# Application Links Documentation

## About Application Links

**Application Links (AppLinks)** is a bundled plugin that allows you to link your JIRA, Confluence, FishEye, Crucible and Bamboo applications. Linking two applications allows you to share information and access one application's functions from within the other. For example, if you link JIRA and Confluence, you can view JIRA issues on a Confluence page via the JIRA Issues macro. You can even link your individual projects, spaces and repositories with each other, across the different applications.

Note that the Application Links plugin is bundled and shipped with the Atlassian applications. You cannot install it yourself. Applications Links is bundled with FishEye 2.4, Confluence 3.5, JIRA 4.3, and all later versions of those applications. In addition, Bamboo 3.1 is compatible with AppLinks. You can configure JIRA-to-Bamboo links via the JIRA administration screens.

## Getting Started

The Quick Start Guide provides instructions on how to set up the most common application link configuration.

## Administrator's Guide

The administrator's guide is for administrators who want to configure application links for the applications. The guide contains information on adding a new application link, configuring the authentication for an application link, setting up project links and more.

## Developer Resources

These resources are for developers who want to develop with the Application Links plugin. Take a look at the Development Hub.

# Application Links Quick Start Guide

This page describes an example of how to set up a common application links scenario — **creating a two-way link between two applications** that trust each other and share the same set of users. For example, you may wish to link your internal JIRA server to a private FishEye server. Setting up an application link allows you to take advantage of integration features like viewing FishEye changesets in JIRA. The instructions below also detail how to **link two entities** of your two linked applications, e.g. a JIRA project to a FishEye repository.

Applications Links is bundled with FishEye 2.4, Confluence 3.5, JIRA 4.3, and all later versions of those applications. In addition, Bamboo 3.1 is compatible with AppLinks. You can configure JIRA-to-Bamboo links via the JIRA administration screens. If one of the applications you are connecting to does not have Applinks, you can still set up an application link to it. See Adding an Application Link.

**On this page:**

## Before You Begin

- The instructions below describe how to link a FishEye server to a JIRA server. However, the instructions will be similar for other Atlassian applications.
- You must be an administrator with permissions to configure changes at the application level, for both of your applications. That is, you must be a 'JIRA System Administrator' in JIRA, not a 'JIRA Administrator'.
- Your local application (FishEye) must have Application Links installed. You can link from an application with Application Links to an application that does not have Application Links (as described on this page), however you will have to configure authentication manually (see Configuring Authentication for an Application Link).
- Make sure that the base URL is set correctly in both the local application and the remote application, (JIRA instructions | Confluence instructions | FishEye/Crucible instructions | Bamboo instructions).

## Adding an Application Link

In this example, we will create a **two-way Trusted Applications link** between a 'local' FishEye server with Application Links ((http://fisheye.example.com/ in this example) and a 'remote' JIRA server without Application Links (http://jira.example.com/ in this example).

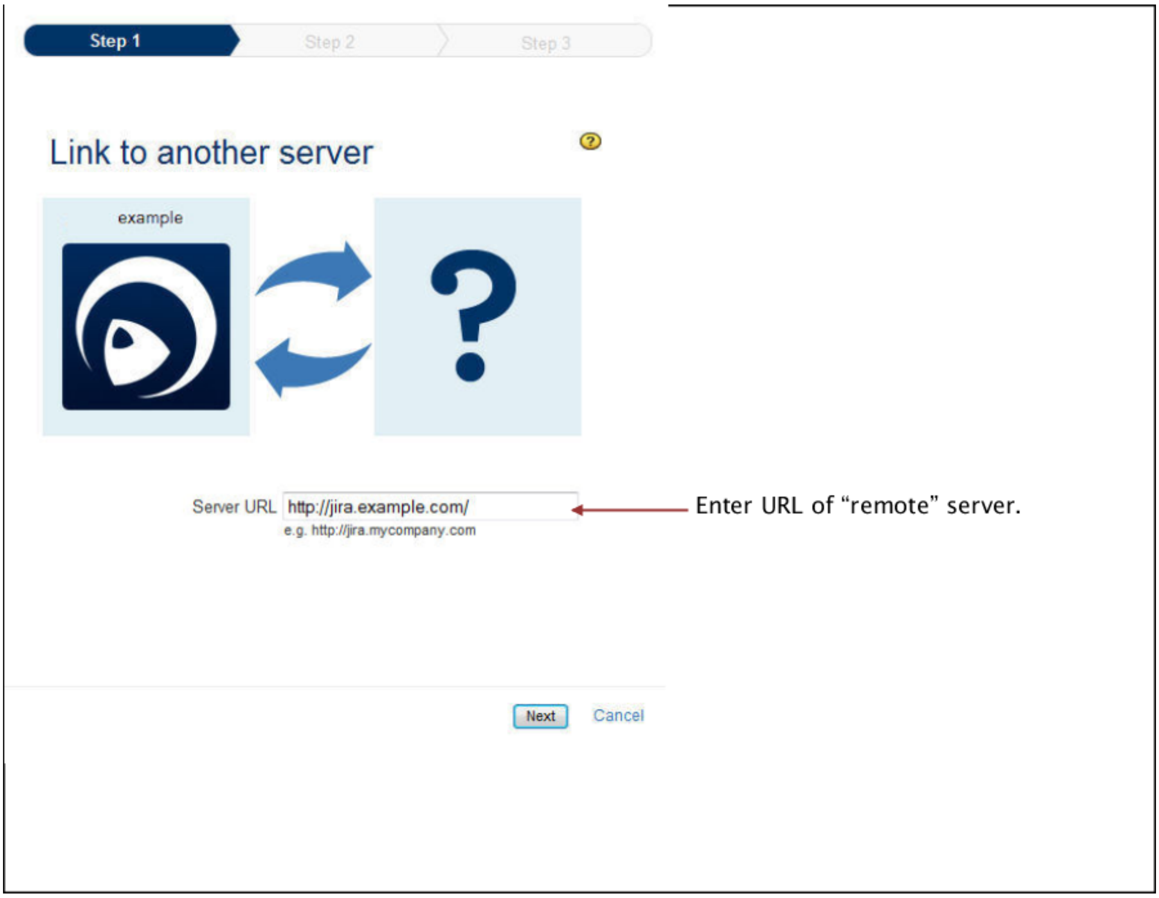
We'll add the link from the FishEye server.

**In FishEye, do the following:** Log in as a system administrator and go to the administration page. Click 'Application Links' in the administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up. Click '**Add Application Link**' to open the add application link wizard.

## 1. Specifying the Remote Application

First, we need to specify which application we are linking to. Enter the URL for the JIRA server in the 'Link to another server' dialogue, as shown in the screenshot below, and click '**Next**'.

*Screenshot: Entering the server URL*



Link to another server

Server URL http://jira.example.com/
e.g. http://jira.mycompany.com

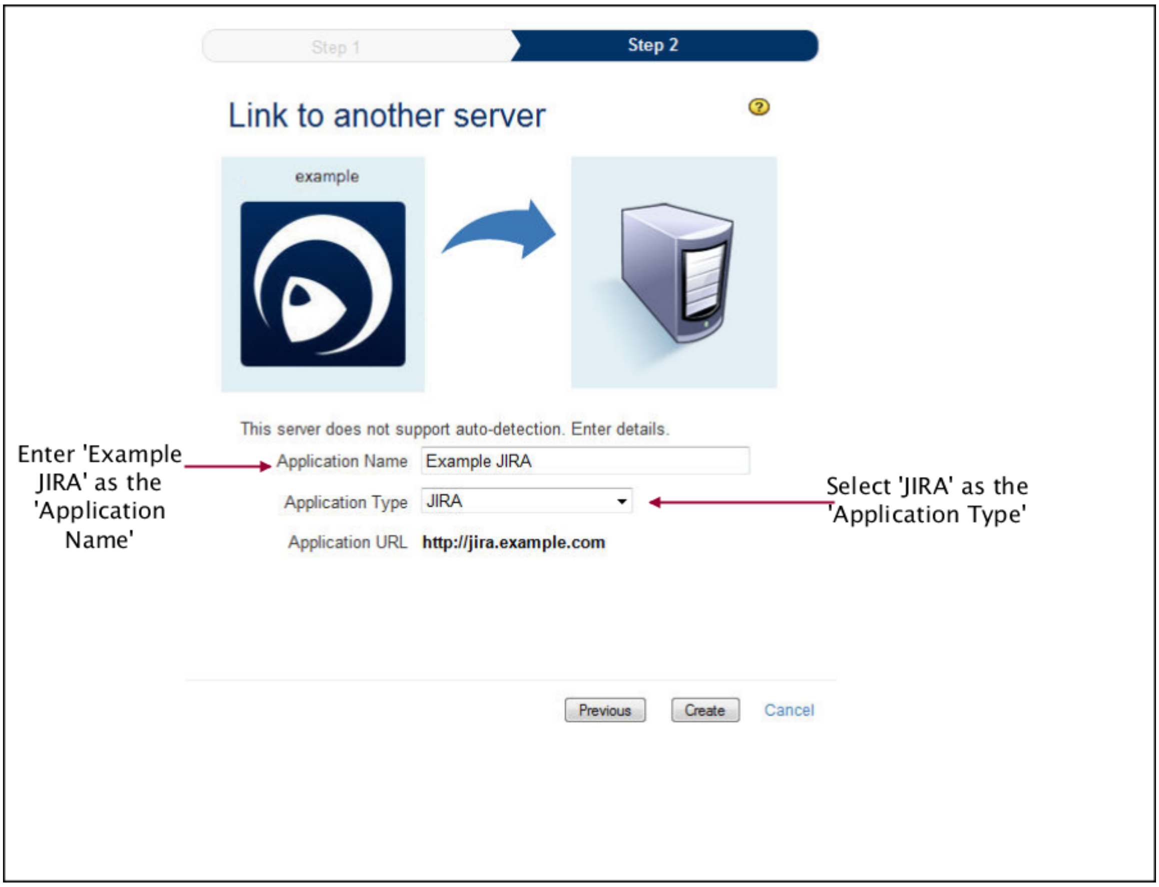
Enter URL of “remote” server.

## 2. Creating the Application Link

Enter an '**Application Name**' and choose the '**Application Type**' to be 'JIRA'. Click the '**Create**' button. The application link will be created and displayed on the 'Configure Application Links' page.

Our JIRA server does not have Application Links, so we cannot automatically create a link back to our FishEye server nor set up authentication. We'll manually set these up in the next step.

*Screenshot: Creating the application link*



Link to another server

This server does not support auto-detection. Enter details.

Application Name: Example JIRA

Application Type: JIRA

Application URL: **http://jira.example.com**

Enter 'Example JIRA' as the 'Application Name'

Select 'JIRA' as the 'Application Type'

### 3. Configuring Authentication for the Application Link

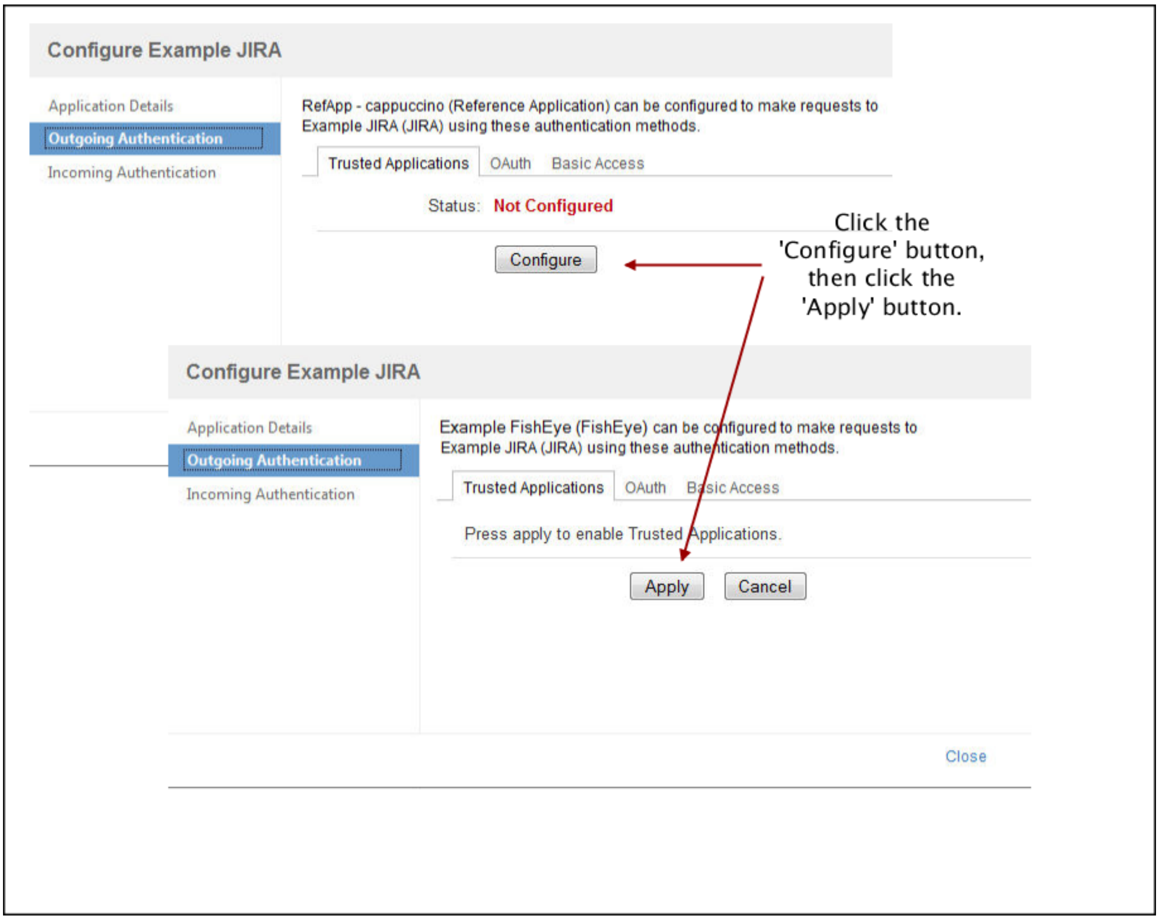
We are going to use Trusted Applications authentication for all incoming and outgoing requests via the application link, as both servers share the same userbase. Trusted Applications authentication is recommended when both applications fully trust each other and share the same set of users (read more about configuring Trusted Applications authentication). Other authentication configurations are described in Configuring Authentication for an Application Link.

On the 'Configure Application Links' page, click the '**Configure**' link next to the application link just created. The configuration dialogue for the application link will be displayed.

First, we need to configure **our JIRA server to trust our FishEye server**. Navigate to the '**Outgoing Authentication**' tab on the configuration dialogue and click the '**Configure**' button on the '**Trusted Applications**' sub-tab. Click the '**Apply**' button to apply Trusted Applications authentication.

We also need to configure **our FishEye server to trust our JIRA server**. Navigate to the '**Incoming Authentication**' tab on the configuration dialogue and click the '**Configure**' button on the '**Trusted Applications**' sub-tab. Click the '**Apply**' button to apply Trusted Applications authentication.

*Screenshot: Setting up Trusted Applications authentication (click to view larger image)*



Configure Example JIRA

RefApp - cappuccino (Reference Application) can be configured to make requests to Example JIRA (JIRA) using these authentication methods.

Status: Not Configured

Click the 'Configure' button, then click the 'Apply' button.

Configure Example JIRA

Example FishEye (FishEye) can be configured to make requests to Example JIRA (JIRA) using these authentication methods.

Press apply to enable Trusted Applications.

### 4. Additional JIRA Configuration

Our JIRA server does not have Application Links, so we need to perform additional Trusted Applications configuration in JIRA before our application link will work.

Follow the instructions on adding a trusted app in the JIRA documentation: Configuring Trusted Applications. You will need to enter the following information:

- '**Base URL**' — Enter the URL of your FishEye server, 'http://fisheye.example.com/'
- '**Application Name**' — Enter the name for your FishEye server, 'Example FishEye'.
- '**Timeout**' — '10000'
- '**IP Addresses**' — Leave this blank, unless you are using a proxy server (if so, see the Trusted Applications documentation for further instructions).
- '**URL Patterns**' — Enter the following URLs: `/sr/jira.issueviews:searchrequest, /secure/RunPortlet, /rest, /rpc/soap`

✓ *You may also want to enable issues activity in your activity stream. To do this, navigate to your Application Links and click the '**JIRA settings**' link next to your application link. Tick the 'Include in '**Activity Streams**' checkbox and click '**Save**'.*

**Congratulations, you've just created an application link!**

## Configuring Project Links across Applications

Let's assume that you are managing a project or team. You would like to connect your project's Confluence space with your JIRA project, and link up your team's source repository too.

When you have connected your applications via Application Links, you can also connect the areas of those applications that contain information relating to your project or team. Using **project links** (also called **entity links**) you can associate one or more projects, spaces and repositories across the linked applications.

To connect all the information relating to the project or team that you are managing, you can link one or more of the following:

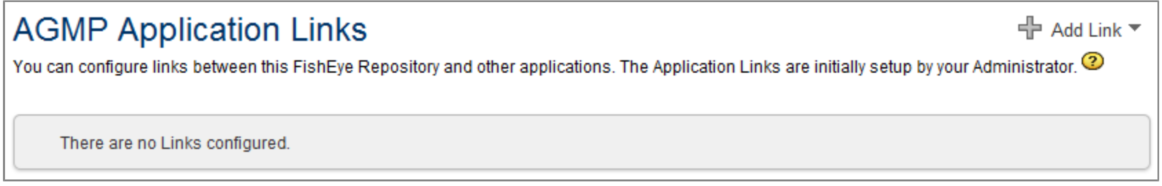
- JIRA projects.
- Confluence spaces.
- FishEye repositories.
- FishEye projects. A FishEye 'project' is the Crucible project if you have installed FishEye and Crucible, otherwise it is the paths associated via the 'FishEye Project Content' function in FishEye.
- Crucible projects.

- Bamboo projects.

In the following example, we'll create an **two-way project link** between a JIRA project (project key is 'MYPROJECT') and a FishEye repository (repository key is 'MYREPO'), for the application link we previously created.

First, navigate to the FishEye administration console and find the repository that you want to link from. Click the ⚙ icon and select 'Application Links' from the dropdown menu. The project links screen will be displayed.

*Screenshot: Project links for a FishEye repository*



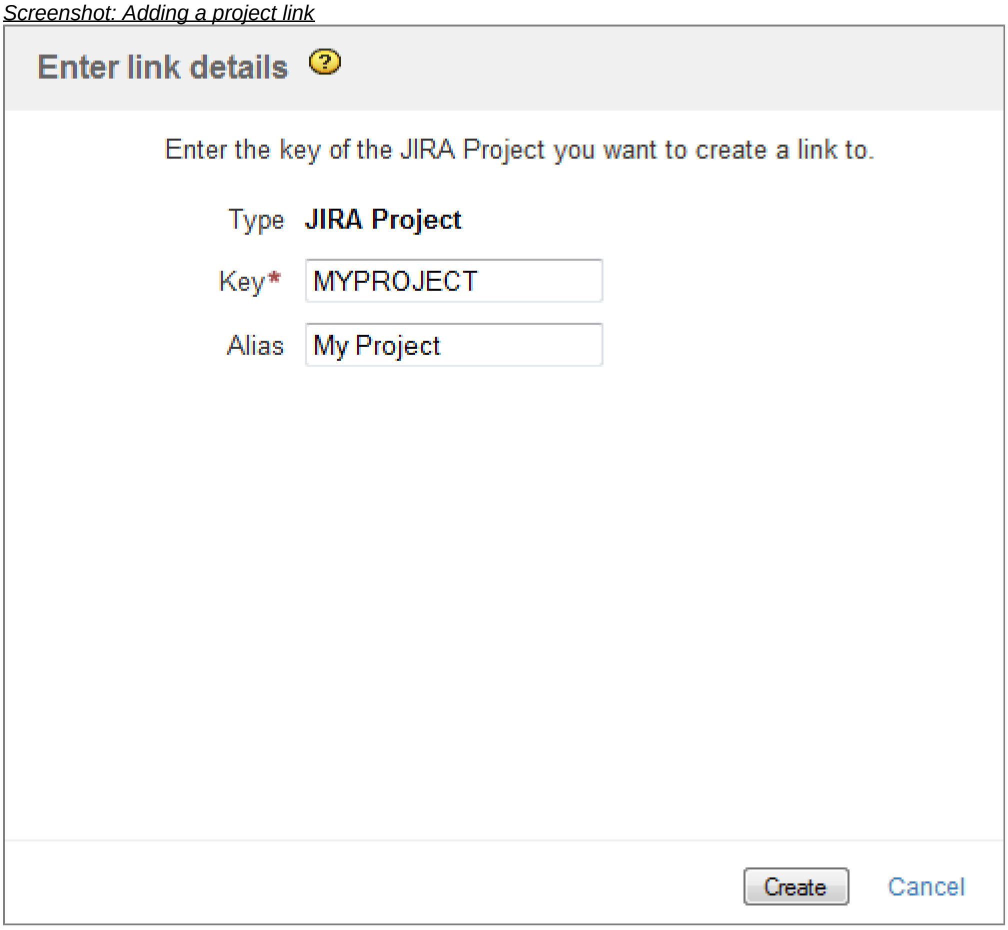
AGMP Application Links

You can configure links between this FishEye Repository and other applications. The Application Links are initially setup by your Administrator.

There are no Links configured.

Click '**Add Link**' and click the remote application where the target project is located. In this case, it will be the 'Example JIRA' application that we linked to previously. The add project link dialogue will be displayed.

*Screenshot: Adding a project link*



Enter link details

Enter the key of the JIRA Project you want to create a link to.

Type **JIRA Project**

Key* MYPROJECT

Alias My Project

Enter the key ('MYPROJECT') and an alias, i.e. display name, for the project ('My Project'). Click '**Create**' to create the project link.

Congratulations, you've just added a project link!

## What Next?

You've now successfully linked your JIRA server to a FishEye server. Try out some of the integration features enabled by your new application link, including:

- Viewing the FishEye changesets in a JIRA project or issue
- Viewing JIRA issues in your FishEye activity
- Add the FishEye Charts Gadget and Recent Changes to your JIRA dashboard.

# Application Links Administrator's Guide

**Application Links (AppLinks)** is a bundled plugin that allows you to link your JIRA, Confluence, FishEye, Crucible and Bamboo applications. Linking two applications allows you to share information and access one application's functions from within the other. For example, if you link JIRA and Confluence, you can view JIRA issues on a Confluence page via the JIRA Issues macro. You can even link your individual projects, spaces and repositories with each other, across the different applications.

Note that the Application Links plugin is bundled and shipped with the Atlassian applications. You cannot install it yourself. Applications Links is bundled with FishEye 2.4, Confluence 3.5, JIRA 4.3, and all later versions of those applications. In addition, Bamboo 3.1 is compatible with AppLinks. You can configure JIRA-to-Bamboo links via the JIRA administration screens.

## Getting Started

The quick start guide describes how to set up an application link between two applications with two-way Trusted Applications authentication:

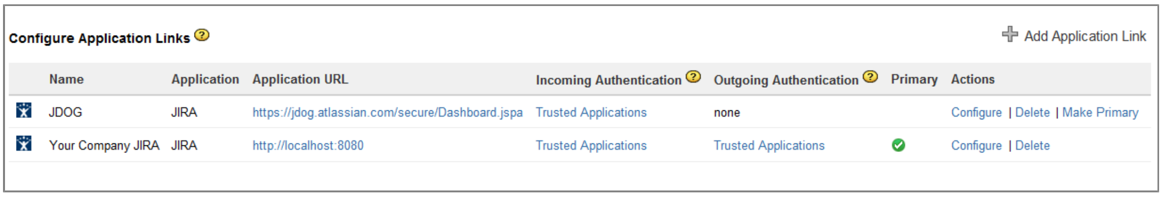
- Application Links Quick Start Guide

## Administrator's Guide

- Configuring Application Links
- Administering Entity Links for an Application Link

# Configuring Application Links

An application link is a trust relationship between two applications. Linking two applications allows you to share information and access one application's functions from within the other.



Configure Application Links ?

Add Application Link

| | Name | Application | Application URL | Incoming Authentication ? | Outgoing Authentication ? | Primary | Actions |
|---|---|---|---|---|---|---|---|
| | JDOG | JIRA | https://jdog.atlassian.com/secure/Dashboard.jspa | Trusted Applications | none | | Configure \| Delete \| Make Primary |
| | Your Company JIRA | JIRA | http://localhost:8080 | Trusted Applications | Trusted Applications | ✓ | Configure \| Delete |

*Screenshot above: Application links for a Confluence server*

## Related Topics

- Adding an Application Link
- Configuring Authentication for an Application Link
- Editing an Application Link
- Making an Application Link the Primary Link
- Deleting an Application Link
- Relocate an Application Link
- Upgrade an Application Link

# Adding an Application Link

This page describes how to add a new application link. The process for adding an application link is different depending on whether the application you are linking to has Application Links installed. If you are linking to a an application that does not have Application Links, you will need to do additional configuration. This is because Application Links in one application will not be able to automatically configure authentication in the application that does not have Application Links.

Please read the appropriate set of instructions below:

- Linking to an application that supports Application Links.
- Linking to an application that does not support Application Links.

**On this page:**

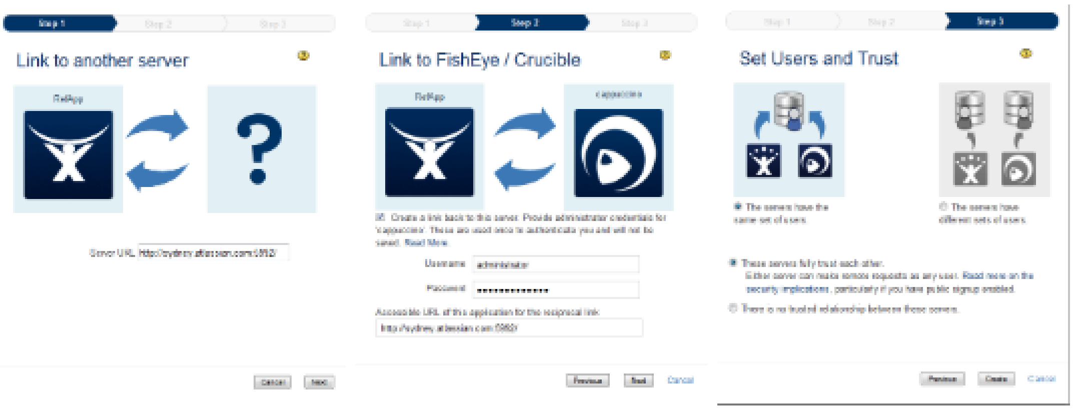
### Adding an Application Link to an Application that Includes Application Links

Before you begin:

- Make sure that the base URL is set correctly in each application which you intend to link, (JIRA instructions | Confluence instructions | FishEye/Crucible instructions | Bamboo instructions). This is required for synchronisation to work correctly.

**To link to an application that includes Application Links:**

1. Log in as a system administrator and go to the administration page. Click 'Application Links' in the administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
2. Click '**Add Application Link**'. Step 1 of the link wizard will appear.
3. Enter the **server URL** of the application that you want to link to (the 'remote application').
4. Click the '**Next**' button. Step 2 of the link wizard will appear.
5. Enter the following information:
    - '**Create a link back to this server**' – Tick this check box if you want to create a two-way link between the remote application and your application. If you want to do this, you will need to enter the username and password of an administrator for the remote application.
      *Note:* These credentials are only used to authenticate you to the remote applicaiton, so that Application Links can make the changes required for the new link. The credentials are not saved.
    - '**Reciprocal Link URL**' – The URL you give here will override the base URL specified in your remote application's administration console, for the purposes of the application links connection. Application Links will use this URL to access the remote application.
6. Click the '**Next**' button. Step 3 of the link wizard will appear.
7. Enter the information required to configure authentication for your application link:
    - '**The servers have the same set of users**' or '**The servers have different sets of users**' – Select one of these options depending on how you manage users between the two applications.
    - '**These servers fully trust each other**' – Tick this check box if you know that the code in both applications will behave itself at all times and are sure each application will maintain the security of its private key.
      *For more information about configuring authentication, see Configuring Authentication for an Application Link.*
8. Click the '**Create**' button to create the application link.



Server URL http://sydney.atlassian.com:8952/

☑ Create a link back to this server. Provide administrator credentials for 'cappuccino'. These are used once to authenticate you and will not be saved. Read More.

Username administrator

Password ••••••••••••••

Accessible URL of this application for the reciprocal link:
http://sydney.atlassian.com:8952/

● The servers have the same set of users.

○ The servers have different sets of users.

● These servers fully trust each other.
Either server can make remote requests as any user. Read more on the security implications, particularly if you have public signup enabled.

○ There is no trusted relationship between these servers.

Step 1
Step 2
Step 3

*Screenshots above: Adding an application link to an application that includes Application Links (click to view full-sized images)*

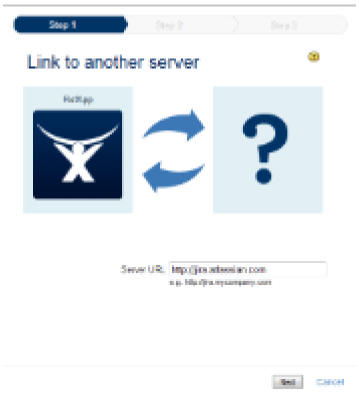
## Adding an Application Link to an Application that Does Not Support Application Links

Before you begin:

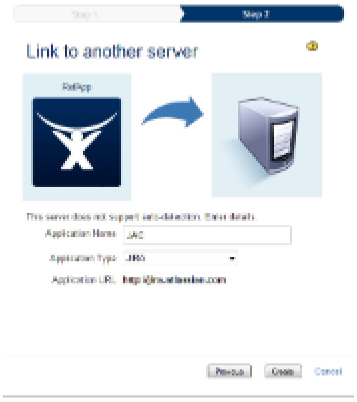
- Make sure that the base URL is set correctly in each application which you intend to link, (JIRA instructions | Confluence instructions | FishEye/Crucible instructions | Bamboo instructions). This is required for synchronisation to work correctly.

**To link to an application that does not support Application Links:**

1. Log in as a system administrator and go to the administration page. Click 'Application Links' in the administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
2. Click '**Add Application Link**'. Step 1 of the 'Link to another server' dialogue will be displayed.
3. Enter the server URL of the application that you want to link to, in the '**Server URL**' field. Click the '**Next**' button. Step 2 of the 'Link to another server' dialogue will be displayed.
4. Fill out the fields, as follows:
    - '**Application Name**' — Enter the name by which this remote application will be referred to, in your application.
    - '**Application Type**' — Select the type of application that you are linking to: Bamboo, FishEye/Crucible, JIRA, Confluence, Subversion.
    - '**Application URL**' — This will be set to the server URL you entered in the previous step and will not be editable.
5. Click the '**Create**' button to create the application link. The 'Configure Application Links' page will be displayed, listing all of the application links that have currently been set up for your application including the one you just added.
6. Configure the desired authentication type (Trusted Applications, OAuth, basic HTTP, none) for your new application link. See Configuring Authentication for an Application Link.
7. In your application that does not support Application Links, configure the same type of authentication that you configured for your application link's *outgoing* authentication (in the previous step). For example, if you configured outgoing Trusted Applications authentication in your Application-Links-enabled application, you also need log into your non-Application-Links application and manually configure Trusted Applications (see the relevant administrator's documentation for the application).



Server URL: http://jira.atlassian.com
e.g. http://jira.mycompany.com



This server does not support auto-detection. Enter details.

Application Name: JAC
Application Type: JIRA
Application URL: http://jira.atlassian.com

Step 1
Step 2

*Screenshots above: Adding an application link to an application that does not support Application Links (click to view full-sized images)*

***Related Topics***

Making an Application Link the Primary Link
Configuring Authentication for an Application Link
Administering Entity Links for an Application Link

# Configuring Authentication for an Application Link

Configuring authentication for an application link is essentially defining the level of trust between the two linked servers.

**On this page:**

- Choosing Authentication for an Application Link
- Security Implications for each Authentication Type
- About Primary Authentication Types
- About Impersonating and Non-Impersonating Authentication Types

## Choosing Authentication for an Application Link

The level of authentication that you should configure for your application link depends on a number of factors.

- Do the two applications you are linking trust each other? i.e. are you sure that the code in the application will behave itself at all times and that the application will maintain the security of its private key?
- Do the two applications you are linking share the same user base or not?
- Do you have administrative access to the application you are linking to?

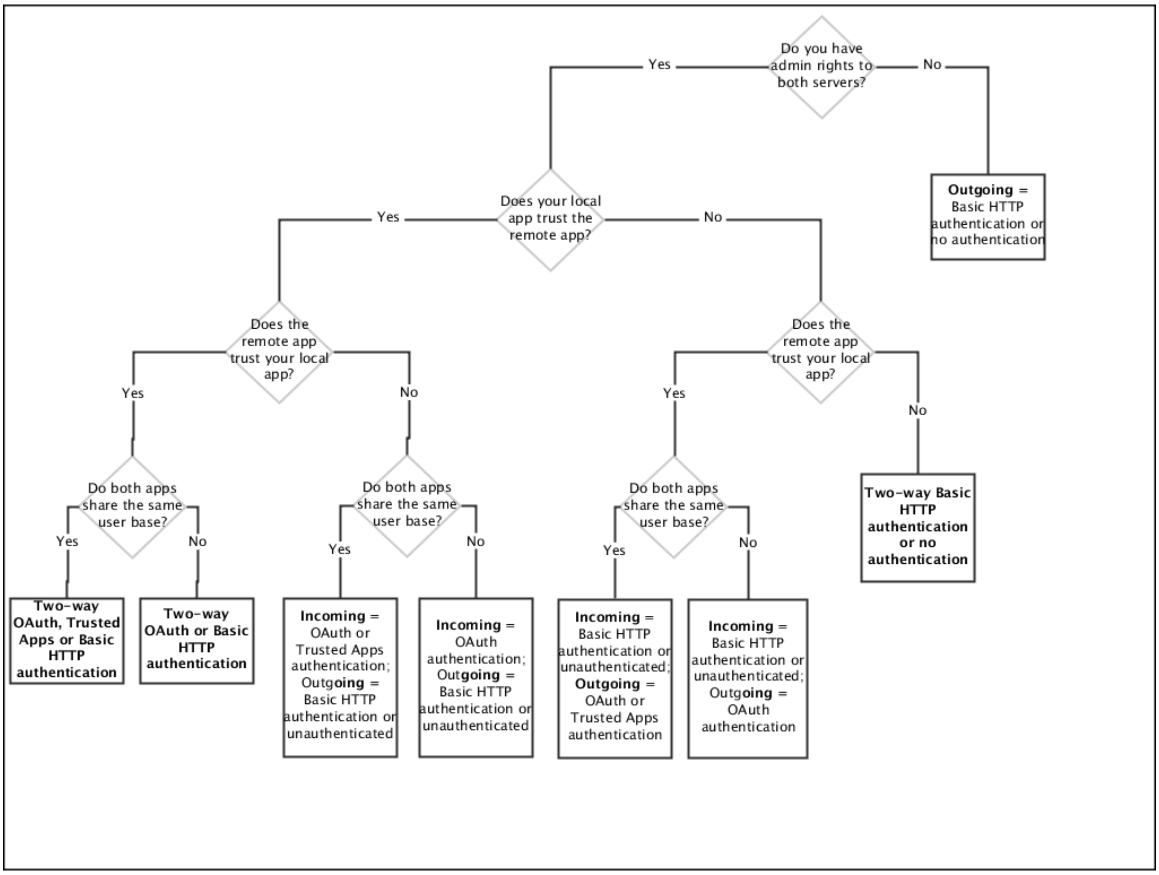
**Common scenarios include:**

- If the two applications you are linking **trust each other** and **share the same user base**, configure **two-way authentication using Trusted Applications** for both incoming authentication (authentication of requests coming from a linked application into this application) and outgoing authentication (authentication of requests sent from this application to a linked application). For example, you may link your internal Confluence instance to an internal JIRA instance.
- If the two applications you are linking **trust each other** but **do not share the same user base**, configure **two-way authentication using OAuth** for both incoming authentication (authentication of requests coming from a linked application into this application) and outgoing authentication (authentication of requests sent from this application to a linked application). For example, you may link your internal Confluence instance to an external (customer-facing) JIRA instance.
- If you **do not have administrative rights to the application that you are linking to** (e.g. linking to a public FishEye instance), configure a **one-way outgoing link** authenticated using **basic HTTP authentication or do not configure any authentication** for the link. For example, you may link your external JIRA instance to a partner organisation's JIRA instance. An unauthenticated link will still allow the local application to render hyperlinks to the remote application or query anonymously-accessible APIs.

The flowchart below provides a guide to what authentication you should configure for your application link.

Read the following topics for information on how to configure authentication for an application link:

- Configuring Basic HTTP Authentication for an Application Link
- Configuring OAuth Authentication for an Application Link
- Configuring Trusted Apps Authentication for an Application Link
- Incoming and Outgoing Authentication


Do you have admin rights to both servers?
Yes
No
Outgoing = Basic HTTP authentication or no authentication
Does your local app trust the remote app?
Yes
No
Does the remote app trust your local app?
Yes
No
Does the remote app trust your local app?
Yes
No
Two-way Basic HTTP authentication or no authentication
Do both apps share the same user base?
Yes
No
Two-way OAuth, Trusted Apps or Basic HTTP authentication
Two-way OAuth or Basic HTTP authentication
Do both apps share the same user base?
Yes
No
Incoming = OAuth or Trusted Apps authentication; Outgoing = Basic HTTP authentication or unauthenticated
Incoming = OAuth authentication; Outgoing = Basic HTTP authentication or unauthenticated
Do both apps share the same user base?
Yes
No
Incoming = Basic HTTP authentication or unauthenticated; Outgoing = OAuth or Trusted Apps authentication
Incoming = Basic HTTP authentication or unauthenticated; Outgoing = OAuth authentication

*Flowchart above: Determining what authentication to configure for an Application Link*

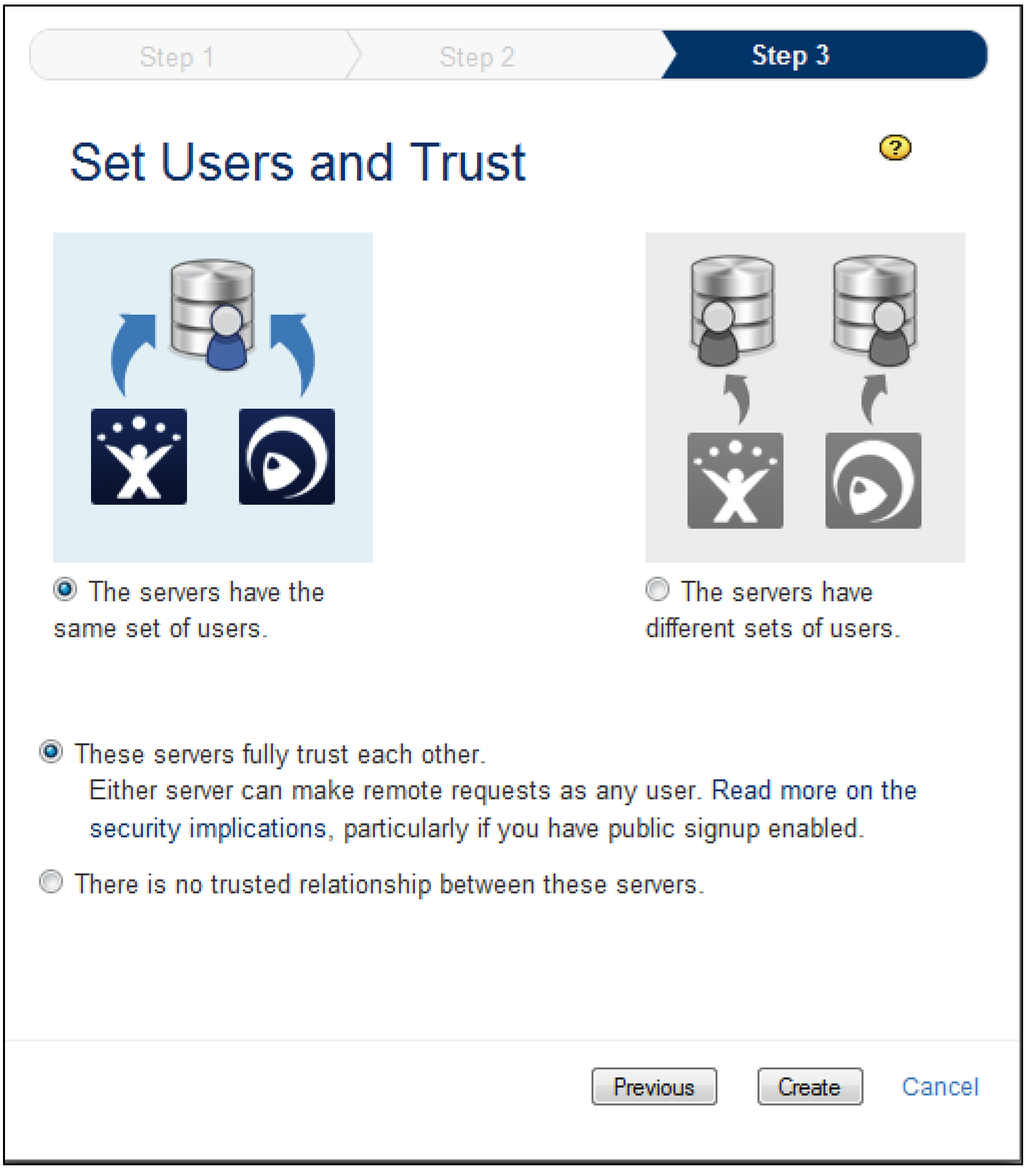
## Security Implications for each Authentication Type

If you configure **Trusted Applications authentication** for your application (your servers have the same set of users and they fully trust each other), please be aware of the following security implications:

- Trusted applications are a **potential security risk**. When you configure Trusted Applications authentication, you are allowing one application to access another as any user. This allows all of the built-in security measures to be bypassed. Do not configure a trusted application unless you know that all code in the application you are trusting will behave itself at all times, and you are sure that the application will maintain the security of its private key.

If you configure **OAuth authentication** for your application (your servers have different sets of users and they fully trust each other), please be aware of the following security implications:

- Adding an OAuth consumer requires the transmission of sensitive data. To prevent 'man-in-the-middle' attacks, it is recommended that you **use SSL** for your applications while configuring OAuth authentication.
- Do not link to an application using OAuth authentication, unless you **trust all code in the application** to behave itself at all times. OAuth consumers are a potential security risk to the applications that they are linked to.



Set Users and Trust

● The servers have the same set of users.

○ The servers have different sets of users.

● These servers fully trust each other.
Either server can make remote requests as any user. Read more on the security implications, particularly if you have public signup enabled.

○ There is no trusted relationship between these servers.

*Screenshot above: Configuring authentication during application link setup*

### About Primary Authentication Types

You can configure multiple authentication types for each application link. When a feature makes a request using an Application Link, it will use one of the configured authentication types. If more than one authentication type is configured, it will by default use the authentication type that is marked as the primary authentication type. The default authentication type is indicated by the green tick ✓ next to the authentication type on the list application link screen.

You **cannot** configure which authentication type is the primary authentication type. The primary authentication type is determined automatically by Application Links and depends on a weight defined by each authentication type method. However, every feature that uses Application Links can also choose to use a specific authentication type and might not use the default primary authentication type.

### About Impersonating and Non-Impersonating Authentication Types

Applications Links allows you to configure 'impersonating' and 'non-impersonating' authentication types:

- **Impersonating authentication types** make requests on behalf of the user who is currently logged in. People will see only the information that they have permission to see. This includes OAuth and Trusted Applications authentication.
- **Non-impersonating authentication types** always use a pre-configured user when making a request. Everyone logged into the system will see the same information. This includes basic HTTP authentication.

## Configuring Basic HTTP Authentication for an Application Link

The instructions on this page describe how to configure **Basic HTTP authentication** for an application link. You can configure outgoing authentication (authentication of requests sent from this application to a linked application) and/or incoming authentication (authentication of requests coming from a linked application into this application).

Basic HTTP authentication allows your application to provide user credentials to a remote application and vice versa. Once authenticated, one application can access specified functions on the other application on behalf of that user. For example, if you supply the credentials of a JIRA administrator on your JIRA server to a remote application, the remote application will be able to access all functions on your JIRA server that the JIRA administrator can access.

This method of authentication relies on the connection between your application and the remote application being secure. We recommend that you use Trusted Applications authentication or OAuth authentication for your application link instead, if possible.

**On this page:**

### *Before You Begin*

- The instructions assume that **both of the applications that you are linking have the Application Links plugin installed**. If the remote application that you are linking to supports Basic HTTP authentication, but does not have the Application Links plugin installed, you will need to configure Basic HTTP authentication from within the remote application (see the relevant administrator's documentation for the application) in addition to configuring the outgoing/incoming authentication for the application link (as described below).

### *Configuring Basic HTTP Authentication for Outgoing Authentication*

Configuring **outgoing basic http authentication** will allow your application to trust a remote application (i.e. allow the remote application to access specified functions in your application).

**To configure basic http authentication for an outgoing application link:**

1. Log in as a system administrator and go to the administration page. Click 'Application Links' in the administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
2. Click the '**Configure**' link next to the application link that you want to configure authentication for.
3. Click the '**Outgoing Authentication**' tab. The outgoing authentication page will be displayed.
4. Click the '**Basic Access**' tab.
5. Click the '**Configure**' button and enter the credentials (username and password) that the remote application will use to log into your application .
6. Click the '**Apply**' button to save your changes.

### *Configuring Basic HTTP Authentication for Incoming Authentication*

Configuring **incoming basic http authentication** will allow the remote application that you are linking to, to trust your application (i.e. allow your application to access specified functions on the remote application it is linked to).

**To configure basic http authentication for an incoming application link:**

1. Log in as a system administrator and go to the administration page. Click 'Application Links' in the administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
2. Click the '**Configure**' link next to the application link that you want to configure authentication for.
3. Click the '**Incoming Authentication**' tab. The incoming authentication page will be displayed.
4. Click the '**Basic Access**' tab.
5. Click the '**Configure**' button and enter the credentials (username and password) that the your application will use to log in to the remote application.
6. Click the '**Apply**' button to save your changes.

### *Notes*

Related Topics

Configuring OAuth Authentication for an Application Link
Configuring Trusted Apps Authentication for an Application Link

## Configuring OAuth Authentication for an Application Link

The instructions on this page describe how to configure **OAuth** for for an application link. You can configure outgoing authentication (authentication of requests sent from this application to a linked application) and/or incoming authentication (authentication of requests coming from a linked application into this application).

OAuth is a protocol that allows a web application to share data/resources with any other OAuth-compliant external application. These external applications could be another web application (such as a different JIRA installation or an iGoogle home page), a desktop application or a mobile device application, provided that they are accessible from within your network or available on the Internet.

For example, you could set up an application link between a JIRA server and an iGoogle page using OAuth authentication. This would allow you to view data from your JIRA server in a JIRA dashboard gadget on the iGoogle page.

A typical scenario is setting up an application link between two applications which trust each other, do not share the same set of users but both applications have the application links plugin installed. In this case, you would configure OAuth for both outgoing authentication and incoming authentication. See Configuring Authentication for an Application Link for other configurations.

**Key OAuth Terminology**

- **Service provider** — An application that shares ('provides') its resources.
- **Consumer** — An application that accesses ('consumes') a service provider's resources.
- **User** — An individual who has an account with the Service Provider.

For more information about OAuth, see Configuring OAuth as well as the OAuth specification.

**On this page:**

## *Before You Begin*

- Adding an OAuth consumer requires the transmission of sensitive data. To prevent 'man-in-the-middle' attacks, it is recommended that you **use SSL** for your applications while configuring OAuth authentication.
- Do not link to an application using OAuth authentication, unless you **trust all code in the application** to behave itself at all times. OAuth consumers are a potential security risk to the applications that they are linked to.

- The instructions assume that **both of the applications that you are linking have the Application Links plugin installed**. If the remote application that you are linking to supports OAuth, but does not have the Application Links plugin installed, you will need to configure OAuth from within the remote application (see the relevant administrator's documentation for the application) in addition to configuring the outgoing/incoming authentication for the application link (as described below).

## *Configuring OAuth for Outgoing Authentication*

Configuring **outgoing OAuth authentication** will allow your application to access data in a remote application on behalf of a user (i.e. allow this application to access specified functions in the remote application).

**To configure OAuth authentication for an outgoing application link:**

1. Log in as a system administrator and go to the administration page. Click 'Application Links' in the administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
2. Click the '**Configure**' link next to the application link that you want to configure OAuth for.
3. Click the '**Outgoing Authentication**' tab. The outgoing authentication page will be displayed.
4. Click the '**OAuth**' tab.
5. If you are not currently logged in to the remote application (or you logged in to the remote application under a variant of the application's hostname, such as the IP address), a login dialogue will display.
    - Enter the '**Username**' and '**Password**' for the remote server, not your local server, and click the '**Login**' button. The remote server needs to learn the identity of your local server for the OAuth protocol to work and your admin credentials are used to store your local server's public key on the remote server. If you are already logged into your remote server, then the appropriate changes can be made without having to log in again.
6. Click the '**Enable**' button to enable OAuth authentication for the outgoing link. Your application will be automatically set up to be the 'consumer' and the remote application as a 'service provider'.

## *Configuring OAuth for Incoming Authentication*

Configuring **incoming OAuth authentication** will allow the remote application that you are linking to, to access data in your application on behalf of its users.

**To configure OAuth authentication for an incoming application link:**

1. Log in as a system administrator and go to the administration page. Click 'Application Links' in the administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
2. Click the '**Configure**' link next to the application link that you want to configure OAuth for.
3. Click the '**Incoming Authentication**' tab. The incoming authentication page will be displayed.
4. Click the '**OAuth**' tab.
5. Click the '**Enable**' button to enable OAuth authentication for the incoming link. The remote application will be automatically set up to be the 'consumer' and your local application as a 'service provider'.

## *Notes*

Related Topics

Configuring Basic HTTP Authentication for an Application Link
Configuring Trusted Apps Authentication for an Application Link

## Configuring Trusted Apps Authentication for an Application Link

The instructions on this page describe how to configure **Trusted Applications** for an application link. You can configure outgoing authentication (authentication of requests sent from this application to a linked application) and/or incoming authentication (authentication of requests coming from a linked application into this application).

Trusted Applications authentication allows one application to allow access to specified functions on another application on behalf of any user, without the user having to log into the second application. For example, if you configure a JIRA server to trust a Confluence server, every Confluence user will see exactly the same list of issues when they view the Confluence 'JIRA Issues' macro as they see when they use the JIRA issue navigator as a logged-in JIRA user.

A typical scenario is setting up an application link between two applications which trust each other, have the same set of users and both have the application links plugin installed. In this case, you would configure Trusted Applications for both outgoing authentication and incoming authentication. See Configuring Authentication for an Application Link for other configurations.

**On this page:**

### *Before You Begin*

- Trusted applications are a **potential security risk**. When you configure Trusted Applications authentication, you are allowing one application to access another as any user. This allows all of the built-in security measures to be bypassed. Do not configure a trusted application unless you know that all code in the application you are trusting will behave itself at all times, and you are sure that the application will maintain the security of its private key.

- The instructions below assume that **both of the applications that you are linking have the Application Links plugin installed**. If the remote application that you are linking to supports Trusted Applications, but does not have the Application Links plugin installed, you will need to configure Trusted Applications from within the remote application (see the relevant administrator's documentation for the application) in addition to configuring the outgoing/incoming authentication for the application link (as described below).

### *Configuring Trusted Applications for Outgoing Authentication*

Configuring **outgoing Trusted Applications authentication** will allow the remote application to trust your local application (i.e. allow your application to access specified functions and data on the remote application).

**To configure Trusted Applications authentication for an outgoing application link:**

1. Log in as a system administrator and go to the administration page. Click 'Application Links' in the administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
2. Click the '**Configure**' link next to the application link that you want to configure Trusted Applications authentication for.
3. Click the '**Outgoing Authentication**' tab. The outgoing authentication page will show, with the '**Trusted Applications**' tab displayed.
4. If you are not currently logged into the remote application (or you logged into the remote application under a variant of the application's hostname, e.g. the IP address), a login dialogue will display.
    - Enter the '**Username**' and '**Password**' for the remote server, (not your local server), and click the '**Login**' button. You need to enter the credentials for the remote server, as the remote server needs to be instructed to trust your local server for the Trusted Applications protocol to work. If you are already logged into your remote server, then the appropriate changes can be made without having to log in again.
5. Configure the settings for the Trusted Applications authentication:
    - '**IP Patterns**' — Enter the IP addresses (IPv4 only) from which the remote application will accept requests (this effectively is the IP address your local server). You can specify wildcard matches by using an asterisk (*), e.g. '`192.111.*.*`' (note, you cannot use netmasks to specify network ranges). If you are entering multiple IP addresses, separate them with commas or spaces.
      ⚠ *Please note, if you are setting up Trusted Applications between two applications that both have the Application Links plugin installed, you can leave this field blank (or explicitly use `*.*.*.*`). However, if your remote application does not have the Application Links plugin installed and you are configuring the IP Patterns in the remote application (not the Application Links plugin),* ***you must not leave this field blank nor use `*.*.*.*`****. Failure to configure IP address restrictions in this scenario is a security vulnerability, allowing an unknown site to log into your site under a user's login ID.*
      Consider the following scenarios, if you want to limit access by using this field:
        - If your local application is using a proxy server, you need to add the proxy server's IP address to this field.
        - If your local application is a clustered instance of Confluence, you need to configure the remote server to accept requests from each cluster node. If you do not set up each node appropriately, your Confluence users may not be able to view any information from the remote server. You can set this up by either specifying each individual IP address for each node of the cluster (e.g. `172.16.0.10, 172.16.0.11, 172.16.0.12`), or specifying the IP address for the clustered Confluence instance using wildcards (e.g. `172.16.0.*`).
    - **'URL Patterns'** — Enter the URLs in the remote application that your local application will be allowed to access. Each URL corresponds to a particular application function. Enter one URL per line, as follows:
        - If your remote application is JIRA, enter the following URL Patterns: `/plugins/servlet/streams`,

        `/sr/jira.issueviews:searchrequest`, `/secure/RunPortlet`, `/rest`, `/rpc/soap`
    - If your remote application is Confluence, enter the following URL Patterns: `/plugins/servlet/streams`, `/plugins/servlet/applinks/whoami`
  - '**Certificate Timeout (ms)**' — Enter the certificate timeout. The default is 10 seconds. The certificate timeout is used to prevent replay attacks. For example, if a Trusted Applications request is intercepted and (maliciously) re-sent, the application will be able to check when the request was first sent. If the second request is sent more than 10 seconds (or whatever the certificate timeout is set to) after the initial request, it will be rejected. Please note, you should not have to change the default value of this field for most application links. Note that the certificate timeout relies on the clocks on both servers being synchronised.

6. Click the '**Apply**' button to save your changes.

### *Configuring Trusted Applications for Incoming Authentication*

Configuring **incoming Trusted Applications authentication** will allow your local application to trust the remote application that you are linking it to (i.e. allow your 'trusted' remote application to access specified functions and data on your local application).

**To configure Trusted Applications authentication for an incoming application link:**

1. Log in as a system administrator and go to the administration page. Click 'Application Links' in the administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
2. Click the '**Configure**' link next to the application link that you want to configure Trusted Applications authentication for.
3. Click the '**Incoming Authentication**' tab. The imconing authentication page will show, with the '**Trusted Applications**' tab displayed.
4. The tab will show whether Trusted Applications is currently enabled or not. Use the '**Modify**' or '**Configure**' button to configure Trusted Applications. The Trusted Applications configuration settings will be displayed:
   - '**IP Patterns**' — Enter the IP addresses (IPv4 only) from which our application will accept requests. You can specify wildcard matches by using an asterisk (*), e.g. '`192.111.*.*`' (note, you cannot use netmasks to specify network ranges). If you are entering multiple IP addresses, separate them with commas or spaces.

     ⚠ *Please note, if you are setting up Trusted Applications between two applications that both have the Application Links plugin installed, you can leave this field blank (or explicitly use `*.*.*.*`). However, if your remote application does not have the Application Links plugin installed and you are configuring the IP Patterns in the remote application (not the Application Links plugin),* ***you must not leave this field blank nor use `*.*.*.*`****. Failure to configure IP address restrictions in this scenario is a security vulnerability, allowing an unknown site to log into your site under a user's login ID.*
     Consider the following scenarios, if you want to limit access by using this field:
     - If the remote application is using a proxy server, you need to add the proxy server's IP address to this field.
     - If the remote application is a clustered instance of Confluence, you need to accept requests from each cluster node. If you do not specify each node's address, Confluence users may not be able to view any data from your application. You can set this up by either specifying each individual IP address for each node of the cluster (e.g. 172.16.0.10, 172.16.0.11, 172.16.0.12), or specifying the IP address for your clustered Confluence instance using wildcards (e.g. 172.16.0.*).
   - **'URL Patterns'** — Enter the local URLs that the remote application will be allowed to access. Each URL corresponds to a particular application function. Enter one URL per line, as follows:
     - If your local application is JIRA, enter the following URL Patterns — `/plugins/servlet/streams`, `/sr/jira.issueviews:searchrequest`, `/secure/RunPortlet`, `/rest`, `/rpc/soap`
     - If your local application is Confluence, enter the following URL Patterns — `/plugins/servlet/streams`, `/plugins/servlet/applinks/whoami`
   - '**Certificate Timeout (ms)**' — Enter the certificate timeout. The default is 10 seconds. The certificate timeout is used to prevent replay attacks. For example, if a Trusted Applications request is intercepted and (maliciously) re-sent, the application will be able to check when the request was first sent. If the second request is sent more than 10 seconds (or whatever the certificate timeout is set to) after the initial request, it will be rejected. Please note, you should not have to change the default value of this field for most application links. Note that the certificate timeout relies on the clocks on both servers being synchronised.
5. Click the '**Apply**' button to save your changes.

### *Notes*

Related Topics

Configuring Basic HTTP Authentication for an Application Link
Configuring OAuth Authentication for an Application Link

## Incoming and Outgoing Authentication

When you configure authentication for an application link, you are defining the level of trust between the two linked servers. When configuring a link from one application to another, you can set up:

- **Incoming authentication** (authentication of requests coming from a linked application into this application).
- **Outgoing authentication** (authentication of requests sent from this application to a linked application).

See Configuring Authentication for an Application Link.

RELATED TOPICS

Application Links Quick Start Guide
Application Links Administrator's Guide

# Editing an Application Link

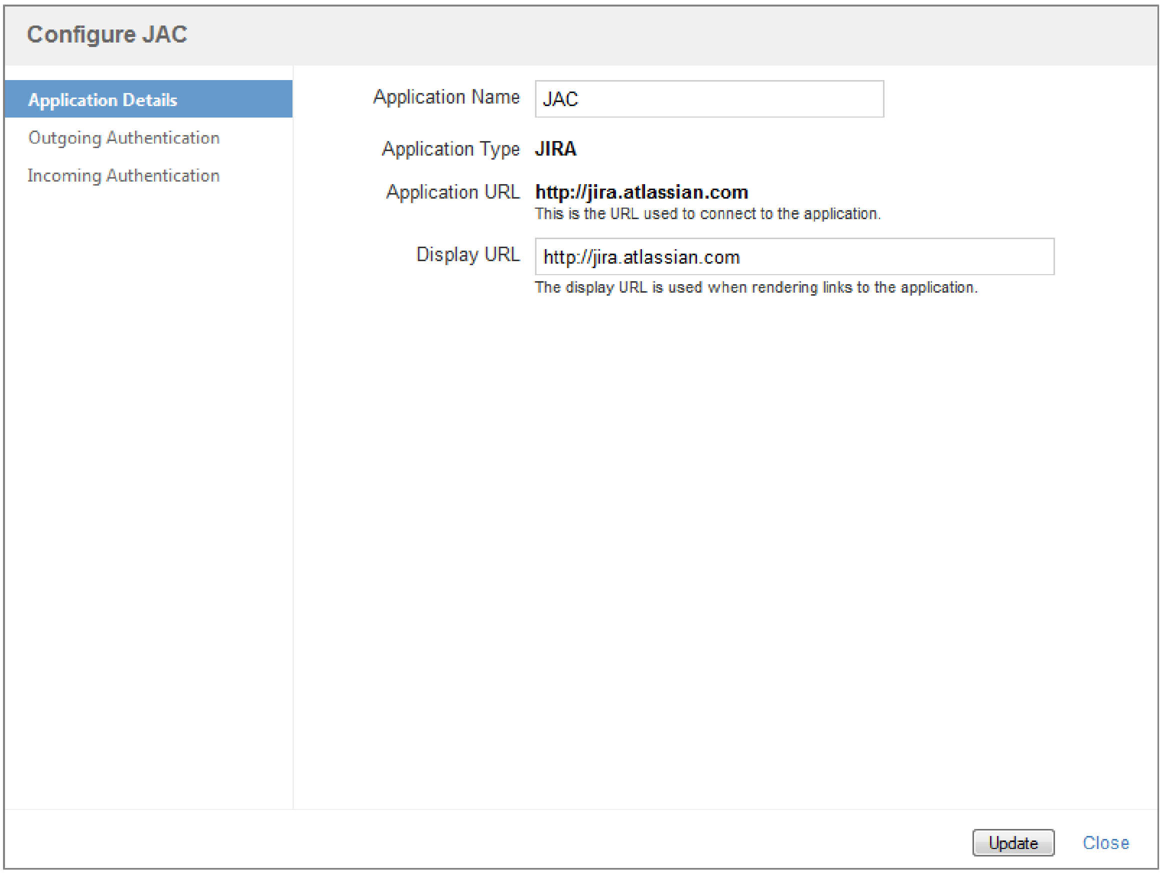
You can change the details, such as the application name and display URL, for an existing application link.

**On this page:**

## Editing an Application Link

**To edit an application link:**

1. Log in as a system administrator and go to the administration page. Click 'Application Links' in the administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
2. Click the '**Configure**' link next to the application link that you want to edit the details for. The application details for the application link will be displayed.
3. Update the application details as desired. Please note, you cannot update the Application Type nor the Application URL.
    - '**Application Name**' — Update this field to change the display name for the application that you are linking to.
    - '**Display URL**' — This URL is used when displaying links to the application in the browser. When creating the application link, you may have used a URL that is not accessible to other users, such as an internal IP address. If so, you can change the display URL to an address in a domain that is accessible to other users.
4. Click the '**Update**' button to save your changes.



Configure JAC

Application Name: JAC

Application Type: JIRA

Application URL: http://jira.atlassian.com

This is the URL used to connect to the application.

Display URL: http://jira.atlassian.com

The display URL is used when rendering links to the application.

*Screenshot above: Editing an application link*

## Notes

***Related Topics***

Configuring Authentication for an Application Link
Making an Application Link the Primary Link
Relocate an Application Link

# Making an Application Link the Primary Link

If you have set up application links to more than one of the same application type, e.g. you have linked your application to two JIRA servers, then one of the servers will be marked as the 'Primary' link. This means that any outgoing requests will be directed to the primary link's application.

For example, if you have set up a Confluence server that is linked to two JIRA servers with two-way authentication for both links, you can nominate an application link to one of the JIRA servers as the primary link. Every time Confluence requests JIRA information (e.g. for a JIRA issues macro), it will request it from the primary link's JIRA server. Note, both JIRA servers can still make requests of the Confluence server (e.g. a Confluence page gadget on the dashboards of each JIRA instance).

Please read about making a project link the primary link, for information on how primary project links also influence the information shared between servers.

**On this page:**

## Making an Application Link the Primary Link

**To make an application link the primary link:**

1. Log in as a system administrator and go to the administration page. Click 'Application Links' in the administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
2. Click the '**Make Primary**' link next to the application link that you want to make the primary link. A '✓' symbol will display in the 'Primary' column next to the application link.
   ℹ *The 'Primary' column and 'Make Primary' link will only display if you have set up application links to more than one of the same application type, e.g. you have linked your application to two JIRA servers.*

## Notes

***Related Topics***

Making an Entity Link the Primary Link

# Deleting an Application Link

Deleting an application link stops the two applications from sharing information. You will no longer be able to make requests from one application to the other. This means that certain features may not work, e.g. JIRA issues macro in Confluence, Confluence Page Gadget in JIRA, etc.

If you have set up application links to multiple servers of the same application type, e.g. you have linked your application to multiple JIRA servers, deleting the primary link will mean that another of the links will be made the primary link.

Deleting an application link will also delete all project links set up for that application link.

**On this page:**

## Deleting an Application Link

**To delete an application link:**

1. Log in as a system administrator and go to the administration page. Click 'Application Links' in the administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
2. Click the '**Delete**' link next to the application link that you want to delete. A confirmation screen will be displayed.
3. Click the '**Confirm**' button to delete the application link.

## Notes

***Related Topics***

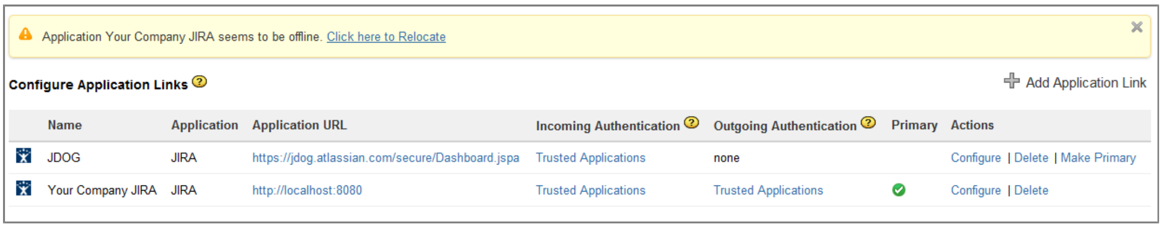
Editing an Application Link
Relocate an Application Link

# Relocate an Application Link

This page describes how to change the location of an application link. You will need to relocate an application link if the target application has been moved to a new address.

**To relocate an application link:**

1. Log in as a system administrator and go to the administration page. Click 'Application Links' in the administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
2. If the remote application for an application link cannot be reached by your application, the '**List Application Links**' page will display a warning message (see 'Relocate Link - Warning Message' screenshot below).
3. If your remote application has been moved to a different address (rather than just being offline temporarily), click the '**Relocate**' link in the warning message (see 'Relocate Link - Updating URL' screenshot below).
4. Enter the new URL for the remote application of your application link and click '**Relocate**'.
5. You will need to confirm the relocation, if the new URL cannot be contacted. Otherwise, the application link will be updated.

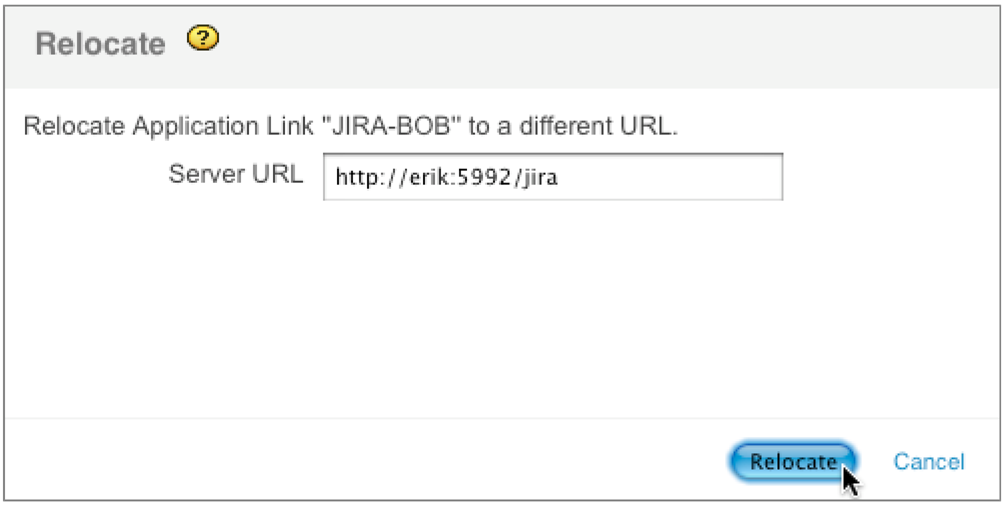


Application Your Company JIRA seems to be offline. Click here to Relocate

Configure Application Links

| Name | Application | Application URL | Incoming Authentication | Outgoing Authentication | Primary | Actions |
|---|---|---|---|---|---|---|
| JDOG | JIRA | https://jdog.atlassian.com/secure/Dashboard.jspa | Trusted Applications | none | | Configure \| Delete \| Make Primary |
| Your Company JIRA | JIRA | http://localhost:8080 | Trusted Applications | Trusted Applications | ✓ | Configure \| Delete |

*Screenshot above: The warning message prompting you to relocate an application link*



Relocate

Relocate Application Link "JIRA-BOB" to a different URL.

Server URL: http://erik:5992/jira

*Screenshot above: Relocate link – Updating the URL*

***Related Topics***

Making an Application Link the Primary Link

## Upgrade an Application Link

The instructions on this page describe how to upgrade an existing application link. You may want to upgrade an application link in either of the two situations below:

- Your local application has been upgraded from a version that does not include Application Links to a version that does. For example, you may have configured Trusted Applications or OAuth in a JIRA 4.2 instance (does not include Application Links) and then upgraded to JIRA 4.3 (includes Application Links).
- Your remote application has been upgraded to a version that includes Application Links. For example, you had set up an application link in a FishEye 2.4 instance (includes Application Links) to a JIRA 4.2 instance (does not include Application Links), and then upgrade to JIRA 4.3 (includes Application Links).

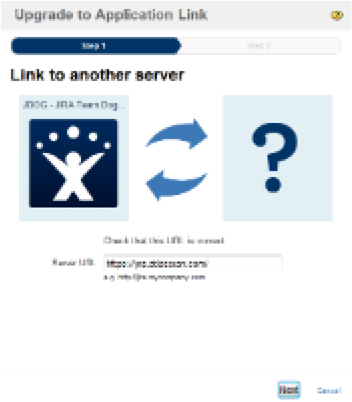
**On this page:**

### Upgrading an Application Link (Local App Upgraded to Include Application Links)

When you upgrade from an application that does not include Application Linksto application that does, you will have the option of converting any Trusted Applications or OAuth links to Application Links. The advantage of converting your links to Application Links are that link configuration will be simplified in future.

**To upgrade an application link when your local application has been upgraded to include Application Links:**

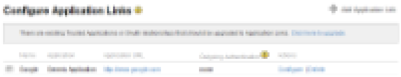
1. After your application upgrade, navigate to the administration console.
2. Click '**Application Links**'. The 'Configure Application Links' screen will be displayed with the following message: *"There are existing Trusted Applications or OAuth relationships that should be upgraded to Application Links.* ***Click here to upgrade.****"*
3. Click the '**Click here to upgrade**' link. The 'Existing Trust Relationships' screen will be displayed showing all Trusted Applications and OAuth relationships that can be upgraded to Application Links.
4. Click the '**Upgrade to Application Link**' link next to the desired trust relationship. The 'Upgrade to Application Link' wizard will be displayed.
5. Complete the wizard. The process will be similar to adding a new link (described on Adding an Application Link), except that most fields should be pre-filled.

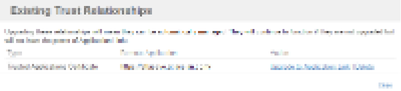


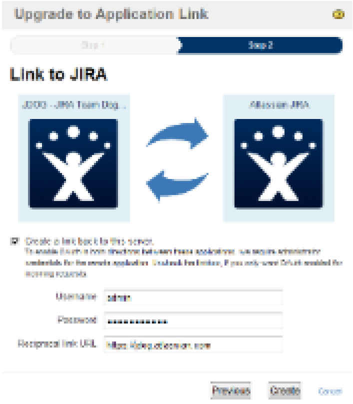
Check that this URL is correct.

Server URL: https://jira.atlassian.com/

e.g. http://jira.mycompany.com




Existing Trust Relationships
Type
Remote Application
Action
Trusted Applications Certificate
Upgrade to Application Link
Done



☑ Create a link back to this server.

Username: admin

Password: ••••••••••••

Reciprocal link URL: https://jira.atlassian.com

Step 1
Step 2
Step 3
Step 4

*Screenshots above: Upgrading an application link for local application*

## Upgrading an Application Link (Remote App Upgraded to Include Application Links)

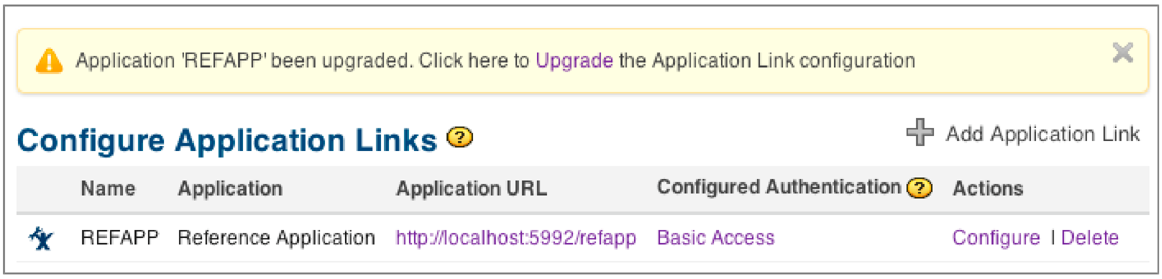
When an application link is created between an Application Links-enabled application and a remote legacy application (either a non-Atlassian product, or an older version of an Atlassian product that did not ship with Application Links), this link is configured to run in "legacy mode". While there is no distinguishable difference to a user, connecting and configuring links without Application Links is a little different. For example:

- Setting up OAuth requires manual configuration by the administrator. In OAuth authentication when both sites have Application Links, exchange of the consumer keys and public keys is done automatically.
- The Trusted Applications protocol (Atlassian-specific) will not be available for authentication.

If you upgrade your remote application to a version that does include Application Links, the application link will continue to work. However, upgrading your link may simplify link configuration and make additional authentication protocols available (as mentioned above).

**To upgrade an application link when your remote application has been upgraded to include Application Links:**

1. After you have upgraded your remote application to a version that includes Application Links, go to the administration console of your local application. A warning will be displayed, requesting that you upgrade the link to full Application Links mode.
2. Click '**Upgrade**' in the warning message to start the upgrade wizard. Note the following:
    - You will be prompted to make your application link a reciprocal link. You will need to provide administrator credentials for your remote application, if you choose to do so.
    - If you make your application link a reciprocal link, you will also be able to make reciprocal links for your project links. For example, you may be able to link your JIRA project to a FishEye repository and also make a link from your FishEye repository back to the JIRA project.

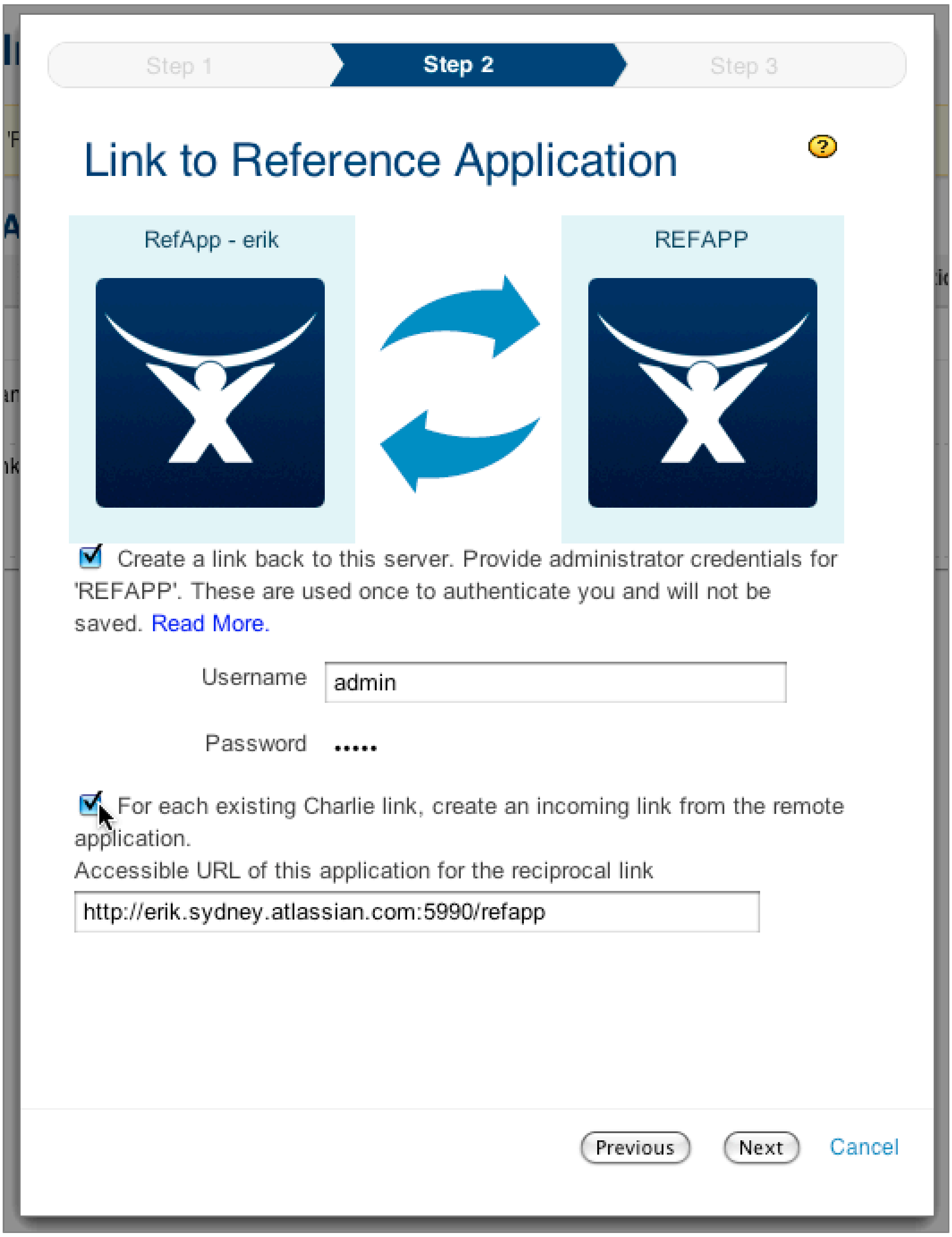


Application 'REFAPP' been upgraded. Click here to Upgrade the Application Link configuration

Configure Application Links

| | Name | Application | Application URL | Configured Authentication | Actions |
|---|---|---|---|---|---|
| | REFAPP | Reference Application | http://localhost:5992/refapp | Basic Access | Configure \| Delete |

*Screenshot above: Upgrading an application link for remote application*



Link to Reference Application

☑ Create a link back to this server. Provide administrator credentials for 'REFAPP'. These are used once to authenticate you and will not be saved. Read More.

Username: admin

Password: •••••

☑ For each existing Charlie link, create an incoming link from the remote application.

Accessible URL of this application for the reciprocal link: http://erik.sydney.atlassian.com:5990/refapp

*Screenshot above: Upgrading an application link wizard*

### Notes

***Related Topics***

Adding an Application Link
Configuring Authentication for an Application Link

# Administering Entity Links for an Application Link

Let's assume that you are managing a project or team. You would like to connect your project's Confluence space with your JIRA project, and link up your team's source repository too.

When you have connected your applications via Application Links, you can also connect the areas of those applications that contain information relating to your project or team. Using **project links** (also called **entity links**) you can associate one or more projects, spaces and repositories across the linked applications.

To connect all the information relating to the project or team that you are managing, you can link one or more of the following:

- JIRA projects.
- Confluence spaces.
- FishEye repositories.
- FishEye projects. A FishEye 'project' is the Crucible project if you have installed FishEye and Crucible, otherwise it is the paths associated via the 'FishEye Project Content' function in FishEye.
- Crucible projects.
- Bamboo projects.

## Uses for Project Links (Also Called Entity Links)

The following integration features use project links:

- Activity streams. For example, the project links determine the activity retrieved from JIRA to display in the activity stream of a FishEye repository or a Crucible project.
- The JIRA FishEye plugin. For example:
    - The link between a JIRA project and a FishEye repository determines the repository searched for a particular issue key when displaying the FishEye source tab in JIRA.
    - The link between a JIRA project and a Crucible project determines the Crucible project scanned for review activity when displaying the Crucible reviews tab in JIRA.
    - When you create a defect in Crucible, Crucible will know which JIRA project to put it in.
- Third-party plugins may make use of project links to enrich their functionality too.

## Managing Project Links

- Adding an Entity Link
- Making an Entity Link the Primary Link
- Delete an Entity Link

**RELATED TOPICS**

Adding an Application Link

# Adding an Entity Link

Let's assume that you are managing a project or team. You would like to connect your project's Confluence space with your JIRA project, and link up your team's source repository too.

When you have connected your applications via Application Links, you can also connect the areas of those applications that contain information relating to your project or team. Using **project links** (also called **entity links**) you can associate one or more projects, spaces and repositories across the linked applications.

To connect all the information relating to the project or team that you are managing, you can link one or more of the following:

- JIRA projects.
- Confluence spaces.
- FishEye repositories.
- FishEye projects. A FishEye 'project' is the Crucible project if you have installed FishEye and Crucible, otherwise it is the paths associated via the 'FishEye Project Content' function in FishEye.
- Crucible projects.
- Bamboo projects.

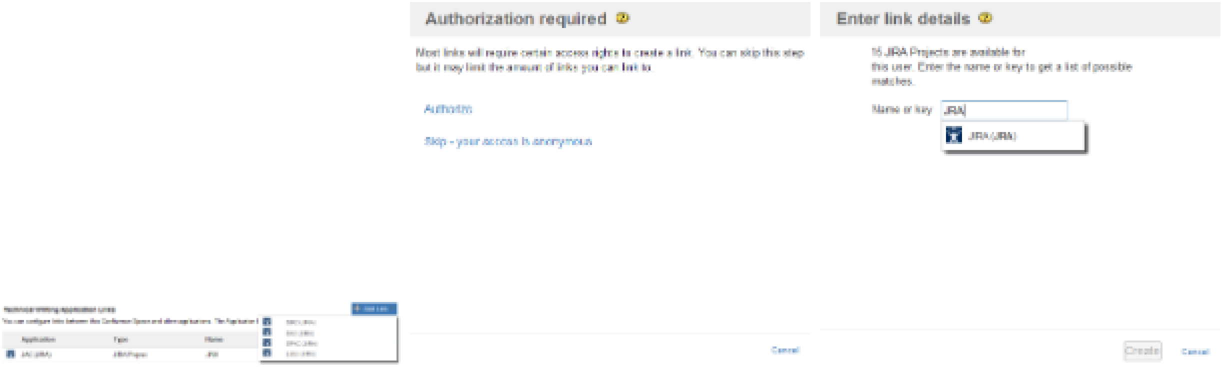
**On this page:**

## Adding a Project Link (Also Called an Entity Link)

**To add a project link:**

1. Log in to your application as an administrator and navigate to the administration page for the project:
    - JIRA project administration
    - Confluence wiki administration
    - FishEye repository administration

        - FishEye/Crucible project administration
        - Bamboo-to-JIRA project links – configured via JIRA project administration
2. Choose the project in your local application that you want to link from. For example, you may want to link a FishEye repository to another type of project.
3. The instructions for adding a project link will vary depending on whether the target application has Application Links:
    - If the target application has Application Links:
        a. Click '**Add Link**'. A dropdown menu will appear listing the applications you have already linked to.
        b. In the dropdown menu, click the application that contains the project you want to link to. For example, if you want to link to a specific JIRA project, click the JIRA site that contains that project. If you want to link to a Confluence space, click the Confluence site that contains that space.
        c. Click one of the options on the 'Authorization required' screen:
            - '**Authorize**' — Click this option if you want to grant your project authorised access to the target project. The target application will open in a new window, so that you can log in and authorise access.
            - '**Skip – your access is anonymous**' — Click this option if you only want to allow anonymous access to the target project.
        d. In the '**Name or Key**' field, enter the name/key of the project in the remote application that you want to link to. For example, if you want to link to a JIRA project, enter the project key. If you want to link to a Confluence space, enter the space key.
        e. Click the '**Create**' button to create the project link.
    - If the target application does not have Application Links:
        a. Click '**Add Link**'. A dropdown menu will display listing the applications you have already linked to.
        b. In the dropdown menu, click the application that contains the project you want to link to. For example, if you want to link to a specific JIRA project, click the JIRA site that contains that project. If you want to link to a Confluence space, click the Confluence site that contains that space.
        c. In the '**Key**' field, enter the name/key of the project in the remote application that you want to link to. For example, if you want to link to a JIRA project, enter the project key. If you want to link to a Confluence space, enter the space key.
        d. *(optional)* Enter the alias for the project in the '**Alias**' field. This is the display name for the project in your administration console.
        e. Click the '**Create**' button to create the project link.



Most links will require certain access rights to create a link. You can skip this step but it may limit the amount of links you can link to.

15 JIRA Projects are available for this user. Enter the name or key to get a list of possible matches.

Step 1
Step 2
Step 3

*Screenshots above: Linking to a JIRA project (where target JIRA server includes Application Links)*

***Related Topics***

Making an Entity Link the Primary Link
Delete an Entity Link

# Making an Entity Link the Primary Link

If you have set up **project links** (also called **entity links**) to more than one project in the same application, for example you have linked your Confluence space to two JIRA projects, then one of the project links will be marked as the primary link. All outgoing requests will be directed to the primary link.

For example, if you have a Confluence space that is linked to two JIRA projects, you can nominate the link to one of the JIRA projects as the primary link. Every time Confluence requests JIRA information (for example, in a JIRA issues macro) it will request it from the primary link's JIRA project. Note, both JIRA projects can still request information from the Confluence space (for example, a Confluence page gadget on the dashboards of each JIRA instance).

**On this page:**

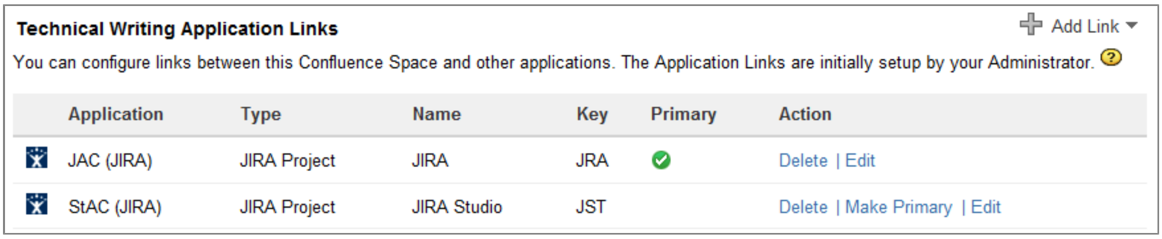
- Making a Project Link the Primary Link
- Notes

## Making a Project Link the Primary Link

**To make a project link the primary link:**

1. Log into your application as an administrator and navigate to the project administration page:
    - JIRA project administration
    - Confluence wiki administration
    - FishEye repository administration
    - FishEye/Crucible project administration
    - Bamboo-to-JIRA project links – configured via JIRA project administration

2. Click the '**Make Primary**' link in the '**Action**' column for the project link that you want to make the primary link. A ✓ symbol will display in the 'Primary' column next to the link.
   *Note:* The 'Primary' column and 'Make Primary' link will appear only if you have set up multiple project links to the same application, for example you have linked a Confluence space to a number of JIRA projects.



**Technical Writing Application Links**

You can configure links between this Confluence Space and other applications. The Application Links are initially setup by your Administrator.

| Application | Type | Name | Key | Primary | Action |
|---|---|---|---|---|---|
| JAC (JIRA) | JIRA Project | JIRA | JRA | ✓ | Delete \| Edit |
| StAC (JIRA) | JIRA Project | JIRA Studio | JST | | Delete \| Make Primary \| Edit |

*Screenshot above: Viewing project links for a Confluence space*

### Notes

#### *Related Topics*

Adding an Entity Link
Delete an Entity Link

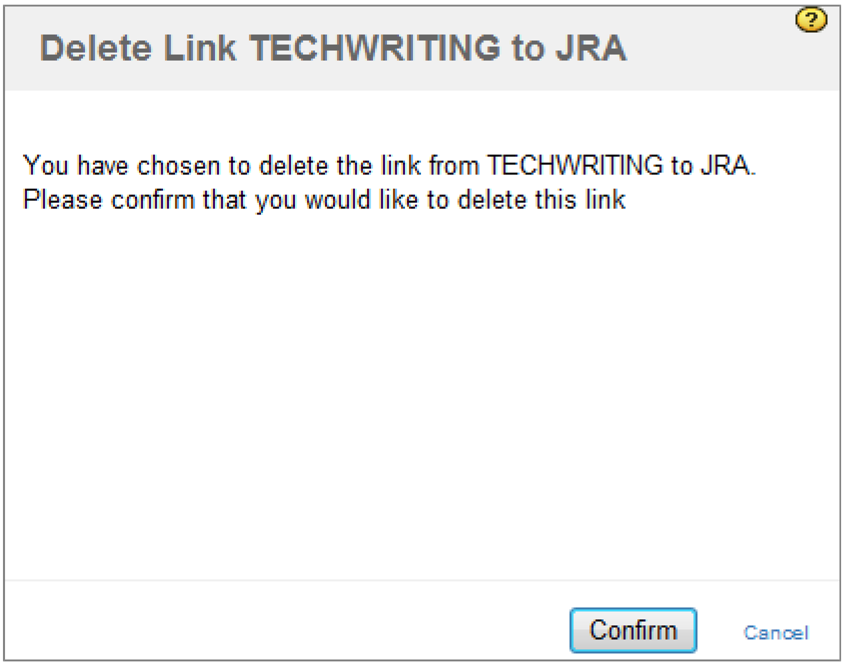
## Delete an Entity Link

Deleting a project link (also called an entity link) stops the projects, spaces and/or repositories from sharing information.

If you have set up multiple project links to the same application, for example you have linked a Confluence space to multiple JIRA projects, deleting the primary link will mean that another of the links will be made the primary link.

**To delete a project link:**

1. Log in to your application as an administrator and navigate to the project administration page:
    - JIRA project administration
    - Confluence wiki administration
    - FishEye repository administration
    - FishEye/Crucible project administration
    - Bamboo-to-JIRA project links – configured via JIRA project administration
2. Click the '**Delete**' link next to the project link that you want to delete. A confirmation screen will be displayed.
3. Click the '**Confirm**' button to delete the project link.



**Delete Link TECHWRITING to JRA**

You have chosen to delete the link from TECHWRITING to JRA. Please confirm that you would like to delete this link

*Screenshot above: Confirming the deletion of a project link*

***Related Topics***

Adding an Entity Link
Making an Entity Link the Primary Link

# Application Links Version Matrix

The matrix below shows the applications that support AppLinks. The applications are listed horizontally across the top and the AppLinks versions are listed vertically on the left.

Notes:

- Application version numbers show the **earliest version** of the application that supports the relevant AppLinks version.
- Version numbers in brackets, such as (3.0.1), show a **future** application release expected to support the relevant AppLinks version.

| | Bamboo | Confluence | Crucible | FishEye | JIRA |
|---|---|---|---|---|---|
| **AppLinks 3.2** | | ✓ Confluence 3.5 | ✓ Crucible 2.4 | ✓ FishEye 2.4 | ✓ JIRA 4.3 |
| **AppLinks 3.3** | | | | | |
| **AppLinks 3.4** | ✓ Bamboo 3.1 | | | | |
| **AppLinks 3.5** | | (Confluence 4.0) | | | (JIRA 4.4) |

ℹ AppLinks is bundled with all application versions shown in the above table. You do not need to install AppLinks into any application.

# Contributing to the Application Links Documentation

Would you like to share your hints, tips and techniques for Application Links? We welcome your contributions. Have you found a mistake in the documentation, or do you have a small addition that would be so easy to add yourself rather than asking us to do it? You can update the documentation page directly.

## Getting Permission to Update the Documentation

Our documentation wiki contains developer-focused documentation (such as API guides, plugin and gadget development guides and guides to other frameworks) as well as product documentation (user's guides, administrator's guides and installation guides). The wiki permissions are different for each type of documentation.

- If you want to update the Developer Network or other developer-focused wiki spaces, just sign up for a wiki username then log in and make the change.
- If you want to update the Application Links product documentation, we ask you to sign the Atlassian Contributor License Agreement (ACLA) before we grant you wiki permissions to update the documentation space. Please read the ACLA to see the terms of the agreement and the documentation it covers. Then sign and submit the agreement as described on the form attached to that page.

## Following our Style Guide

Please read our short guidelines for authors

## How we Manage Community Updates

Here is a quick guide to how we manage community contributions to our documentation and the copyright that applies to the documentation:

- **Monitoring by technical writers.** The Atlassian technical writers monitor the updates to the documentation spaces, using RSS feeds and watching the spaces. If someone makes an update that needs some attention from us, will make the necessary changes.
- **Wiki permissions.** We use wiki permissions to determine who can edit the various types of documentation spaces.
  - Developer documentation (API guides, plugin development and gadget development): Anyone can edit these spaces, provided they have signed up for a wiki username and logged in to the wiki.
  - Product documentation (user's guides, administrator's guides, installation guides): We ask people to sign the Atlassian Contributor License Agreement (ACLA) and submit it to us. That allows us to verify that the applicant is a real person. Then we give them permission to update the documentation.
- **Copyright.** The Atlassian documentation is published under a Creative Commons 'cc-by' license. Specifically, we use a Creative Commons Attribution 2.5 Australia License. This means that anyone can copy, distribute and adapt our documentation provided they acknowledge the source of the documentation. The cc-by license is shown in the footer of every page, so that anyone who contributes to our documentation knows that their contribution falls under the same copyright.

***RELATED TOPICS***

Contributing to the JIRA Documentation
Contributing to the Confluence Documentation
Contributing to the FishEye Documentation
Contributing to the Crucible Documentation
Contributing to the Bamboo Documentation
Contributing to the Crowd Documentation
Author Guidelines
Atlassian Contributor License Agreement