

# PyCell Studio, Version 3.1.0, Release Notes

## Table of Contents

Introduction.....	1
New Features.....	1
Changes.....	3
Change Requests.....	3
2006-06-02.....	3
2006-05-26.....	4
2006-05-17.....	4

## Introduction

Welcome to PyCell Studio, Version 3.1.0, released 2006-06-02.

See Changes Requests for specific differences from prior releases of Version 3.1.0.

These release notes summarize the important enhancements, bug fixes, and other changes which have been made since Version 3.0.1. For complete technical details, please see the related documentation.

## New Features

1. New DRC capability  
PyCell Studio adds integrated design rule checking. PyCell designers can quickly and easily verify their PyCells. Design rules and commands are contained in the Santana.tech technology file. Supported DRC operations include common boolean layer operations (AND, OR, etc.) and constraint rules (WIDTH, SPACING, AREA, etc.). Example of the new technology file format provided in CNI 130b technology files.
2. fgPlace() - smart function made smarter  
fgPlace() enhanced to understand complex spacing constraints. DRC commands which are used to verify PyCell designs are brought forward into layout construction with fgPlace(). Rules requiring multiple steps for derivation are supported.
3. Pyros
  - Ciranova Layout Viewer gets a name, Pyros.
  - `cngui` command has been renamed to `pyros`.
  - New DRC capability invoked with the green double checkmark toolbar button.
  - DRC panel for reviewing DRC results.
  - Messages panel for errors, warnings, and other information.
4. Rewritten Grouping class
  - Old Grouping class used a method, `getGrouping()`, which acted both to construct and retrieve the object. This ambiguity was error prone. A separate constructor, `Grouping()`, is introduced which is consistent with other class constructors.
  - New Grouping no longer allows members to exist in different Groupings simultaneously.
  - Groupings can now be nested. Previously, adding a Grouping to another Grouping meant adding members of one Grouping to another. Now, Groupings can hierarchically contain other Groupings.
  - The special case invocation, `getGrouping()` with no name, which included all current

PhysicalComponents, has been removed. Instead, use either DloGen.makeGrouping() or Grouping( "name\_of\_grouping", DloGen.getComps()).

- DloGen.addToGrouping() has been removed. Instead, use Grouping.add() method.

#### 5. User-defined CompoundComponents

CompoundComponent class was originally designed for implementation of Ciranova smart objects written in C++, such as ContactRing. This class has been extended with methods, lock() and unlock(), to enable Python authors to design their own custom CompoundComponents.

#### 6. Choppable ContactRing

ContactRing class enhanced to support ContactRing.chop() method, which enables a geometry on a routing layer to pass through the ContactRing.

#### 7. New smart fg\*() functions

- Boolean operators, DloGen.fgAnd(), DloGen.fgOr(), DloGen.fgNot(), DloGen.fgXor(), DloGen.fgSize(), added to conveniently generate complex shapes.
- DloGen.fgAddEnclosingPolygon() introduced for creating contoured shapes, analogous in functionality to DloGen.fgAddEnclosingRects().

#### 8. Stretch handle support

Ciranova layout viewer, Pyros, supports stretch handles. PyCell authors can test their stretch handle code prior to releasing to layout designers using other layout editors. Support for stretch handles and device auto-abutment formalized in a Python package for integration with third party tools.

#### 9. Explorer mode

cnexp command added for quick and convenient experimentation with Python API.

```
usage: cnexp [options]
```

```
options:
```

```
-h, --help          show this help message and exit

--libname=LIBNAME  name of the OA library to be created
                   [MyTestingLib]

--libpath=LIBPATH  path to the OA library
                   [/tmp/MyTestingLib]

--cellname=CELLNAME name of the design to be create
                   [MyTestingCell]

--techfile=TECHFILE path to the techfile to be used
                   [%CNI_ROOT/tech/cni180/santanaTech/Santana.tech]
```

#### 10. IDE training mode

Ciranova menu in the IDE adds an entry, "New PyCell Project... (Training)", for a default setup for new users to learn the IDE.

#### 11. CNI 130 and CNI 130b

Sample **fictional** technology files for 130nm processes included.

- CNI180 – older Santana format, similar to what was released in Version 3.0.1

- CNI130 – older Santana format, new in Version 3.1.
- CNI130b – new Santana format, new in Version 3.1.

## 12. ArrayInstContact

ArrayInstContact is a memory efficient implementation of the Contact class, optimized for larger contacts/vias with many rows and columns of cuts.

## 13. InstanceArray

InstanceArray class added for memory efficient instantiation of instances.

## Changes

### 1. ShapeFilter class replaces Layerlist & boolean flag

Many class methods used two arguments, a Layerlist and a boolean flag, to designate layers which should be included or excluded from calculation. New ShapeFilter class introduced which replaces this construct. The ShapeFilter class provides for future extensibility, as well as better abstraction.

The following class methods and functions are updated to use the new ShapeFilter class.

- DloGen.getBBox()
- Instance.getBBox()
- Grouping.getBBox()
- Shape.getBBox()
- ContactRing() constructor
- DloGen.fgPlace() method
- getMinSpacing() global function

### 2. Iterators removed

The iterators for Ciranova classes, DloGen, Grouping, and CompoundComponent, as well as any derived classes have been replaced with methods which return a list; e.g. DloGen.iterComps() is replaced with DloGen.getComps(). Methods returning lists were found to be more useful.

### 3. Layer class

- Layer.getName() replaced with getLayerName().
- Layer.getNumber() replaced with getLayerNumber().
- Layer.getPurpose() replaced with getPurposeNumber().
- Layer.attribute layer.purpose replaced with layer.purposeNumber.

### 4. Tech class

- Tech.getSantanaLayers() replaced with getSantanaLayerNames().
- Tech.getLayers() enhanced to also accept purpose name or purpose number.

## Change Requests

Like many companies, Ciranova uses Bugzilla ([www.bugzilla.com](http://www.bugzilla.com)) to track change requests. If your request was assigned a Bugzilla number, you should have been notified by Ciranova technical support when it was closed. Otherwise, you can check the following listing of Bugzilla requests which were closed for this release.

### 2006-06-02

ID#	Description
521	discrepancy in DRC engine results report and displayed shapes numbers
541	Error message when running Wing IDE
566	Wing Ide Plugin: View only PyCells in this active files sometimes doesn't work

### 2006-05-26

ID#	Description
466	PyCell Studio releases to include additional OpenAccess libraries
467	PyCell Studio releases to include Python .h files
553	Release notes are missing from docs/ directory
554	Add a README or VERSION file to indicate software version
557	Missing bashrc & tcshrc from quickstart/ in 3.1.0 download
558	Missing env variable causes PyCellKit_cni180 to not export GDS2

### 2006-05-17

ID#	Description
19	Pcell gets unselected after "Apply"
56	Layout snap grid for different layers
65	addContactRing() does not produce minimum geometry for guard-ring
66	smart place() function needs to consider layers (and rules) to exclude
67	contact ring should be choppable
100	Performance issues for reading in large design databases for GUI
119	Need to document use model for Pcells with Open Access
165	Python wrappers for C++ functions with boolean and integer parameters
174	API enhancement to specify layer purpose
175	API enhancements to fully support "purposes"
197	The Python Lib object should export the C++ close() method
212	rotate*() methods should accept reference point
223	Group name should be unique among groups
227	DloGen.fgFill() method to be exposed and documented
231	Request for ChoiceConstraint to support tuples
237	Do not export dloGen::undo() method to Python API
238	Do not export DloGen::insert() method to Python API
246	API inconsistency: Contact DLO/anchor; Instance/getOrigin, setOrigin
280	ContactRing should create minimum ContactRing if no instances in design
281	Contact object should have method to return number of cuts
289	Request method to create two-dimensional arrays of Instances
291	Constraint support for one-sided limits

293 Problem in Python Wrapper for Route::checkLayerWidth() method  
294 Route class object can not use Python keywords for parameter names  
296 Python API destroy() method should not have any parameters  
297 DloGen methods should raise exception if Grouping does not exist  
298 Spec and change use model of Grouping object  
300 cngenlib -create destroys user's directory  
303 Better error message when constraint is violated  
320 Need conditional rules for CNI130  
322 Provide an iterator for Groupings in DloGen class  
336 Enhance GUI menu for Open Design for Edit vs for Viewing  
342 Typo in error messages for DLO Explorer  
343 Create minimal default Tech binding for cnpy to allow easy use of Python API  
351 cngui crashes when reading in transistor array test case  
353 Mirroring of some physical components not drawn correctly  
360 Iterators need more protection against crashes  
362 How to create instances non-PyCell masters?  
368 Install script for PyCell Studio 1/25/06 - Doesn't always correctly detect Python version

372 Contact abutment bug with abutDir=ANY  
374 Contact hangs in loop when using addLayers with no primary cut layer  
375 Contact.setMinCuts(4) not enforced for some Contacts  
382 Viewer & IDE menu presumes to create the OA library  
385 Please install PyLint and PyChecker tools  
388 Minor changes/typos in cngenlib error messages  
400 fgPlace() incorrectly using wide metal spacing rules  
401 Pcell plug-in coredumps applications when PYTHONPATH is incorrect  
414 DloGen Grouping All object should provide destroy() method  
417 use of ContactRing getContact() method can lead to core dump  
423 CiraNova Python interpreter not properly handling errors from Clone method  
427 Problems with AbutContact.setAbutViaSpaceFactor() method  
429 ContactRing getContact() method can cause Santana to crash  
432 disappearing stretch handle  
441 The Text.getImp() methods returns oaShape, not oaText object  
449 Segmentation fault when destroying rectangles  
450 Moving a Grouping() does not preserve relative locations of members  
455 Bar Contact iterator can lead to segmentation fault crash  
457 Grouping derived objects can not have any properties  
458 Dlo.getDloName() not properly wrapped for Python  
460 Please create cnversion utility  
461 Viewer moves objects instead of measuring with ruler  
464 Rename LayerFilter class to ShapeFilter  
465 Replace LayerList with ShapeFilter  
466 PyCell Studio releases to include additional OpenAccess libraries  
467 PyCell Studio releases to include Python .h files  
472 Provide specialized find() method for InstanceArray class  
473 Design closing dialog box should give a choice of saving the design  
478 Formalize cnreplay as part of released software  
479 Provide stretchHandle() and autoAbutment() in cni.integ  
481 Provide cnexp utility to simplify learning Ciranova Python API  
482 Provide training menu entry for Wing IDE integrated with Ciranova API  
483 Enhance Layout Viewer to support running Ciranova DRC

- 484 Enhance Layout Viewer to add message pane for errors, warnings, etc.
- 485 Enhance Layout Viewer to support stretch handles
- 489 Viewer support for configuration file needs to be documented
- 490 Add boolean operators - fgSize, fgAnd, fgNot, fgOr, fgXor
- 491 Expose fgAddEnclosingPolygon to Python API
- 492 New fgPlace() to replace existing fgPlace()
- 493 Enhance Santana.tech to support complex DRC rules