Issue Connectors

- What is an Issue Connector?
- Linking Commit/Checkin Issue Fixes
- Why use an Issue Connector?
- Link an Issue Connector to a Repository
- Creating and Editing Issue Connector
    - Name
    - Template
    - Commit Message Regex
    - Issue Server URL
    - Validation
- Issue Connector Rules and Restrictions
    - All Named Groups in the Commit Message Regex must have at least 1 matching group in the Issue Server URL
    - Both the Commit Message Regex and the Issue Server URL MUST include the id group
    - You can have multiple groups in an Issue Server URL/Commit Message Regex
    - Named Groups can appear multiple times in the Issue Server URL
    - Multiple Issue Ids in a commit message produce multiple Issue links

What is an Issue Connector?

Issue Connectors link issues from commit/checkin messages to your issue tracking system. They work by detecting an issue id in your commit messages and creating a hyperlink within the Continua CI environment to your issue tracking system.

Linking Commit/Checkin Issue Fixes

When Continua gets the latest changesets from your source repository, it automatically retrieves all commit/checkin messages associated with that changeset. If your source repository is linked to an issue connector then it will also try to parse the commit/checkin messages for issue Ids. It will then take those issue Ids from the commit message and generate a hyperlink to that issue in your issue tracking system. The list of hyperlinks will be displayed in the "Issues" tab when viewing a build. Issue Connectors do not interact with your issue tracking system, they just generate hyperlinks to it! It won't open or close issues or require any kind of login to your system.

Why use an Issue Connector?

Lets say you have set up a Repository Trigger so every time a check in is made, the configuration is built. Every time a configuration is built, the build always knows about the latest changeset, or in this case, the changeset that triggered the build. If your company's policy is to have meaningful commit messages that reference your issue tracking system, then an Issue Connector will link Continua to it. With an Issue Connector, you can give it a regular expression to retrieve the issue Id and other information from the commit/checkin message and use this information to create hyperlinks to a specific issue.

Example:

Set up an Issue Connector like so:

Issue Sever URL: http://www.issues.com/issue/{id}

Commit Message Regex: #(?<id>\d+)

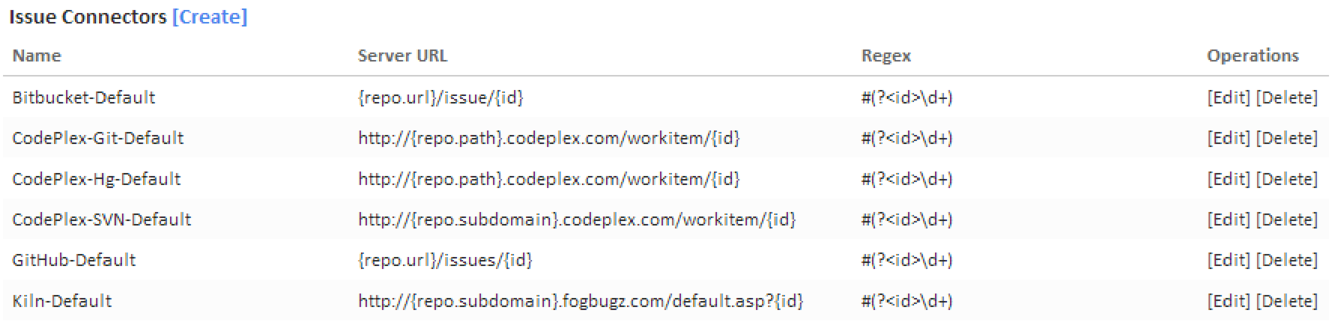
Assume a user commits something to the repository with the message "Fixed annoying deadlock bug, solves issue #29981". The Commit Message Regex in this example extracts any number that is preceded by the # (Hash) symbol. The Issue Connector would extract the value "29981" out of the commit message and use the Issue Server URL to generate the link "http://www.issues.com/issue/29981".

Link an Issue Connector to a Repository

Once an Issue Connector has been created, it needs to be linked to a repository. An Issue Connector can be linked by editing a repository and selecting the correct Issue Connector in the Issue Connector dropdown. Once the repository has been saved, the Issue Connector will monitor any repository commits.

Creating and Editing Issue Connector

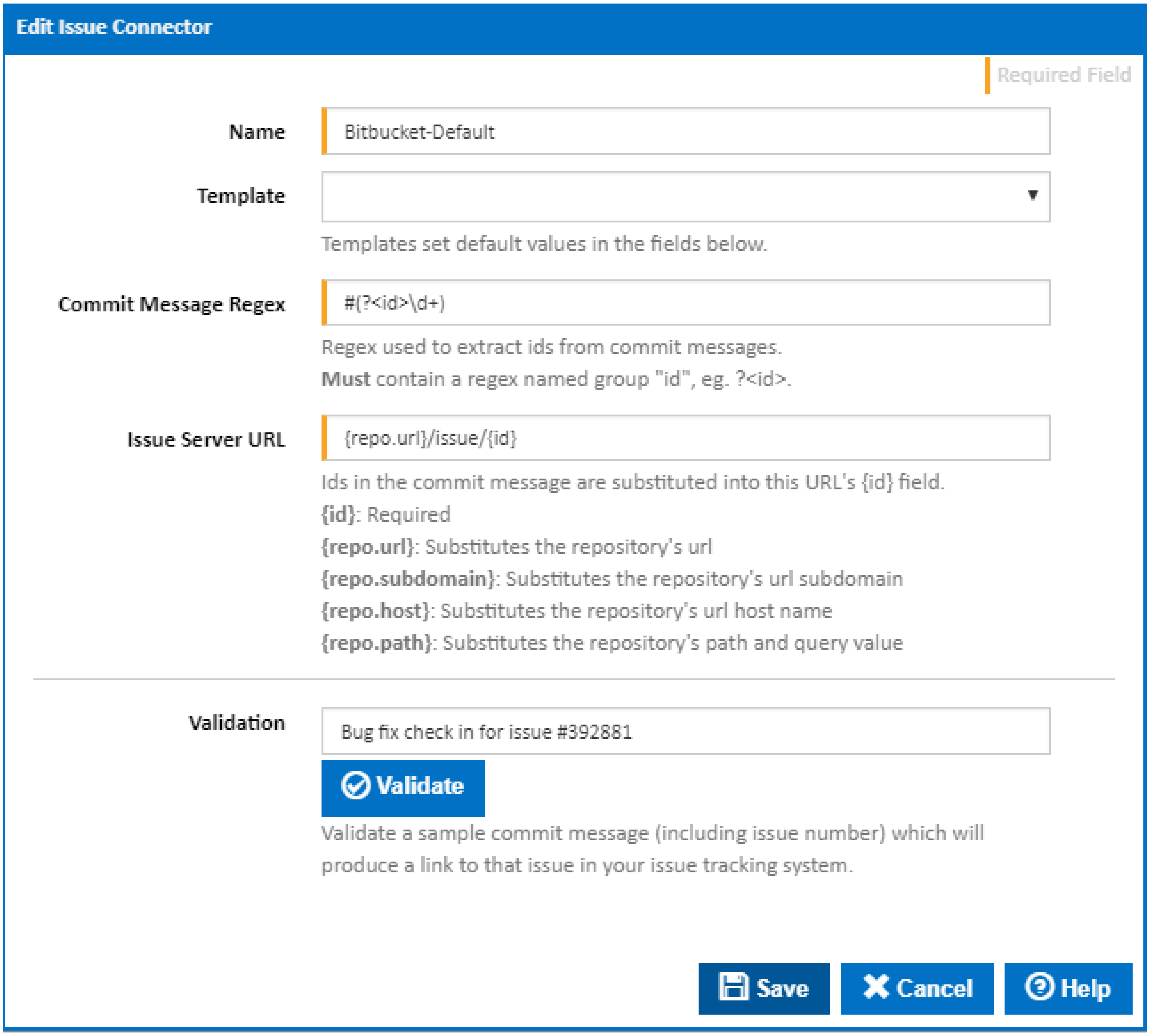
Issue Connectors can be created and edited in the **Administration area**, under the **Continuous Integration section**. On the Issue Connectors page you should see several default issue connectors that have already been created. These issue connectors link to some of the major issue tracking / repository hosting systems and they should work with most of these systems out of the box.



Issue Connectors [Create]

Name	Server URL	Regex	Operations
Bitbucket-Default	{repo.url}/issue/{id}	#(?<id>\d+)	[Edit] [Delete]
CodePlex-Git-Default	http://{repo.path}.codeplex.com/workitem/{id}	#(?<id>\d+)	[Edit] [Delete]
CodePlex-Hg-Default	http://{repo.path}.codeplex.com/workitem/{id}	#(?<id>\d+)	[Edit] [Delete]
CodePlex-SVN-Default	http://{repo.subdomain}.codeplex.com/workitem/{id}	#(?<id>\d+)	[Edit] [Delete]
GitHub-Default	{repo.url}/issues/{id}	#(?<id>\d+)	[Edit] [Delete]
Kiln-Default	http://{repo.subdomain}.fogbugz.com/default.asp?{id}	#(?<id>\d+)	[Edit] [Delete]

New Issue Connectors can be created through the **[Create]** link at the top of the page, or existing issue connectors can be edited through the **[Edit]** links. Both these links will bring up the Issue Connector dialog:



Edit Issue Connector

Required Field

Name: Bitbucket-Default

Template:

Templates set default values in the fields below.

Commit Message Regex: #(?<id>\d+)

Regex used to extract ids from commit messages.
Must contain a regex named group "id", eg. ?<id>.

Issue Server URL: {repo.url}/issue/{id}

Ids in the commit message are substituted into this URL's {id} field.
{id}: Required
{repo.url}: Substitutes the repository's url
{repo.subdomain}: Substitutes the repository's url subdomain
{repo.host}: Substitutes the repository's url host name
{repo.path}: Substitutes the repository's path and query value

Validation: Bug fix check in for issue #392881

Validate a sample commit message (including issue number) which will produce a link to that issue in your issue tracking system.

Name

The Name of the issue connector is used to reference specific issue connectors when adding an issue connector to a repository.

Template

By selecting a template you are pre-populating the remaining fields with appropriate values for a certain issue tracking system. Eg. If you select the GitHub template, it will set Commit Message Regex to **#(?<id>\d+)**, Issue Server URL to **{repo.url}/issues/{id}** and Validation to **Bug fix check in for issue #392881**.

Note that changing the template will overwrite any values that you have already entered.

Commit Message Regex

The Commit Message Regex is a regular expression that tells the issue connector to extract certain data out of commit messages. Any information that should be extracted MUST be placed into a named group. Named groups have the following syntax **<namedGroup>**. For example, to extract an issue Id, you could use the following regex: **#(?<id>\d+)**. This regex gets any number that is preceded by a # (hash) symbol and saves the value into the **<id>** named group.

Note that every commit message regex **MUST contain the <id> named group**.

Named groups can then be inserted into the Issue Server URL using the following syntax: **{ namedGroupName }**. For example, to insert the <id> named group into the Issue Server URL, you would include **{id}** in the Issue Server URL.

Issue Server URL

The Issue Server URL is a template URL for generating links to your issue system. It allows you to dynamically set the value of the URL based on specific build information. Any information that is pulled out of a commit message using **named groups** (read above) can be inserted into the URL. All of the default issue connectors simply pull out the issue Id by using the **<id>** named group in the Commit Message Regex and insert the Id field into the generated issue url by using the **{id}** syntax (See the example provided in the 'Why Use an Issue Connector?' section).

Additional information can be included within the Issue Server URL. For example, if you use the project name in your commit messages, you can set your Commit Message Regex to also extract the project name using the **<project>** named group. This project name can then be inserted into the Issue URL with **{project}** syntax.

Issue Connectors also allow for certain repository information to be included within the Issue Server URL. The following tags can be used:

Tag	Description	Url and Value	Example
{repo.url}	Extracts the source url from the repository and places it in the Issue Server URL	Repository Source URL: https://github.com/username/testGitHub.git {repo.url}: https://github.com/username/testGitHub	Issue Server URL: {repo.url}/issues/{id} Generated Link: https://github.com/username/testGitHub/issues/1
{repo.subdomain}	Extracts the subdomain from the repository source url and place it in the Issue Server URL	Repository Source URL: https://myrepo.kilnhg.com/Code/Repositories/Group/Test1 {repo.subdomain}: myrepo	Issue Server URL: http://{repo.subdomain}.fogbugz.com/default.asp?{id} Generated Link: http://myrepo.fogbugz.com/default.asp?1
{repo.host}	Extracts the host from the repository source url	Repository Source URL: https://myrepo.kilnhg.com/Code/Repositories/Group/Test1 {repo.host}: myrepo.kilnhg.com	Issue Server URL: http://{repo.host}/default.asp?{id} Generated Link: http://myrepo.kilnhg.com/default.asp?1
{repo.path}	Extracts the path and query from the repository source url	Repository Source URL: https://git01.codeplex.com/mygitrepo {repo.path}: mygitrepo	Issue Server URL: http://{repo.path}.codeplex.com/workitem/{id} Generated Link: https://mygitrepo.codeplex.com/workitem/1

Validation

Once you have given the Commit Message Regex and the Issue Server URL a value, you can validate your settings by providing a typical commit message you would use when checking in a fix for a bug. If the validation was successful, a link will be generated that points to your issue tracker.

Note that if any of the tags in the table above have been used in the Issue Server URL then the validation link will not point to the right location. The reason for this is that these repository tags need a repository to work. On validation, these tags will be generate dummy information such as www.yourepositoryurl.com.

Issue Connector Rules and Restrictions

All Named Groups in the Commit Message Regex must have at least 1 matching group in the Issue Server URL

Any named group that is used in the Commit Message Regex MUST also be present in the Issue Server URL. For example, if your regex has the named group <project>, the {project} **must** be present in the Issue Server URL. For example:

Commit Message: "Fixed bug for issue #myawesomeproject-29981"

Issue Sever URL: http://www.issues.com/***{project}***/issue/*{id}*

Commit Message Regex: #(***?<project>***\w+)-(*?<id>*\d+)

Generated URL: http://www.issues.com/myawesomeproject/issue/29981

Both the Commit Message Regex and the Issue Server URL MUST include the id group

For example:

Issue Sever URL: http://www.issues.com/issue/***{id}***

Commit Message Regex: #(***?<id>***\d+)

You can have multiple groups in an Issue Server URL/Commit Message Regex

If you have an issue tracking system that takes in a project as well as an id, then this can be handled by using multiple named groups. For example:

Commit Message: "Fixed bug for issue #myawesomeproject-29981"

Issue Sever URL: http://www.issues.com/***{project}***/issue/***{id}***

Commit Message Regex: #(***?<project>***\w+)-(***?<id>***\d+)

Generated URL: http://www.issues.com/myawesomeproject/issue/29981

Named Groups can appear multiple times in the Issue Server URL

Named Groups can only appear in the Commit Message Regex once, however their corresponding group can appear in the Issue Server URL multiple times. For example:

Commit Message: "Fixed bug for issue #myawesomeproject-29981"

Issue Sever URL: http://***{project}***.issues.com/***{project}***/issue/***{id}***

Commit Message Regex: #(***?<project>***\w+)-(***?<id>***\d+)

Generated URL: http://myawesomeproject.issues.com/myawesomeproject/issue/29981

Multiple Issue Ids in a commit message produce multiple Issue links

Using our first example's Issue Connector settings, a commit message "Fixed annoying deadlock bug, solves issues #29981, #44412 and #9881" would generate three links:

http://www.issues.com/issue/29981
http://www.issues.com/issue/44412
http://www.issues.com/issue/9881