




orthanc: An R Interface to Orthanc DICOM Servers for Medical Imaging Workflows

Matthew T. Warkentin ^{1¶}

¹ Department of Oncology, Cumming School of Medicine, University of Calgary, Calgary, AB, Canada ¶
Corresponding author

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Open Journals](#) 

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright
and release the work under a
Creative Commons Attribution 4.0
International License ([CC BY 4.0](#)).

Summary

Medical imaging plays a central role in modern biomedical research and clinical practice. The Digital Imaging and Communications in Medicine (DICOM) standard underpins the storage, exchange, and management of medical images across modalities such as computed tomography (CT), magnetic resonance imaging (MRI), positron emission tomography (PET), ultrasound, and digital pathology ([DICOM Standards Committee, 2020](#)). As imaging data become increasingly integrated into quantitative research workflows, particularly in artificial intelligence (AI) and machine learning (ML), there is a growing need for robust, reproducible, and programmatic access to imaging archives.

Over the past decade, there has been an explosion in the volume of medical imaging data generated worldwide. Advances in imaging technology, increased clinical utilization, multi-phase acquisition protocols, and the growth of large-scale research cohorts have resulted in rapidly expanding DICOM archives. Modern scanners routinely produce hundreds to thousands of slices per study, and longitudinal imaging further multiplies data volume per patient. This dramatic growth in image data intensifies the need for scalable, programmatic tools to efficiently query, manage, and analyze imaging repositories.

Orthanc is a lightweight, open-source DICOM server that exposes a comprehensive REST API for managing, querying, retrieving, and modifying DICOM resources ([Jodogne, 2018](#)). It is widely adopted in research settings due to its ease of deployment, extensibility, and compliance with DICOM standards. While Orthanc provides a powerful HTTP-based interface, interacting with its REST API directly requires manual construction of requests, careful handling of authentication and responses, and detailed knowledge of Orthanc's resource hierarchy.

The orthanc R package provides a high-level, idiomatic interface to the Orthanc REST API for the R language ([Warkentin, 2026](#)). Inspired by the design and usability of the PyOrthanc Python package ([Couture et al., 2025](#)), orthanc enables R users to interact programmatically with Orthanc servers using familiar R paradigms. The package abstracts HTTP details, models DICOM resources as structured R objects, and supports reproducible workflows for querying, retrieving, filtering, and managing imaging data. By bridging Orthanc and R ecosystems, orthanc facilitates imaging-based research, statistical analysis, and AI development within a unified environment.

Statement of Need

Medical imaging has become foundational to precision medicine, population health research, and the development of AI-driven diagnostic and prognostic tools. Large-scale imaging cohorts, hospital-based archives, and multi-center research studies generate terabytes to petabytes of DICOM data. These datasets are increasingly used to train predictive models, extract radiomic

41 features, validate quantitative biomarkers, and integrate imaging with clinical and genomic
42 data. As a result, the ability to programmatically access and manage DICOM data is now a
43 core requirement for many data scientists.

44 DICOM is the international standard for medical imaging. It defines not only file formats but
45 also metadata structures, hierarchical relationships (Patient -> Study -> Series -> Instance),
46 and communication protocols. While DICOM ensures interoperability, it is inherently complex.
47 Navigating DICOM metadata, managing identifiers, extracting encapsulated or embedded
48 documents (e.g., PDF) and handling binary pixel data can be technically demanding, especially
49 when interacting with remote servers.

50 Orthanc addresses many of these challenges by providing a modern, RESTful interface to
51 DICOM archives. It supports storage, search, anonymization, modification, plugin-based
52 extensions, and integration with external systems. Its REST API exposes resources
53 corresponding to patients, studies, series, and instances, as well as endpoints for querying
54 metadata, downloading files, and performing server-side operations. Orthanc has therefore
55 become a popular choice in academic and research settings, where flexibility and open-source
56 tooling are essential.

57 Despite its strengths, interacting directly with the Orthanc REST API presents several
58 challenges:

- 59 1. Low-level HTTP handling: Users must construct requests, manage authentication headers,
60 handle response codes, and parse responses.
- 61 2. Hierarchical resource navigation: Traversing patients, studies, series, and instances
62 requires repeated calls and careful bookkeeping of identifiers.
- 63 3. Error handling and robustness: Direct API use requires explicit management of network
64 failures, timeouts, and malformed responses.
- 65 4. Reproducibility and abstraction: Embedding raw HTTP calls within analytical scripts
66 can reduce clarity and hinder maintainability.

67 In the Python ecosystem, packages such as PyOrthanc provide higher-level abstractions that
68 simplify these interactions. However, the R ecosystem has lacked a comparable, dedicated
69 interface to Orthanc. The orthanc R package has been designed with the goal of achieving
70 feature parity with PyOrthanc, providing comparable resource abstractions, querying capabilities,
71 and server interactions so that R users can access a similarly complete and expressive interface
72 to the Orthanc REST API.

73 Without a dedicated interface, R users must either rely on *ad hoc* HTTP code, external scripts
74 in other languages, or manual exports from Orthanc, introducing inefficiencies and potential
75 reproducibility issues. The orthanc package addresses this gap by providing an R-native,
76 object-oriented interface to the Orthanc REST API. It enables researchers to remain within
77 the R environment for the entire imaging workflow: from data discovery and retrieval to
78 downstream statistical analysis and AI model development. By reducing boilerplate HTTP
79 code and encapsulating Orthanc's resource model in intuitive R classes and functions, orthanc
80 lowers the barrier to entry and promotes reproducible imaging research.

81 Features and Functionality

82 The orthanc package provides a comprehensive and user-friendly interface to the Orthanc
83 REST API, designed to align with idiomatic R workflows while preserving the structure and
84 semantics of DICOM resources.

85 The orthanc package leverages the `httr2` R package for HTTP communication (Wickham,
86 2025), providing a modern, robust, and well-structured framework for constructing requests,
87 handling authentication, managing errors, and processing responses, thereby ensuring reliable

88 and maintainable interactions with the Orthanc REST API. The package also utilizes `mirai` to
89 provide an asynchronous client interface, enabling non-blocking API requests and concurrent
90 operations that improve performance and scalability when interacting with large imaging
91 archives (Gao, 2025a).

92 Client Abstraction

93 At the core of the package is an R6-based client object that represents a connection to an
94 Orthanc server. Implemented using R6 object-oriented programming, the client encapsulates:

- 95 ■ Server URL configuration
- 96 ■ Authentication credentials
- 97 ■ HTTP request handling
- 98 ■ Response parsing and binary data management

99 Using R6 allows the client to maintain state (e.g., base URL, authentication headers,
100 configuration options) across method calls, providing a natural and efficient interface for
101 repeated interactions with an Orthanc server (Chang, 2021). Methods on the client object
102 wrap Orthanc REST endpoints, abstracting low-level HTTP details and returning structured R
103 objects rather than raw responses. This design promotes clarity, reuse, and testability while
104 shielding users from the complexities of direct API interaction.

105 Object-Oriented Representation of DICOM Resources

106 The package models Orthanc resources (patients, studies, series, and instances) as R6 classes
107 that respect the DICOM hierarchy. Each resource is represented as an R6 object that:

- 108 ■ Encapsulates the corresponding Orthanc identifier
- 109 ■ Maintains a reference to the originating client
- 110 ■ Provides methods and fields to retrieve metadata
- 111 ■ Supports navigation across hierarchical relationships
- 112 ■ Enables downloading of DICOM files or pixel data

113 By leveraging R6, these objects combine data and behavior, allowing users to interact with
114 DICOM resources through intuitive method calls (e.g., retrieving child resources or accessing
115 metadata) while preserving internal state. The use of R6 supports encapsulation, inheritance
116 where appropriate, and clean separation of public and private methods, making the API
117 expressive yet structured.

118 Reflecting the DICOM hierarchy directly in R6 classes enables intuitive traversal of imaging
119 archives (e.g., from patient to study to series to instance) and promotes readable, object-oriented
120 workflows within the R environment.

121 Querying and Filtering

122 `orthanc` provides high-level functions to query and filter DICOM resources based on metadata.
123 Users can:

- 124 ■ Search for patients or studies using DICOM tags
- 125 ■ Apply predicate functions (filters) in R to refine results
- 126 ■ Iterate across resources using vectorized or functional programming tools

127 This design integrates naturally with the programming paradigms common in modern R
128 workflows.

129 Data Retrieval and Management

130 The package supports:

- 131 ▪ Downloading DICOM instances as raw binary data
- 132 ▪ Exporting resources to disk
- 133 ▪ Accessing and modifying metadata
- 134 ▪ Performing anonymization and other server-side operations (when supported by the
- 135 Orthanc configuration)

136 Binary handling is abstracted so users can focus on analysis rather than low-level file
137 management.

138 Asynchronous Parallelization and Scalability

139 To support high-throughput imaging workflows, orthanc integrates asynchronous execution via
140 the mirai R package (Gao, 2025a). This enables non-blocking API requests and concurrent
141 operations across multiple Orthanc resources, which is particularly valuable when querying large
142 cohorts, uploading or retrieving many DICOM resources, or performing repeated metadata
143 lookups. By allowing requests to be dispatched and resolved in parallel using background
144 R processes, the package improves responsiveness and overall throughput in data-intensive
145 research settings.

146 The mirai framework is built on top of nanonext (Gao, 2025b), an R interface to the Nanomsg
147 Next Generation (NNG) messaging library. This architecture provides a lightweight, high-
148 performance, message-passing infrastructure that supports scalable parallel computation with
149 minimal overhead. By leveraging mirai and its NNG-based backend, orthanc offers a robust
150 and efficient asynchronous client model that scales from local multicore environments to
151 distributed systems, while maintaining a simple and idiomatic R interface.

152 Integration with the R Ecosystem

153 By operating natively in R, orthanc enables seamless integration with:

- 154 ▪ Statistical modeling packages
- 155 ▪ Machine learning frameworks
- 156 ▪ Visualization tools
- 157 ▪ Reproducible reporting systems (e.g., R Markdown, Quarto)

158 This integration is particularly valuable for imaging-based AI workflows, where DICOM metadata
159 and derived features must be linked to clinical outcomes and analyzed statistically.

160 Conclusion

161 In summary, orthanc fills a critical gap in the R ecosystem by providing a robust, high-level
162 interface to the Orthanc DICOM server. As medical imaging continues to drive innovation in
163 AI and quantitative research, accessible and reproducible tools for managing DICOM data are
164 essential. By simplifying interaction with Orthanc and embedding imaging workflows directly
165 within R, orthanc supports the growing community of researchers working at the intersection
166 of imaging, statistics, and data science.

167 Acknowledgements

168 I gratefully acknowledge the authors and maintainers of Orthanc for developing and maintaining
169 an open-source, standards-compliant DICOM server that has become an essential tool in medical
170 imaging research. Orthanc's thoughtful design, comprehensive REST API, and commitment to
171 openness have made it possible for researchers to build reproducible and extensible imaging
172 workflows across institutions and disciplines. The continued development of open-source
173 infrastructure is foundational to collaborative, transparent, and reproducible research in medical
174 imaging.

175 I also thank the authors and contributors of PyOrthanc, whose work provided important
176 conceptual and design inspiration for this package. Their efforts in creating a clear and
177 accessible programmatic interface to Orthanc in the Python ecosystem helped inform the
178 structure and usability goals of the orthanc R package.

179 References

- 180 Chang, W. (2021). *R6: Encapsulated Classes with Reference Semantics*. <https://doi.org/10.32614/CRAN.package.R6>
181
- 182 Couture, G., Lemaréchal, Y., & Després, P. (2025). PyOrthanc: A python interface for orthanc
183 DICOM servers. *Journal of Open Source Software*, 10(106), 7562.
- 184 DICOM Standards Committee. (2020). *Digital imaging and communications in medicine*
185 *(DICOM) standard*. NEMA.
- 186 Gao, C. (2025a). *Mirai: Minimalist async evaluation framework for r*. <https://doi.org/10.5281/zenodo.7912722>
187
- 188 Gao, C. (2025b). *Nanonext: NNG (nanomsg next gen) lightweight messaging library*. <https://doi.org/10.5281/zenodo.7903429>
189
- 190 Jodogne, S. (2018). The orthanc ecosystem for medical imaging. *Journal of Digital Imaging*,
191 31(3), 341–352. <https://doi.org/10.1007/s10278-018-0082-y>
- 192 Warkentin, M. T. (2026). *Orthanc: Programmatic interface to 'orthanc' REST API*. <https://github.com/mattwarkentin/orthanc>
193
- 194 Wickham, H. (2025). *httr2: Perform HTTP requests and process the responses*. <https://doi.org/10.32614/CRAN.package.httr2>
195