `Diameter`, the bump is concave.

The smoothness of a bump can be controlled by specifying the number of segments around the circumference with `CircumferenceSections`.

The bump height is determined by the distance between the die surfaces between which a bump is defined. A thin ring is slipped between each pad of the bump and the surfaces of the chips to which the bump connects to weld the bump surfaces to chip surfaces which may consist of multiple polygons and holes with unplanarized top surfaces.

A skin of outer segments is created to the depth of `EdgeSlice`, if that field is specified in the wire bond setup.

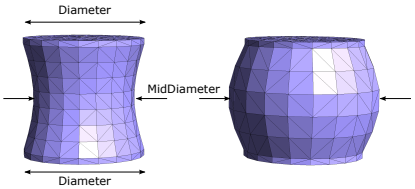

Figure 13.5: Concave and convex solder bump models. Both bumps have 20 sections around the circumference.

Output Files

InductEx creates several output files. The format of some of these files is presented here to aid the interpretation of results. File names can be prepended with the string defined with the `-w` command line parameter.

14.1 Solution output

The simulation result is always printed to the standard output as well as written to the file `sol.txt`.

14.1.1 Solution result

The simulation output file lists the results for a component such as impedance (each entry of which represents a single branch inductance and resistance) in a block that contains two head lines and the component entries in the following format:

`Inductance [`*unit*`]`
`Name Design Extracted AbsDiff PercDiff`
Elementname DesignVal ExtractedVal Absdiff Percdiff

InductEx lists the followings:

Elementname
is the name of the netlist element.

DesignVal
is the inductance design value if inductance is listed in the netlist. If no value is provided in the netlist, "`--`" is printed.

ExtractedVal
is the extracted inductance value in the units specified in the header line. If the extracted value is infinite (an open circuit), "`--`" is printed.

Absdiff
is the absolute difference between the extracted and design values of the element. If no design value is provided in the netlist, *Absdiff* is equal to the extracted value of the element.

Percdiff
is the percentage difference between the extracted and design values of the element. If no design value is provided in the netlist, "`--%`" is printed.

Outputs for resistance, mutual inductance and Josephson junction area have similar structure.

14.1.2 Mesh quality

The cuboid mesh quality when filament meshing is used is listed as a set of values in the line:

`Mesh info: dx =` *Mindx* `-` *Maxdx*`; dy =` *Mindy* `-` *Maxdy*`;`
`Worst L/W ratios: min =` *Minlw*`; max =` *Maxlw*

where:

Mindx
is the minimum dimension of any segment in the *x* direction, specified in units.

Maxdx
is the maximum dimension of any segment in the *x* direction, specified in units.

Mindy
is the minimum dimension of any segment in the *y* direction, specified in units.

Maxdy
is the maximum dimension of any segment in the *y* direction, specified in units.

Minlw
is the minimum length-to-width ratio of any mesh segment.

Maxlw
is the maximum length-to-width ratio of any mesh segment.

A high quality mesh is one for which the segment length-to-width ratios stay close to unity. If the mesh is of poor quality, the accuracy of current distribution calculations and subsequently the extracted element values degrades, and stability of the solver can be compromised.

If the length-to-width ratio falls outside the range 0.1 to 10, strongly consider smaller values for the `SegmentSize` parameter or careful grid snapping of layout object vertices.

14.1.3 Quality of the singular value decomposition for inductive networks

InductEx calculates the extracted values for inductance, resistance and mutual inductance in inductive networks with the singular value decomposition technique. Feedback on the quality of the solution is provided as:

`SVD solution information (A*x = b):`
`- Unknowns =` *unknowns*`; rank =` *rank*`.`
`- Condition number:` *K*
`- Max error (Max(|A*x - b|)):` *maxerror*
`- Relative error (||A*x - b||/||b||):` *relerror*

The quality indicators are:

K
is the result of dividing the largest singular value by the smallest one.

unknowns
is the number of unknown element values to be solved, which is the sum of all inductance and resistance components and all mutual inductances.

rank
is rank of the branch current matrix.

maxerror
is the worst error in any port voltage calculated with the extracted element values and the reference currents found with *TetraHenry*. For standard inductance calculation, unless the amplitude of any voltage source has been specified in the netlist file, the maximum value of any element in **b** is 1, and the maximum error should be only a small fraction of 1.

relerror
is the norm of the error vector (extracted element values multiplied by the branch current matrix) divided by the norm of the loop voltage vector **b**, expressed as a percentage.

For solutions with high precision, *K* should be close to unity. The precision of all solutions degrades by one significant digit for every order of magnitude in the value of *K*. **If *K* exceeds 1000**, consider that the circuit model (netlist) may be wrong. **If *K* approaches 1E6 or above, treat all results as suspect.**

If *rank* is lower than *unknowns*, treat all results as suspect and consider altering the circuit netlist, unless fluxon or external magnetic field coupling is enabled.

Maxerror does not provide any information on the error or loss of accuracy in any given extracted element value. It only shows the largest error in the voltage at any port between what was calculated with the numerical engine (and thus used in the singular value decomposition) and that which is found from multiplying branch currents with the extracted element values. *Maxerror* should be much smaller than 1 (below 0.05, typically) for purely inductive networks that have no mutual inductances to indicate a high quality solution. In some voltage loops with much lower total impedance than other loops, *Maxerror* can be inflated.

The relative error *relerror* is more useful. *Relerror*, expressed as a percentage, should be at most 5%. If it exceeds about 10%, then it is highly likely that parasitic coupling or some strong mutual inductance is present in the layout that was not accounted for in the netlist. There is no clear limit for *relerror* above which results cannot be trusted, but for circuits without significant mutual inductance it is strongly advised that all results be treated as suspect if *relerror* exceeds about 10 %. The error can be reduced significantly by the inclusion of the correct mutual inductance elements in the circuit netlist, or it might indicate a bad agreement between the netlist and the layout.

If all mutual inductance is accounted for correctly, and there is a good match between netlist and layout, *relerror* should be comfortably below 0.1%. A very good match produces *relerror* smaller than 0.01%.

Field Plotting

InductEx allows the plotting of magnetic and electric fields when volumes or planes are specified.

A field plot can be generated when *InductEx* is executed in inductance extraction or capacitance extraction mode with the *TTH* engine.

In MQS or EMQS mode, request `Plot [B, H, J]` in the IXI `$COMMAND` record to generate magnetic flux density B in Tesla, magnetic field strength H in A/m, and current density J. In EQS mode, request `Plot [E, D, Q]` for electric field E in V/m, electric displacement D in C/m^2, and charge Q. Select only the plots required. Multiple requested plots are generated independently.

The legacy command line switches `-M` (H), `-MB` (B), `-E` (E), `-ED` (D), and `-Q` (Q) remain supported with a migration warning; use IXI commands for new workflows.

15.1 Viewing fields in Studio

InductEx Studio opens the solver's legacy ASCII VTK scalar and vector outputs directly in its 3D canvas. ParaView is not required for these views.

Run the IXI extraction in Studio, refresh the project's Results page, and open a generated VTK output. Alternatively, open an existing VTK file through File/Open. In the viewer's Field visualization panel, select the array for the required excitation, frequency, and real or imaginary part, choose a display, and click Apply field view.

Heatmap
Colour by scalar value, vector magnitude, or an X/Y/Z component, with an automatic or manually specified range. Supported GPU colouring retains a CPU fallback. A field volume initially shows its exterior; use field planes to inspect a particular interior section.

Directional arrows
Equal-length arrows show the full vector direction; colour shows the selected magnitude or component. Density and arrow size control readability.

Streamlines
Traces follow the selected vector array in both directions through the sampled mesh. In a volume they follow the 3D field. On a field plane they follow its tangential projection, so a purely normal field has no in-plane streamlines: use arrows for that case. A projected plane trace does not describe the field away from that plane.

Heatmaps can also be combined with arrows or streamlines. Streamlines use deterministic mesh-based seeds, linear point-data interpolation (constant values for cell data), and fourth-order Runge–Kutta integration. Traces stop at the domain boundary, zero or invalid vectors, closed loops, or the length/step limit. They do not extrapolate across missing samples. Density selects up to 32, 96, or 256 seeds; each direction is limited to 800 steps and twice the longest mesh extent.

For AC results, each real or imaginary vector array is viewed separately. Vector magnitude is the magnitude of that selected array; it is not a combined complex amplitude or a time animation. Streamlines are spatial traces of that selected component, not particle trajectories.

Show model mesh adds a grey, uncoloured context overlay when the corresponding model MSH file is beside the field output. It does not apply the field values to the conductor mesh. Use Solid to return to geometry, and the viewer's snapshot export to save the view. The shipped `examples/fields` directory contains magnetic plane/volume and electric plane examples.

15.2 Volumes

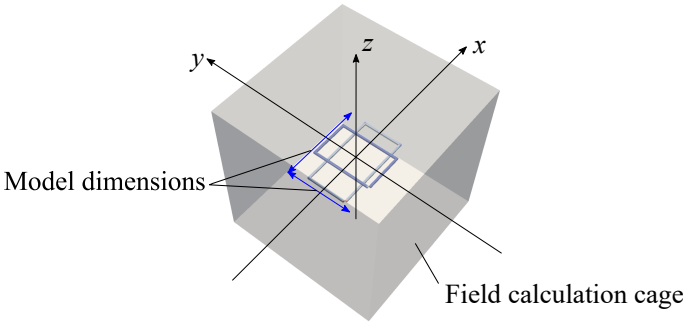

Figure 15.1: Field calculation volume around an extraction model.

The default field plot is generated over a cuboid volume that surrounds the extraction model, as shown in Fig. 15.1. The volume is centered around the centre of the extraction model, with side lengths in the *x* and *y* axes equal to the largest dimension of the extraction model multiplied by the scaling factor `FieldCageScale`. The cuboid volume's height in the *z* direction is derived from the side length in the *xy* plane multiplied by the vertical scaling factor `FieldCageZScale`.

The cuboid volume is meshed and field values are calculated at every node. The mesh size is specified with `FieldMeshSize`, which has default value of 2 Units.

Results are written into the output directory as `B_<model>.vtk`, `H_<model>.vtk`, `E_<model>.vtk`, or `D_<model>.vtk`; current and charge plots use `J_<model>.vtk` and `Q_<model>.vtk`. An optional output prefix precedes these names. The array names retain the excitation, frequency, and real/imaginary identity supplied by TTH.

15.3 Planes

Field calculation over a volume is computationally expensive. For quick analysis or symmetrical structures it is often sufficient to view the field in a plane that slices through the space that contains an extraction model.

A plane can only be defined in the IXI input file as an FPlane record (described in Section 10.1.5). A plane is defined as a rectangle between two opposing corners in the *xy* plane, after which it is translated and rotated as specified.

Multiple planes can defined for one extraction model. Fields are then calculated over each of the planes. The mesh size for all planes is specified with `FieldMeshSize`.

When any plane is defined, it overrides the field volume calculation.

All requested planes are included in each field's `<field>_<model>.vtk` output. The field-plane meshes are written as `fieldplane0.msh`, `fieldplane1.msh`, and so on. Older VTK field files can still be opened directly in Studio.

III Command Line Reference

16 Command Line Parameters 70

17 Exit Codes 71

Bibliography 73

Index 74

Command Line Parameters

16.1 *InductEx* command line paramters

16.1.0.0.1 Current InductEx 8 behaviour.

Use Studio and an IXI project for new work. Existing legacy command-line workflows remain supported and emit migration guidance where applicable. The earlier transition description retained below describes the IXI-first interface; its statement that all simulation flags are rejected does not apply to the restored compatibility path. A normal IXI invocation is `inductex project.ixi`. Use `–help` for the current options, `–version` for the version and `–build-info` for build details. The GUI equivalent is to open the project and select Run full extraction.

16.1.1 Earlier IXI transition description

The following text records the earlier interface transition. Use the current behaviour above when running InductEx 8.

Starting with *InductEx* 8.0, command line parameters for control of a simulation are not accepted anymore. All control is now exercised through the COMMAND block in the main IXI file.

InductEx now takes a single command line parameter as the main input file (of type IXI).

The only other command line parameters supported are:

-h
: Prints help blurb to screen.

-id
: Lets *InductEx* write out the system identifier information to `SM_sysID.txt`.

-r
: Recalculates the inductance, coupling and resistance solutions (for MQS mode) for the supplied project file without remeshing and running *TetraHenry*. Useful if a change to the netlist was made to improve the SVD solution error.

Exit Codes

Exit codes are printed in the exit message to the standard output. The comprehensive list, with some solutions where applicable, is:

0
Clean exit.

1
Error while trying to open layout input file (GDS or IXI). The file does not exist, or cannot be read.

2
Error while trying to open port label text file. The file does not exist, or cannot be read.

4
Error while trying to open process (layer definition, LDF) file, or process file not defined.

5
Error while trying to open netlist file.

6
Error while trying to open *FFH/TTH* port current or capacitance result file. Engine did not execute correctly (examine `ffh.log` or `tth.log`) or directory has write restrictions.

7
Error while trying to open mesh file.

8
Error reading solver output file `solver_out.txt`. Check if solver is in the path.

9
The input file format is not supported anymore (CIF or DXF).

10
InductEx cannot create an output file. Check of working directory has write permission.

20
Cannot resolve all inductor (branch) currents when solving inductance; no solution possible.

30
No parameter block defined in LDF file, leaving modelling parameters undefined. Check if parameter block starts with `$Parameters`.

31
Invalid input format or syntax in LDF file. `$End` statement found without an appropriate opening statement.

32
Duplicate layer number in LDF file, or no layer number defined in layer block. Layer number ambiguity could cause modelling errors and must be corrected.

33
Process uses a ground plane, but ground plane layer not identified. Add the parameter `gplayer=`*n* to the LDF file, where *n* is the ground plane layer's GDS number.

34
Capacitance conductor layer definition invalid, or does not support conductors (should be FilmType `S` or `R`).

35
Duplicate layer name in LDF file, or no layer name defined in layer block. Layer name ambiguity could cause modelling errors and must be corrected.

36
Superconductor layer has zero or negative lambda.

37
Duplicate layer order in LDF file, or no layer order defined in layer block. Layer order ambiguity makes it impossible to construct a layer stack.

38
Missing Text or Term layer, or layer number non-fabrication layer number collision.

39
Range check error – length of sorted and unsorted layer arrays mismatch.

40
Component (such as inductor or port) name duplicated in circuit netlist file. Remove ambiguity by renaming one of the components (case-insensitive).

41
Error in port definition in netlist, such as a missing negative terminal.

42
Cannot link positive or negative terminal of a port to a layout figure. Check layer names in port labels or in LDF file.

43
Port positive or negative terminal layer definition invalid, or does not support terminals (should be FilmType `S` or `R`).

44
Cannot identify port surface while building Gmsh geometry file.

45
Component in parallel with another of its kind (inductor-inductor or port-port). This cannot be resolved for extraction.

46
No cycles in netlist, or bad cycle found. Check for floating inductors or resistors, where one or more nodes are not connected to the circuit.

47
There is a mismatch between the number of ports (including magnetic field sources) in the netlist, layout and current output file (`ix.cur` or `ixth.cur`) when inductance is solved.

48
A port misses a terminal in layout.

49
Port defined in netlist not found in layout.

50
Layout object (e.g. path, poly, sref) outside of Struct record in IXI file. Structure cannot be resolved.

51
Coordinates not specified for a layout object in IXI file.

52
Non-unity SREF magnification in layout input file not supported by *InductEx*. Model geometry will not match layout.

53
Collapsed polygon in layout.

54
No objects in layout, either because layout file is empty or objects are all deleted by cut layers or operators, or were declared on layers not defined in the LDF file. A 3D extraction model cannot be built.

55
Duplicate terminal labels in layout at different locations - ambiguity cannot be resolved.

56
Invalid record length in GDS file - read-in aborted.

57
Structure name in GDS file exceeds 127 characters.

58
Port terminal defined on layer with no objects.

59
Record or command not recognised in IXI file.

60
No structures read from input an IXI layout input file. This mostly happens when an IXI file from before v6.00 is used, and no STRUCT records are defined.

61
Virtual cut terminal cannot slice conductor object into two separate polygons (it either does stretch edge-to-edge, or cuts conductor polygon multiple times).

62
Layout cannot be identified in multiple-layout IXI input or for linking bonds.

63
IXI record not supported when cuboid meshing is used.

64
Port terminal defined as oblique line (not in *x* or *y* direction) with a cuboid mesh, which is not allowed.

70
Abnormal termination or failed execution of external module (such as *TTH* or Gmsh).

82
Incompatible format in Solver output file `solver_out.tx┘t`. Use the solver version packaged with this version of *InductEx*.

90
Different capacitance conductors linked to the same conductive structure.

91
No capacitance conductors identified.

110
Licence key tier does not support requested functionality (as listed in the exit message).

111
Licence key error.

112
Licence key is for older version of *InductEx* , and is not supported anymore.

113
Invalid licence.

114
Cannot decrypt protected process file.

120
Mesh file not found. Gmsh did not complete mesh. Cannot execute *TTH* without a mesh.

190
No valid edge found in fundamental circuit construction. Check netlist.

191
Positive and negative node the same for an edge (component) in fundamental circuit construction. Check netlist.

192
Missing node declaration for an edge (component) in fundamental circuit construction. Check netlist.

200
Component lookup returned invalid dynamic array index - will result in memory error.

210
Syntax error in expression string (such as missing parenthesis).

211
Cannot parse variable in expression string. Variable identification symbol might be absent, or variable not defined.

212
Syntax error boolean expression string.

220
No I_C read for Josephson junction in JoSIM netlist.

230
No filament info found in *FFH* mesh file - check `ffh.log`.

231
Exception – positive or negative terminal of a port has no nodes. This is an *InductEx* node-building error for cuboid meshes and should be reported if encountered.

240
Construct construct type not defined in LDF file.

255
Catch-all code for entry into prohibited code (typically during range check error debugging).

Bibliography

[1] C. J. Fourie, O. Wetzstein, T. Ortlepp, and J. Kunert, "Three-dimensional multi-terminal superconductive integrated circuit inductance extraction," *Superconductor Science and Technology*, vol. 24, no. 12, p. 125015, Nov 2011.

[2] C. J. Fourie and W. J. Perold, "Simulated inductance variations in rsfq circuit structures," *IEEE Transactions on Applied Superconductivity*, vol. 15, no. 2, pp. 300–303, June 2005.

[3] C. J. Fourie, O. Wetzstein, J. Kunert, H. Toepfer, and H.-G. Meyer, "Experimentally verified inductance extraction and parameter study for superconductive integrated circuit wires crossing ground plane holes," *Superconductor Science and Technology*, vol. 26, no. 1, p. 015016, Nov.

[4] C. J. Fourie, O. Wetzstein, J. Kunert, and H. Meyer, "Sfq circuits with ground plane hole-assisted inductive coupling designed with inductex," *IEEE Transactions on Applied Superconductivity*, vol. 23, no. 3, p. 1300705, June 2013.

[5] C. J. Fourie, "Full-gate verification of superconducting integrated circuit layouts with inductex," *IEEE Transactions on Applied Superconductivity*, vol. 25, no. 1, pp. 1–9, Feb 2015.

[6] K. Jackman and C. J. Fourie, "Flux trapping analysis in superconducting circuits," *IEEE Transactions on Applied Superconductivity*, vol. 27, no. 4, pp. 1–5, June 2017.

[7] K. Jackman and C. J. Fourie, "Flux trapping experiments to verify simulation models," *Superconductor Science and Technology*, vol. 33, no. 10, p. 105001, aug 2020. [Online]. Available: https://doi.org/10.1088/1361-6668/aba79b

[8] C. J. Fourie and K. Jackman, "Experimental verification of moat design and flux trapping analysis," *IEEE Transactions on Applied Superconductivity*, vol. 31, no. 5, pp. 1–7, 2021.

[9] K. Jackman and C. J. Fourie, "Tetrahedral modeling method for inductance extraction of complex 3-d superconducting structures," *IEEE Transactions on Applied Superconductivity*, vol. 26, no. 3, pp. 1–5, April 2016.

[10] C. Geuzaine and J.-F. Remacle, "Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities," *International Journal for Numerical Methods in Engineering*, vol. 79, no. 11, pp. 1309–1331, 2009.

[11] S. Nagasawa, T. Satoh, K. Hinode, Y. Kitagawa, M. Hidaka, H. Akaike, A. Fujimaki, K. Takagi, N. Takagi, and N. Yoshikawa, "New nb multi-layer fabrication process for large-scale sfq circuits," *Physica C: Superconductivity*, vol. 469, no. 15, pp. 1578 – 1584, 2009, proceedings of the 21st International Symposium on Superconductivity (ISS 2008).

Index

A

Array 40

B

Back-annotate 59

C

Compact model 59
Current source 61

E

Error
  Bound 66
  Maximum 66
Extraction boundary 57

G

GDS 56

H

Holes 58
  Port 61

I

Infinite extracted value 65
Installation
  Download 12
  Linux 14
  macOS 14
  MS Windows 14
IXI
  AbsTol 46
  Analysis 46
  Ball bond 49
  Bump bond 49
  Commands 46
  Component 49
  Controls 49
  Cores 47
  Debug 47
  Die 50
  EqualSegments 47
  Excitation 47
  ExportFile 47
  Expressions 55
  Fidelity 47
  Fplane 50
  Frequency 47
  GDS 51
  HideTerminal 47
  HoleBridge 47
  ImportLayout 51
    Top, 51
  ImportMesh 51
  Iterations 47
  Licence 47
  Log 47
  Mask 52
  Material 52
  MeshFile 47
  MeshType 47
  Mode 48
  Netlist 48, 52
  NoNetlist 48
  Path 52
  Physical 53
  Plot 48
  Poly 53
  Port 53
  Prefix 48
  Process 48
  ProcessKey 48
  Records 49
  Resistance 48
  Set 49
  SParam 53
  Sparsity 48
  Sref 54
  Struct 54
  TetraHenry 48
  Text 54
  Tol 48
  Top 48
  TopLevelTextOnly 48
  Variable 54
  Verbosity 49
  Wedge bond 54
  Wire bond 55

J

Josephson junction 58–61

L

Layer definition file 39
LDF
  Global parameters 40
  Layer parameters 43
  MeshMaskLayer 45
  Operators 45

M

Modelling parameters 39

N

Netlist . . . 58
Circuit elements . . . 58
Comments . . . 58
Control block . . . 58
Ports . . . 58
Set . . . 59
Sweep . . . 59
Title line . . . 58

O

Operator precedence . . . 56
Operators
Declaration . . . 57

P

Packaging . . . 63
Physical surface . . . 63
Physical volume . . . 63
Port
Virtual cut . . . 61
Ports
Draw on layout . . . 60
Hole . . . 61
Identifier . . . 60
Identifiers . . . 58
Label input file . . . 56
Labels on layout . . . 60
Netlist definition . . . 58
Terminal object . . . 61
Process file . . . 39
Comments . . . 39
Header . . . 39
Open . . . 39
Protected . . . 39
Record block . . . 40
Records . . . 40
Process paramters . . . 39

S

Set command . . . 59
Source
Type . . . 58
SPICE . . . 58
Compact model . . . 59
Studio
Field plots . . . 27
GDS viewer . . . 24
Getting started . . . 16
Help . . . 31
IXI projects . . . 19
Mesh viewer . . . 27
Preferences . . . 31
Process view . . . 24
Results . . . 27
S-Parameters . . . 28
Sweep command . . . 59

T

Technology file . . . 39
Terminal
Geometry . . . 61

W

Wire bonds . . . 63