



## RECOMMENDATIONS FOR SUPPORTING ORCID IN REPOSITORY PLATFORMS

 [Wesley Barry](#)  [Josh Brown](#)  [Tom Demeranville](#)  [Paola Galimberti](#)  
 [Stephen Grace](#)  [Daryl M. Grenz](#)  [Laure Haak](#)  [Masaharu Hayashi](#)  
 [Salwa Ismail](#)  [Liz Krznarich](#)  [Bénédicte Kuntziger](#)  [Agnès Magron](#)  
 [Alice Meadows](#)  [Michele Mennielli](#)  [Lars Holm Nielsen](#)  [Sheila Rabun](#)  
 [Andrea Szwajcer](#)  [Washington R. de Carvalho](#)

## Contents

[1. Background](#)

[2. Audience](#)

[3. Purpose](#)

[4. Scope](#)

[5. Definitions](#)

[6. Recommendation summary](#)

[7. Recommended system requirements for repository platforms](#)

[7.1 Collecting ORCID iDs](#)

[7.1.1 Via direct interaction with a user](#)

[Considerations surrounding revoking OAuth2 access tokens:](#)

[7.1.2 Via mediated deposit by administrator](#)

[7.1.3 Via external item deposits/imports from authorized third parties](#)

[7.2 Displaying ORCID iDs](#)

[7.3 Pulling information from ORCID](#)

[7.4 Pushing information to ORCID](#)

[7.5 Administrative features](#)

[7.6 Machine-readable exposure of ORCID iDs](#)

[7.7 Documentation & communication](#)

[Appendix 1: ORCID API documentation & support](#)

[Appendix 2: Relevant third-party standards](#)

[Appendix 3: Additional community suggestions](#)

## 1. Background

ORCID can be used in repositories to clearly link authors—and all their name variants—with their publications, affiliations, funding awards and other research activities, improving search and retrieval and supporting networking and collaboration. Using the ORCID API, repositories can exchange data with the ORCID Registry to populate local author profiles. They can also update ORCID records with publication information each time a repository deposit is made.

ORCID adds visibility to repository content and authors, facilitates collaborations and networking, and helps organizations with institutional reporting, national assessment programs, and management of workflows throughout the research cycle.

To guide developers and repository managers in writing specifications and building effective integrations that maximize the benefits ORCID can bring to repositories, the following recommendations were developed with input from a [task force](#) comprising repository experts from around the world. Many other members of the repository community also contributed feedback during a public comment period.

## 2. Audience

These recommendations are intended primarily for those involved in designing/developing repository software platforms for distribution and use by other organizations, either installed locally or offered as a hosted solution.

These recommendations may also be useful to those extending an existing software repository platform or developing a completely customized application for use by a single organization, however, some aspects such as [Administrative features](#) and [Documentation & communication](#) are not applicable for software that will not be distributed for use by other organizations.

## 3. Purpose

These recommendations are intended to ensure a consistent base level of support for ORCID in repository platforms, so that entities implementing those platforms are able to use ORCID effectively ‘out of the box’, without need for extensive customization.

## 4. Scope

These recommendations are intended to support a basic level of integration with ORCID; as such, they are not exhaustive and may not fit every use case. Additional customization may be needed to achieve the goals of a particular repository system/service.

Additionally, these recommendations were developed considering current best practices, current ORCID functionality/features, and current support of ORCID in other systems that may impact or interact with repositories. Some aspects of these current circumstances are not ideally aligned with repository workflows; however, these recommendations serve as a starting point which will evolve as both ORCID and the scholarly communication ecosystem evolve.

## 5. Definitions

- **Authenticated ORCID iD:** An ORCID iD that was obtained after a user signed into their ORCID account and authorized a trusted organization to obtain their ORCID iD. Processes for obtaining authenticated iDs within an integration are described in the [Basic tutorial: Get an authenticated ORCID iD](#) and [ORCID Auth with OpenID Connect](#). Authenticated iDs can be obtained using the [ORCID Public API, Basic Member API or Premium Member API](#).
- **Permission:** The ability to update an ORCID record, which is represented by an [OAuth2 access token](#) obtained through an OAuth interaction with the record holder. Processes for obtaining permission within an integration are described in [Basic tutorial: Add and update data on an ORCID record](#) and [ORCID Auth with OpenID Connect](#). Permission can be obtained using the [ORCID Basic Member API or Premium Member API](#).
- **Non-authenticated/unauthenticated ORCID iD:** An ORCID iD that was not obtained using OAuth, as described above.
- **Repository:** System or service that ingests, preserves, and disseminates metadata and associated digital assets. Even though the use case of open access to an anonymous audience of human and machine users is at the heart of many repository services, each repository owner makes its own decisions which content is ingested, and how that content is managed and shared.
- **Repository platform:** Software intended for distribution and use by entities external to the developer of the platform, either installed locally or offered as a hosted solution, in order to enable the operation of repository services/systems by those entities
- **Repository platform instance:** An implementation of a given repository platform by a particular entity
- **Administrator:** An individual known to a repository platform instance who manages some or all aspects of that instance
- **User:** An individual known to a repository platform instance who contributes content to or consumes content from that instance

## 6. Recommendation summary

While individual repository system/service needs and workflows may vary, the [ORCID in Repositories Task Force](#), with input from repository community members, has identified the following as key ORCID features that repository platforms should strive to include:

- **Support collecting authenticated ORCID iDs**, which means that users sign into their ORCID accounts and authorize the repository to obtain their ORCID iD and (optionally) permission to update their ORCID records
- **Support other ways of obtaining ORCID iDs**, including in mediated deposits and bulk uploads by repository managers, as well as automated deposits from other systems

- **Allow administrators to request authenticated ORCID iDs** and ORCID record update permission from authors and co-authors, in cases where iDs are missing or have not been authenticated
- **Support displaying ORCID iDs** wherever user/contributor information is displayed, according to the [Guidelines on the display of ORCID iDs in publications](#). ORCID iDs that have not been authenticated by their owner should be displayed slightly differently from those that have been authenticated
- **Support pulling and pushing information to and from ORCID**. Types of information, as well as frequency and other options should be configurable by repository managers
- **Provide testing, logging and reporting features** to help administrators troubleshoot issues and manage ORCID-related data in their repository
- **Support exposing ORCID iDs in metadata outputs**, such as OAI-PMH XML, wherever possible
- **Provide documentation** about ORCID features, for both administrators and end users

## 7. Recommended system requirements for repository platforms

### 7.1 Collecting ORCID iDs

There are three generalised processes through which iDs are collected into a repository platform instance:

- [7.1.1 Via direct interaction with a user](#): An authenticated iD (and, optionally, permission to update the user's ORCID record) is obtained from a user through an OAuth interaction, facilitated by the repository platform
- [7.1.2 Via mediated deposit](#): An unauthenticated iD is associated with an item or user by an individual other than the iD owner. Since an OAuth interaction did not take place, the repository platform instance does not have permission to update the user's ORCID record
- [7.1.3 Via external 3rd party import/deposit](#): An iD is extracted from metadata obtained via import or deposit from an external source. This iD may or may not have been authenticated by the metadata source. Since an OAuth interaction did not take place, the repository platform instance does not have permission to update the user's ORCID record.

### 7.1.1 Via direct interaction with a user

1. Support collecting authenticated ORCID iDs from users via the ORCID Member and Public APIs as described in [Basic tutorial: Get an authenticated ORCID iD](#).
2. Support collecting OpenId connect id tokens as described in [ORCID Auth with OpenId Connect](#).
3. For institutions with access to the ORCID Member API, also allow users to grant a repository platform instance permission to add/update information on their ORCID record as described in [Basic tutorial: Add and update data on an ORCID record](#).
4. Allow users to grant these permissions while initially authenticating their ORCID iD and at a later time (if permission was not granted initially).
5. Support storing authenticated ORCID iDs, associated OAuth2 access tokens, OpenId Connect ID tokens, and related data returned by the ORCID API, including:
  - **access\_token**: token returned by the ORCID API following a successful OAuth2 process
  - **refresh\_token**: token returned by the ORCID API following a successful OAuth2 process
  - **expires\_in**: expiration timestamp for the access token above
  - **scope**: permissions granted via the ORCID API during the OAuth2 authorization process that resulted in the above access token
  - **ORCID iD**: 16-digit ORCID identifier associated with the user who granted above permissions
  - **id\_token (for OpenID Connect implementations)**: Publicly shareable JSON web token that can prove a user authenticated using ORCID at a specific time
6. Allow users to change the ORCID iD connected to their account or profile within the repository platform instance at any time, in case an incorrect iD was previously connected.
7. Allow users to remove the connection to their ORCID iD at any time, revoking any OAuth2 access tokens stored within repository, as described in [Revoke tokens](#). Revoking a token removes any permissions granted to the system, but the iD itself can still be considered authenticated.
8. While removing or changing the connection to their ORCID iD, allow users the option to remove any data from the repository that was retrieved from their ORCID record, if this feature is enabled by an administrator. Allow administrators to control whether this option is available to users, as needed to comply with local data protection regulations/policies. If this feature is not enabled, provide messaging to the user indicating that data

previously retrieved from ORCID will remain in the repository after the connection is removed/changed.

Considerations surrounding revoking OAuth2 access tokens:

- ORCID record holders may [revoke permission](#) in the [account settings](#) section of their ORCID record at any time. This disables all OAuth2 access tokens issued to an ORCID API client that have identical permission scope(s), without notification to the API client. This result is different from the result of an API client revoking tokens as described in [Revoke tokens](#), which revokes only the token(s) specified in the API requests
- When registering for an ORCID account, each ORCID record holder agrees to [Terms of Use](#) that do not require that data retrieved from their ORCID record be removed from external systems when permission is revoked in the account settings section of their ORCID record
- In cases where an entity operates multiple systems integrated with ORCID, token revocation outcomes vary as follows:

Case	Permission revoked by API client	Permission revoked by user in ORCID account settings
An ORCID API client ID <b>AND</b> OAuth2 tokens are shared between 1 or more systems	Only token(s) specified in API request(s) are disabled for all systems	All tokens with identical permission scope(s) are disabled for all systems
An ORCID API client ID is shared between 1 or more systems, but OAuth2 tokens are <b>NOT</b> shared between systems	Only token(s) specified in API request(s) are disabled for the system that made the request(s)	All tokens with identical permission scope(s) are disabled for all systems
Different ORCID API client IDs are used in each system	Only token(s) specified in API request(s) are disabled for the system that made the request(s)	Tokens with identical permission scope(s) are disabled only for the system using the ORCID API client ID those tokens were issued to

### 7.1.2 Via mediated deposit by administrator

1. Allow administrators to associate ORCID iDs with authors/contributors within the repository platform instance, as well as any associated OAuth2 access tokens, OpenID Connect ID tokens, token scopes and token expiration dates collected by other local systems.

2. iDs without an accompanying OAuth2 access token and/or OpenID Connect ID token should be considered non-authenticated:
  - Each non-authenticated iD should be resolved to an ORCID record to ensure that it is valid (ie: [https://orcid.org/\[non-authenticated iD\]](https://orcid.org/[non-authenticated iD]) returns an http '200' response); if it does not resolve, the iD should be rejected
  - Non-authenticated iDs should be differentiated in display, as described in [Displaying ORCID iDs](#)
  - Pushing information to ORCID, as described below, is not possible for non-authenticated iDs
3. Provide functionality that allows administrators to prompt authors/contributors to authenticate a non-authenticated iD that has been associated with their account/profile and (optionally) grant permission to the repository platform instance to update their ORCID record(ex: via an email sent when a non-authenticated iD is associated with an author by a administrator). Aspects of this functionality, such as whether such notifications are sent automatically or when initiated by an administrator, should be configurable by the administrator. When a non-authenticated iD becomes authenticated, its display should be updated as described in [Displaying ORCID iDs](#).

### **7.1.3 Via external item deposits/imports from authorized third parties**

Repositories often receive metadata and/or associated digital objects from authorized third parties via machine interfaces. Examples of such “authorized third parties” include:

- CRIS (Symplectic Elements, PURE, Converis, etc)
- Notification brokers like [PubRouter](#)
- Publishers that deposit directly to repositories, like in the case of [BioMed Central SWORD deposits](#)
- Other third party APIs that a repository chooses to import data from





Processing of data after a deposit/import event from an authorized third party depends on the policy of the repository service and its associated quality control/workflow systems, therefore, handling of ORCID iDs in such cases should be highly configurable by the repository administrator.

1. Allow ingestion of ORCID IDs included in SWORDv2 or SWORDv3 deposited items.
2. Allow ingestion of ORCID IDs included in repository platform-specific API deposits.
3. Allow administrators to configure policy for characterizing imported iDs as authenticated, based on factors including:
  - A trust relationship between the repository and the data source. If it is known that the third party system authenticates iDs as a matter of course, then iDs from that

system can be imported as such (ex: a CRIS operated by the same organization as the repository, which is known to include authenticated iDs in deposits)

- Presence of an `authenticated="true"` attribute associated with an iD, which is provided by some third party metadata sources, such as [Crossref](#)
  - Presence of an [OAuth2 access token](#) and/or an [OpenID Connect ID token](#) which can be used to prove that authentication occurred
4. Provide functionality that allows administrators to prompt authors/contributors to authenticate imported iDs that have been associated with their account/profile and (optionally) grant permission to the repository platform instance to update their ORCID record (ex: via an email sent when a non-authenticated iD is associated with an author by a administrator). Aspects of this functionality, such as whether such notifications are sent automatically or when initiated by an administrator, should be configurable by the administrator. When a non-authenticated iD becomes authenticated, its display should be updated as described in [Displaying ORCID iDs](#).

## 7.2 Displaying ORCID iDs

1. ORCID iDs should be displayed where user/contributor information is displayed, in locations such as:
  - User account/profile pages
  - Item summary/abstract pages
  - Browse by author/creator/contributor pages
2. Authenticated iDs should be displayed according to the [Guidelines on the display of ORCID iDs in publications](#), which includes the following formats:
  - iD icon followed by the full iD URI, hyperlinked to the iD URI:  
 <https://orcid.org/0000-0002-1825-0097>
  - iD icon, hyperlinked to the https iD URI: 
  - https iD URI, hyperlinked to the iD URI: <https://orcid.org/0000-0002-1825-0097>
3. Unauthenticated iDs should be displayed using one of the options above, followed by text such as “(unconfirmed)”, “(unverified)” or “(unauthenticated)”, or the equivalent in the local language:
  - iD icon followed by the full iD URI, hyperlinked to the iD URI:  
 <https://orcid.org/0000-0002-1825-0097> (unconfirmed)
  - iD icon, hyperlinked to the https iD URI:  (unconfirmed)

- [https id URI](https://orcid.org/0000-0002-1825-0097), hyperlinked to the id URI:  
<https://orcid.org/0000-0002-1825-0097> (unconfirmed)

## 7.3 Pulling information from ORCID

*Note: Pulling information from ORCID can be performed using any ORCID API ([Public API](#), [Basic Member API](#) or [Premium Member API](#)), however, data returned will vary depending on the user's [visibility settings](#), the API type, and permission granted to a given API client.*

1. Allow the user and/or a administrator to pull data from the user's ORCID record into the repository as described in [Basic tutorial: Read data on an ORCID record](#), including (where applicable):
  - [Personal information](#) (country, names, person identifiers, website URLs, etc.)
  - [Research activities](#) (particularly [Works](#), such as publications and datasets, but also others including [Funding](#), and [Education](#) or [Employment](#) affiliations, if supported in your repository platform)
2. Allow administrators to configure settings for pulling information, including the ability to:
  - Enable/disable pulling information from ORCID for the entire repository platform instance
  - Choose to pull information for all ORCID iDs stored in the repository platform instance or only authenticated iDs
  - Choose to pull information on demand (one-time) or automatically, on a regular basis. If automatic, regular pulling is configured, ORCID data in the repository platform instance should be synchronized with the ORCID Registry, to incorporate additions, removals or changes made in the ORCID Registry since the last pull. This can be done using [put-codes](#) or [created/last modified datesw](#), which are included in the metadata for each item on an ORCID record
  - Choose which types of information to pull (personal information, education/employment affiliations, works, etc.). For works, it is also desirable to allow administrators to choose specific [work types](#) and sources
3. If the repository platform allows users to control pulling their own data from ORCID, also allow users to configure settings for pulling information, including:
  - Enable/disable pulling information from ORCID
  - Choose to pull information on demand (one-time) or automatically, on a regular basis. If automatic, regular pulling is configured, ORCID data in the repository platform instance should be synchronized with the ORCID Registry, to incorporate additions, removals or changes made in the ORCID Registry since the last pull. This can be done using [put-codes](#) or [created/last modified dates](#), which are included in the metadata for each item on an ORCID record.

4. Allow administrators to notify users when information is pulled from their ORCID records.

## 7.4 Pushing information to ORCID

*Note: Pushing information to ORCID requires permission to update a user's ORCID record, which is available only to ORCID member organizations using the [Basic Member API](#) or [Premium Member API](#).*

1. If the repository platform instance has an authenticated ORCID iD for a given user and permission to update their ORCID record, allow the user and/or administrator to push data from the repository to the user's ORCID record as described in [Basic tutorial: Add and update data on an ORCID record](#), including (where applicable):
  - [Personal information](#) (country, names, person identifiers, website URLs, etc.)
  - [Research activities](#) (particularly [Works](#), such as publications and datasets, but also others including [Funding](#), [Education](#) or [Employment](#) affiliations, if available in your repository platform)
2. Allow administrators to configure settings for pushing information, including the ability to:
  - Enable/disable pulling information from ORCID for the entire repository platform instance
  - Choose to push information on demand (one-time) or automatically, on a regular basis.
  - Choose which types of information to push (personal information, education/employment affiliations, works, etc.).
3. If the repository platform allows users to control pushing their own data to ORCID, also allow users to configure settings for pushing information, including the ability to:
  - Enable/disable pushing information from ORCID
  - Choose to push information on demand (one-time) or automatically, on a regular basis.

## 7.5 Administrative features

1. Require administrators to obtain a set of ORCID Public or Member API credentials and supply them to their repository platform instance (for example: by entering them into a configuration form or file).

*Consider providing an option to test that credentials were entered correctly by requesting a [two-legged OAuth /read-public access token](#).*

2. Provide administrators with information about how to obtain API credentials, including:
  - Direction on [which API is required to use your repository platform](#)
  - Direction on how to obtain [member API credentials](#) and [public API credentials](#)

*Some fields in the ORCID member API credential request forms can be pre-filled by providing query parameters in links to the forms. Please contact [support@orcid.org](mailto:support@orcid.org) for more information.*

3. Provide an option for testing against the [ORCID sandbox](#), in which administrators can enter [sandbox API credentials](#) and test features such as collecting authenticated IDs, pulling and pushing information using ORCID Sandbox records.
4. Allow administrators to export a report from the repository of stored ORCID IDs, access tokens and/or ID tokens, and related data, including refresh tokens, scopes, and token expiry.
5. Follow [best practices for logging](#) interactions with the ORCID API, where possible.

## 7.6 Machine-readable exposure of ORCID IDs

Aside from the section on “Displaying ORCID IDs”, where ORCID IDs are exposed to human users accessing the repository through a browser, following requirements are key to ensure that repositories can continue to serve as a trustworthy source of high quality metadata, where as many elements as possible are uniquely identified.

1. Include ORCID IDs in machine-readable metadata outputs whenever possible, such as:
  - OAI-PMH XML
  - Repository platform-specific APIs/exports
2. Express ORCID IDs according to standards/best-practices whenever possible, such as those listed in [Appendix 2: Relevant third-party standards](#).

## 7.7 Documentation & communication

1. Provide documentation for administrators describing how to:
  - Configure ORCID features within a repository platform instance
  - Use the repository platform’s ORCID-related administrative features
2. Provide documentation for end users describing how to use the repository platform’s ORCID features (link to articles in the [ORCID Knowledge Base](#), where appropriate).

3. Provide ORCID with administrator and end user documentation described above, so that ORCID staff are prepared to answer questions sent to its help desk, and so your repository platform can be listed as an [ORCID-enabled system](#).
4. [Notify ORCID](#) when your repository platform's ORCID features are initially released and when there are changes to those features so we can update our documentation.
5. Agree to initial and annual reviews of your repository platform's ORCID features by ORCID staff.

## Appendix 1: ORCID API documentation & support

- [ORCID API resources site](#) ([Getting started guide](#), [Tutorials](#), [Graphics](#), [code examples](#) & [other helpful resources](#))
- [ORCID API FAQs](#)
- [ORCID OAuth authentication reference](#)
- [ORCID Schema and API reference](#)
- [ORCID API users Google group](#)
- [Contact ORCID](#)

## Appendix 2: Relevant third-party standards

- [RIOXX 2.0](#) describes how to represent ORCID IDs in OAI-PMH feeds for repository-repository interoperability
- [OpenAIRE v4 Creator guidelines based on Datacite v4.1 Creator](#)
- [DCMI Best Practice proposal](#)

## Appendix 3: Additional community suggestions

As noted in the [Scope](#) section above, these recommendations are intended to support a basic level of integration with ORCID; there are certainly many more ORCID-related features that might be useful to include in a repository platform.

Some relevant suggestions from the community include:

- Support ingesting ORCID iDs from the eduPersonOrcid attribute stored within an institution's federated identity and/or directory data
- Provide features for preventing duplicates when importing data from ORCID
- Indicate ORCID as the source of data retrieved via the ORCID API (additionally, since each piece of information on an ORCID includes a source, the source that added the data to ORCID can also be indicated)