Molecular Inorganics in InChI

Extending the InChI for Molecular Inorganics: Implementation, Validation and Integration

Nauman Ullah Khan[1], Gerd Blanke[2], Jonathan M. Goodman[3], Ulrich Schatzschneider[4], Andrey Yerin[5], Richard Hartshorn[6], Felix Bänsch[7], Djordje Baljozovic[1], Jan Brammer[1], Sonja H. Herres-Pawlis[1]

1. Institute of Inorganic Chemistry, RWTH Aachen University, Aachen, Germany; 2. StructurePendium Technologies GmbH, Essen, Germany; 3. Yusuf Hamied Department of Chemistry, University of Cambridge, CB2 1EW, UK; 4. Institute of Inorganic Chemistry, Julius Maximilians University Würzburg, Germany; 5. Advanced Chemistry Development Inc; 6. University of Canterbury, New Zeeland; 7. Beilstein Institute, Frankfurt(Main), Germany

RDKit UGM 2026 Darmstadt

Take-home message: Molecular Inorganics incorporates chemically informed preprocessing into InChI, enabling the normalization of coordination and organometallic structures without the need to reimplement identifier generation.

1 Why was this needed?

InChI is a structure-derived identifier used across chemical databases and toolkits. Standard InChI was designed around organic chemistry, where ordinary localized covalent bonds dominate.

Historical limitation

Metal–ligand bonds were often disconnected before identifier generation. This has simplified the handling of molecular inorganics but made the InChI less informative.

Scientific problem

Preserve important metal–ligand connectivity when it defines identity
Avoid chemically inconsistent partial disconnections
Handle coordination and haptic bonding consistently
Keep established organic InChIs backward-compatible

Identifier, not storage format

2 Architecture: preprocessing layer

The implementation does not replace the InChI workflow. It classifies eligible structures, applies Molecular Inorganics preprocessing, and then passes the normalized graph to the existing InChI normalization and canonicalization pipeline.

InChI input: Molfile (V2, V3) → **Molecular Inorganics**

↓

Standard InChI workflow → **InChI/InChIKey**

Why this design?

A targeted preprocessing layer keeps the core identifier generator stable while allowing inorganic-specific chemistry to be handled before canonicalization.

3 Coordinative/dative and haptic bond input

Molecular Inorganics extends the input handling to coordinative and haptic bonds. This is important because the same organometallic compound can be encoded using more than one drawing convention.

Coordinative Bonds
Coordinative/dative bonds are parsed and internally modified into single bonds with valence-adapted atoms.

Valence handling
Coordinative and haptic bonds are internally turned into single bonds. The valences of the non-metal atoms are adjusted so that the hydrogen count of the atoms is preserved.

Haptic bonds
Haptic bonds described by ENDPTS/ATTACH information in the Molfile format are expanded into explicit metal–ligand coordination links. * Atoms (star or asterisk atoms) used in haptic encoding describe the geometric center of the haptic bond system.

Zeise salt: three depictions → one identifier

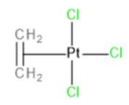
Depiction	Bond type(s) between C and Pt atoms	InChI w/o molecular inorganics	InChI with molecular inorganics
	Coordinative / dative bond	InChI options: Error 108 (no InChI; Unrecognized bond type: 9) inp	InChI=1B/C2H4Cl3Pt/c3-6(4,5)1-2-6/h1-2H2/q-1
	Haptic bond	InChI=1S/C2H4.3ClH.Pt/c1-2;;;;/h1-2H2;3*1H;/q;;;;+2/p-3	InChI=1B/C2H4Cl3Pt/c3-6(4,5)1-2-6/h1-2H2/q-1
	Angle bond with adapted valences of the carbon atoms to ensure correct hydrogen atom handling	InChI=1S/C2H4.3ClH.Pt/c1-2;;;;/h1-2H2;3*1H;/q;;;;+2/p-3	InChI=1B/C2H4Cl3Pt/c3-6(4,5)1-2-6/h1-2H2/q-1

Table 1 shows an example of Zeise salt with 3 different depictions.

With Molecular Inorganics enabled, coordinative/dative, haptic, and valence-adapted depictions of Zeise salt are normalized to the same InChI output.

4 Molecular Inorganics decision logic

The preprocessing uses a conservative decision tree. It only disconnects bonds that pass all chemical and graph-based checks; otherwise, the metal coordination environment is retained.

D1–D3 — Entry conditions
Ammonium handling, metal detection, and then bond-by-bond processing for metal centers.

↓

D4 — Eligible bond?
The bond must be a single bond and must not carry stereobond information.

↓

D5 — Metal valence in range?
Allowed valence and coordination states are checked against element-specific data.

↓

D6 — Binary disconnection rule?
Element-pair rules, based on electronegativity, determine whether disconnection is possible.

↓

D7–D8 — Multi-metal preservation
Bonds involved in links between metal centers are retained unless terminal-metal logic permits evaluation.

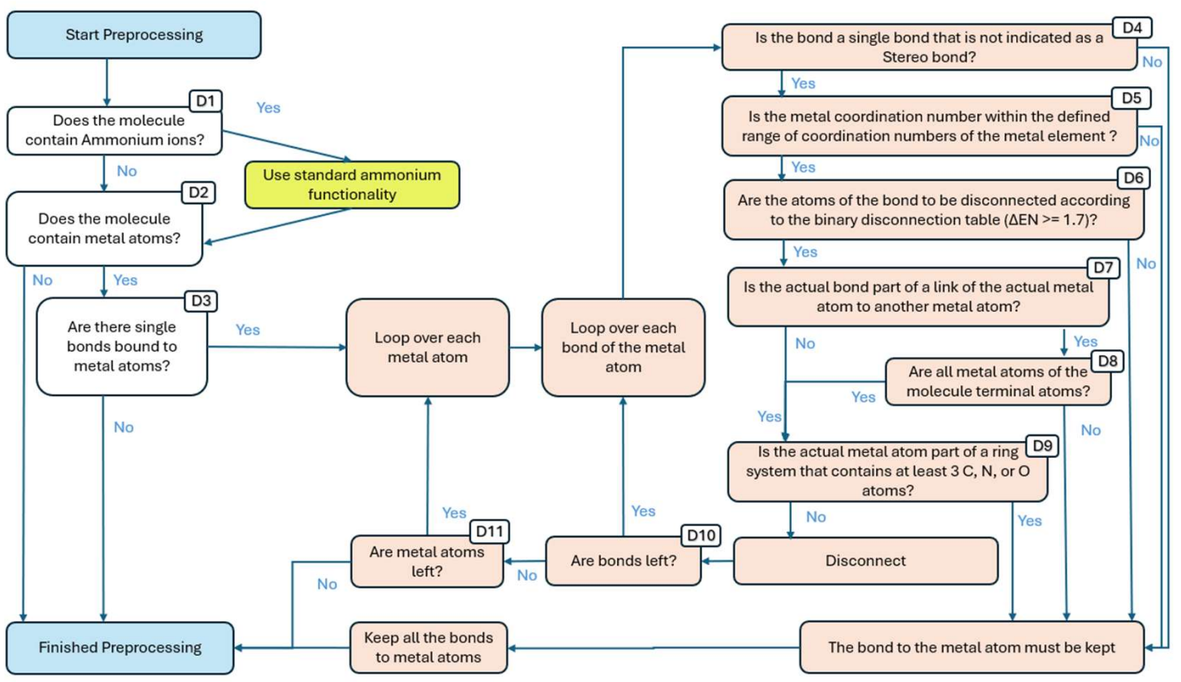
↓

D9 — Ring-system preservation
Metal atoms embedded in C/N/O-containing ring systems are protected from disconnection.

If every check passes → Disconnect
Otherwise → Keep the metal–ligand bond

Implementation focus

connected-component analysis via DFS
metal-count and ring-system protection
element-specific valence lookup table
symmetric graph and charge updates after disconnection

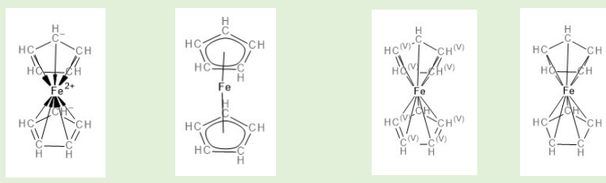


5 Representative outcomes

The same decision rules cover textbook cases and complex edge cases. The ferrocene examples below illustrate different reasons for keeping or disconnecting metal–ligand bonds.

Ferrocene
Metal–ring connectivity is preserved when criteria indicate that the organometallic framework should remain connected.

Organometallic depictions



InChI without Molecular Inorganics
InChI=1S/2C5H5.Fe/c2*1-2-4-5-3-1;/h2*1-5H;/q2*-1;+2

InChI with Molecular Inorganics
InChI=1SB/C10H10Fe/c1-2-4-5-3(1)11(1,2,4,5)6-7(11)9(11)10(11)8(6)11/h1-10H

Different organometallic drawings should not force avoidable differences in identifier behavior when they describe the same chemically relevant connectivity.

6 Internal testing and validation

The expanded implementation is now tested against curated examples that exercise chemical rules and software edge cases, not only simple textbook structures.

- ✓ **Decision-edge cases**
  what are the InChIs of the following compounds: NaCl, Na_2SO_4, VCl_3, $V_2O_2Cl_6$?
- ✓ **V3000 features**
  coordination bond type 9 and haptic ENDPTS/ATTACH data
- ✓ **Graph preservation**
  multi-metal components and ring-system protection
- ✓ **Backward compatibility**
  standard organic workflows remain unaffected by default
- ✓ **CI-ready workflow**
  tests can be run against versioned binaries and development builds

Validation goal
The output should change only when Molecular Inorganics intentionally changes the normalized molecular graph.

7 Relevance for RDKit workflows

RDKit users already rely on InChI generation through wrapper functions and linked InChI binaries/libraries. Molecular Inorganics follows the same model: toolkits can expose the new option without reimplementing the underlying chemistry.

RDKit molecule / MolBlock

↓

IUPAC InChI library + Molecular Inorganics option

↓

InChI / InChIKey for inorganic structures

Practical impact

Database registration can be recalculated for affected inorganic records
Existing organic InChIs remain stable by default

8 Scientific contribution

Molecular Inorganics release turns the long-standing representational limitation of metal atoms in the InChI into an implemented preprocessing workflow that provides a complete representation of metal bonds. It adds element-specific metal valence checks, graph-based bond preservation, and V3000 coordination/haptic bond handling before standard InChI canonicalization.

The result is a more useful identifier pathway for inorganic and organometallic chemistry while preserving compatibility with the infrastructure already used by databases and cheminformatics toolkits.

9 Test the InChI developments on Web-Demo

Test the InChI developments on Web-Demo

Availability: official IUPAC-InChI GitHub repository; versioned release; open community testing through the web demo and issue tracker.

Join the discussion on: https://github.com/IUPA-InChI/InChI/issues
Let us know how you have implemented the InChI into your software:

Funding and support:

NFDI4Chem
Bundesministerium für Bildung und Forschung
Funded by DFG Deutsche Forschungsgemeinschaft German Research Foundation NFDI4Chem is supported by DFG under project number 441958208
Finanziert von der Europäischen Union NextGenerationEU
Data Literacy Alliance
VolkswagenStiftung The Volkswagen Foundation, with the project Universal molecular syntax for sustainable machine learning applications.
InChITRUST

References: Heller et al., J. Cheminform. 2015; Herres-Pawlis/Blanke et al., Faraday Discuss. 2025; IUPAC-InChI GitHub; RDKit documentation.