

1 Overview

Throughout this document, the term “web application” is used. The web application can be any forum, CMS, blogging software, etc. that uses MySQL for its database.

These are the major features of Community Bridge:

1. Minecraft player to web application user “linking”.
2. Group synchronization (currently a misnomer, since the “synchronization” is only one way)
3. Recording statistical information about the players in web application custom fields.
4. Kicking of players who are banned from the web application.

The first of the features is essential, without it, none of the other features of the plugin would work. Given that the linking feature’s importance, it is the first thing to evaluate when ensuring that CommunityBridge is operating correctly.

1.1 Proposed config.yml Preamble

```
#
# Community Bridge Configuration File
#

# Log level controls the degree of detail that is sent to the console/log
# The possible settings are (in order of quietest to noisiest:
#   info, config, fine, finer, finest, all
# - During configuration, I recommend using 'config'.
# - During normal operation, I recommend using 'info'.
# - If you want to see the notifications that the synchronize and reminder
#   notices have been sent, use 'fine'
# - During troubleshooting of problems, set this to either finest or all.
log-level: config

# The permissions system (PEX, bPermissions, GroupManager, PermsBukkit)
permissions-system: GroupManager

# Set the unit of measurement for sync and reminder scheduling. Options are:
# ticks, seconds, minutes, or hours. Note that this applies to both
# auto-sync-every and auto-remind-every.
auto-every-unit: minutes

# This is a timer that will check and sync all data with your database at a set
# interval. Otherwise group and player stats will only be synchronized when
# they join and leave the server.
auto-sync: true

# The interval for the auto-sync timer, this should be no less than
# 15-20 minutes.
auto-sync-every: 20

# Some definitions:
# - web-application: The forum, CMS, blogging app, etc. that you're using on
#                   your website.
```

1.2 Proposed config.yml Database Section

```
# Please ensure if you run your Minecraft server on a remote host
# that your MySQL server allows REMOTE connections. This WILL NOT WORK unless
# your MySQL server is configured to allow connections from the machine that
# your Minecraft server is running on. If you don't know what that means, consult
# your hosting provider on how to configure your MySQL server correctly.
# All of this information should be the same server and database used by your
# web-application...
database:
  # Hostname for your MySQL Server
  hostname: localhost

  # MySQL Port
  port: 3306

  # Database Name
  name: databasename

  # Database Username
  username: username

  # Database Password
  password: password
```

2 Linking Feature

For each of the following configurations, CB should:

- Recognize a player as not being registered, e.g., not present in the web application's database.
- Correctly identify a player as being registered—identifying the correct user.
- (NYI) If authentication is off **DO NOT** force authentication.
- (NYI) If authentication is on, handle a valid authentication by...
- (NYI) If authentication is on, respond to an failed authentication attempt by...

Possible Configurations (order of complexity):

- Minecraft playername and web application username are required to be the same. Required information: users table, user id column, username column. Configuration Name: same_name.
- Minecraft playername and web application username can be different; the Minecraft playername is in a separate column on the users table. Required information: users table, user id column, minecraft playername column. Configuration Name: diff_name_same_table.
- Minecraft playername and web application username can be different; the Minecraft playername is on a different table in its own column (frequently, this is because it is implemented through a custom profile field in the web application). Required information: tablename, user id column, minecraft playername column. Configuration Name: diff_name_diff_table
- Minecraft playername and web application username can be different; the Minecraft playername is on a different table that data is stored in the key-value format. Required information: tablename, user id column, key column, value column, data key name. Configuration Name: diff_name_diff_table_keyed

2.1 Proposed config.yml Section

```
# Settings associated with linking a minecraft player with a web application's
# user. As this feature is a prerequisite for all other features, it cannot
# be disabled.
player-user-linking:
  # Set this to true if you want to require the player to confirm that they're
  # the player that is registered on the web application by providing their
  # web application password.
  authentication-required: false

  # Set these to true to inform players when they log in if they're linked to
  # the web application. These correspond to the link-unregistered-player and
  # link-registered-player messages in messages.yml.
  notify-unregistered-player: true
  notify-registered-player: true

  # This is a timer that will notify unregistered users to register
  auto-remind: true

  # The interval for the auto-remind timer, this should be no less than
  # 5-10 minutes. Note that this uses the units specified above in
  # auto-every-unit.
  auto-remind-every: 10

  # If you want the player disconnected from the game if they haven't
  # registered set this to true.
  kick-unregistered: false

  # This needs to be set to either same-name, same-table, multi-table, or
  # multi-table-with-key.
  # same-name: Use this option when you want the minecraft playername and the
  # web application username to be the same.
  # The remaining three options are for when you want the minecraft playername
  # and web application username to be different.
  # same-table: Use this option when the player's minecraft name on the same
  #               table as the users table in its own column.
  # multi-table: Use this option when the player's minecraft name is on a table
  #               separate from the users table in its own column.
  # multi-table-with-key: Use this option when the player's minecraft name is
  #                       on a table separate from the users table that stores
  #                       data in key-value pairs.

  linking-mode: same-name
  # Configuration for the table that the web application stores user information
  # on.
```

```
users-table:
  # The name of the table that the web application stores users on.
  name: users_table
  user-id-column: user_id
  username-column: username
# Configuration for the table that the minecraft player name is stored on by
# the web application.
playername-table:
  # Name of the table that contains the minecraft player name.
  name: users_table
  # Column containing the user's user id on the table containing the
  # minecraft player name.
  user-id-column: user_id
  playername-column: minecraft_playername
  # These three are only used if the player's minecraft name is on a
  # table that stores its data in a key-value pair.
  playername-key: ''
  key-column: ''
  value-column: ''
```

2.2 Proposed messages.yml section

Messages related to the linking feature in the messages.yml.

```
# Replace APPURL with your web application's website url.
link-unregistered-player: Unregistered Account - Please register at APPURL for full access
link-registered-player: Registered Account, Linked to Forums.
link-unregistered-reminder: Just a reminder to visit APPURL and register today!
```

3 Group Synchronization

Many web systems have a notion of a “primary” group. However, very few of the permissions plugins do. This is not a problem when the synchronization is “Forums override Minecraft”, but it presents a difficulty the other direction, “Minecraft override Forums”.

3.1 Proposed config.yml Section

```
# Settings to control group synchronization features.
group-synchronization:
  # Set this to true to use any of the group synchronization features. If this
  # is false, the rest of this section is ignored.
  enabled: false

  # This subsection contains settings for primary group synchronization.
  primary:
    enabled: false

    # Notify player that they've been placed in a primary group.
    notify-player: false

    # Set this to true if your permissions system uses primary groups.
    permissions-system-uses-primary-group: false

    # This subsection contains details about how the web-application stores
    # its primary group information.
    group-table:
      # Set this to true if your web application uses primary groups or its
      # equivalent.
      enabled: false

      # Set this to the name of the table that the users' primary group is
      # stored on.
      name: primary_group_table

      # On the table where primary group information is stored, the user_id
      # column/field
      user-id-column: user_id_on_primary_group_table

      # On the table where primary group information is stored, the primary
      # group column/field
      group-column: group_id_on_primary_group_table
    secondary:
      enabled: false

    # This subsection contains details about how the web-application stores
    # secondary groups.
    group-table:
      # Set this to true if your web application uses secondary groups or its
      # equivalent.
      enabled: false
```



```
# Set this to the name of the table that the users' secondary group is
# stored on.
name: secondary_group_table

# On the table where primary group information is stored, the user_id
# column/field
user-id-column: user_id_on_secondary_group_table

# On the table where primary group information is stored, the primary
# group column/field
group-column: group_id_on_secondary_group_table
```

3.2 Proposed messages.yml Section

group-synchronization-primary-notify-player: You have been placed in the primary group '

A Full Proposed config.yml

```
#
# Community Bridge Configuration File
#

# Log level controls the degree of detail that is sent to the console/log
# The possible settings are (in order of quietest to noisiest:
#   info, config, fine, finer, finest, all
# - During configuration, I recommend using 'config'.
# - During normal operation, I recommend using 'info'.
# - If you want to see the notifications that the synchronize and reminder
#   notices have been sent, use 'fine'
# - During troubleshooting of problems, set this to either finest or all.
log-level: config

# The permissions system (PEX, bPerms, GroupManager, PermsBukkit)
permissions-system: GroupManager

# Set the unit of measurement for sync and reminder scheduling. Options are:
# ticks, seconds, minutes, or hours. Note that this applies to both
# auto-sync-every and auto-remind-every.
auto-every-unit: minutes

# This is a timer that will check and sync all data with your database at a set
# interval. Otherwise group and player stats will only be synchronized when
# they join and leave the server.
auto-sync: true

# The interval for the auto-sync timer, this should be no less than
# 15-20 minutes.
auto-sync-every: 20

# Some definitions:
# - web-application: The forum, CMS, blogging app, etc. that you're using on
#                   your website.

# Please ensure if you run your Minecraft server on a remote host
# that your MySQL server allows REMOTE connections. This WILL NOT WORK unless
# your MySQL server is configured to allow connections from the machine that
# your Minecraft server is running on. If you don't know what that means, consult
# your hosting provider on how to configure your MySQL server correctly.
# All of this information should be the same server and database used by your
# web-application...
database:
```

```
# Hostname for your MySQL Server
hostname: localhost

# MySQL Port
port: 3306

# Database Name
name: databasename

# Database Username
username: username

# Database Password
password: password

# Settings associated with linking a minecraft player with a web application's
# user. As this feature is a prerequisite for all other features, it cannot
# be disabled.
player-user-linking:
  # Set this to true if you want to require the player to confirm that they're
  # the player that is registered on the web application by providing their
  # web application password.
  authentication-required: false

  # Set these to true to inform players when they log in if they're linked to
  # the web application. These correspond to the link-unregistered-player and
  # link-registered-player messages in messages.yml.
  notify-unregistered-player: true
  notify-registered-player: true

  # This is a timer that will notify unregistered users to register
  auto-remind: true

  # The interval for the auto-remind timer, this should be no less than
  # 5-10 minutes. Note that this uses the units specified above in
  # auto-every-unit.
  auto-remind-every: 10

  # If you want the player disconnected from the game if they haven't
  # registered set this to true.
  kick-unregistered: false

  # This needs to be set to either same-name, same-table, multi-table, or
  # multi-table-with-key.
  # same-name: Use this option when you want the minecraft playername and the
```

```
# web application username to be the same.
# The remaining three options are for when you want the minecraft playername
# and web application username to be different.
# same-table: Use this option when the player's minecraft name on the same
#               table as the users table in its own column.
# multi-table: Use this option when the player's minecraft name is on a table
#               separate from the users table in its own column.
# multi-table-with-key: Use this option when the player's minecraft name is
#                       on a table separate from the users table that stores
#                       data in key-value pairs.

linking-mode: same-name
# Configuration for the table that the web application stores user information
# on.
users-table:
  # The name of the table that the web application stores users on.
  name: users_table
  user-id-column: user_id
  username-column: username
# Configuration for the table that the minecraft player name is stored on by
# the web application.
playername-table:
  # Name of the table that contains the minecraft player name.
  name: users_table
  # Column containing the user's user id on the table containing the
  # minecraft player name.
  user-id-column: user_id
  playername-column: minecraft_playername
  # These three are only used if the player's minecraft name is on a
  # table that stores its data in a key-value pair.
  playername-key: ''
  key-column: ''
  value-column: ''

# Settings to control group synchronization features.
group-synchronization:
  # Set this to true to use any of the group synchronization features. If this
  # is false, the rest of this section is ignored.
  enabled: false

# This subsection contains settings for primary group synchronization.
primary:
  enabled: false

  # Notify player that they've been placed in a primary group.
  notify-player: false
```

```
# Set this to true if your permissions system uses primary groups.
permissions-system-uses-primary-group: false

# This subsection contains details about how the web-application stores
# its primary group information.
group-table:
  # Set this to true if your web application uses primary groups or its
  # equivalent.
  enabled: false

  # Set this to the name of the table that the users' primary group is
  # stored on.
  name: primary_group_table

  # On the table where primary group information is stored, the user_id
  # column/field
  user-id-column: user_id_on_primary_group_table

  # On the table where primary group information is stored, the primary
  # group column/field
  group-column: group_id_on_primary_group_table
secondary:
  enabled: false

# This subsection contains details about how the web-application stores
# secondary groups.
group-table:
  # Set this to true if your web application uses secondary groups or its
  # equivalent.
  enabled: false

  # Set this to the name of the table that the users' secondary group is
  # stored on.
  name: secondary_group_table

  # On the table where primary group information is stored, the user_id
  # column/field
  user-id-column: user_id_on_secondary_group_table

  # On the table where primary group information is stored, the primary
  # group column/field
  group-column: group_id_on_secondary_group_table
```

B Full Proposed messages.yml

```
# Replace APPURL with your web application's website url.  
link-unregistered-player: Unregistered Account - Please register at APPURL for full access  
link-registered-player: Registered Account, Linked to Forums.  
link-unregistered-reminder: Just a reminder to visit APPURL and register today!  
  
group-synchronization-primary-notify-player: You have been placed in the primary group '
```