Mail System

Documentation

Table of content

Table of content 2
Mail System 3
Requirements 3
Features 3
MySql 4
Setup 4
Client 5
Setup 5
Interface customization 6
Server 7
Setup 7
Scripting Client Side 8
Connect 8
Setup 8
SendMail 9
UName 9
Enable 10
Con 10
Scripting Server Side 12
StartServer 12
GetDate 12
SendAllDataOnConnected 12
GetUserNetPlayer 13
AddUser 13
GetInbox, GetDraft, GetSent 13
GetPlayerID 13

Mail System

A multi platform (Mac/Windows/Web Player/Android/iOS) mail system solution for Unity. It allows Unity developers to implement mail system to their applications (games, simulations, etc.). System is using Unity3D built-in networking system.

Requirements

MySQL database.

Features

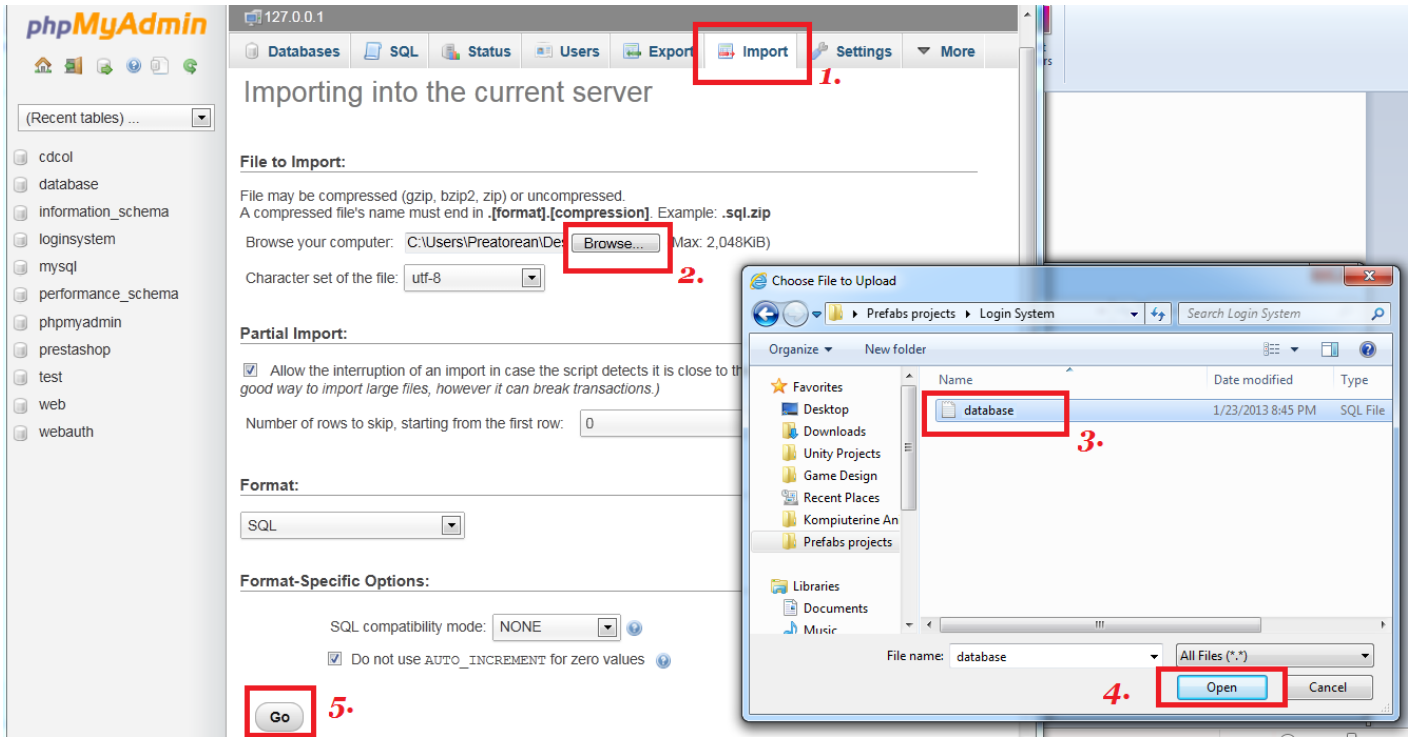
- The mail system is fully integrated in the Unity editor (Mac and Win)
- Supports any Mac and Windows standalone applications. Plius Web Player, Android, iOS.
- All code is commented as much as possible, to be easy understandable for user.
- Easy customization of mail interface using built-in unity GUI system.
- Database optimized for best performance and storage saving.
- All information is saved to database.
- Ability to send message to multiple users.
- When you send message to the user, your message is saved in sent table and you can read it later if you need.
- After reading new message, it's status in database changes to checked.
- Can get all information at once on login or each specific data by request.
- Demonstration scenes included.

NOTICE: Server and client packages can't be imported to the same project. You must to import to the separate projects (Found in Packages for importing folder). As client and server packages contains scripts with the same name (Sys, INIT)!

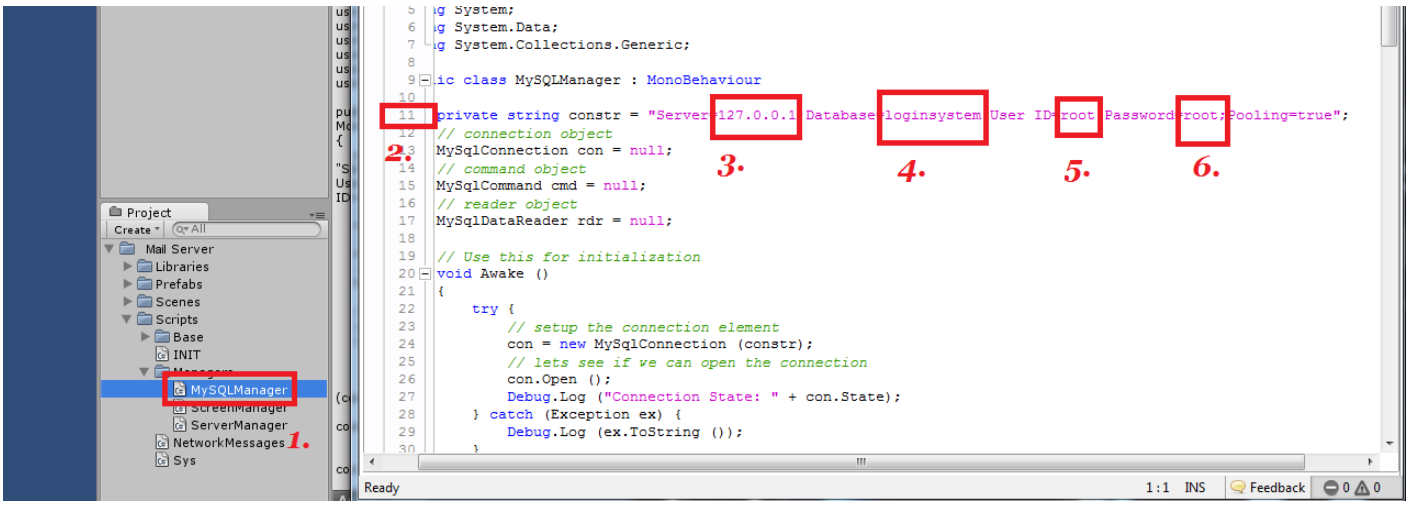
DISCLAIMER: Use this package at your own risk. We have tested the package sufficiently in our development environments but we cannot be held responsible for your usage of this package.

MySql

Setup



1. Login into your database using MyPHPAdmin or other tool and go to import tab.
2. Press Browse…
3. Select database.sql file found in the Mail Server folder and (4.) press open.
5. Press go and you are done with database setup for the login system.
6. Open MySQLManager script found in "MailServer/Scripts/Managers/". In the script find line 11. There is number 3 write your database address, leave default if MySQL server is on same machine. 4. Database name, leave default if you haven't changed it. 5. and 6. Your database user name and password.

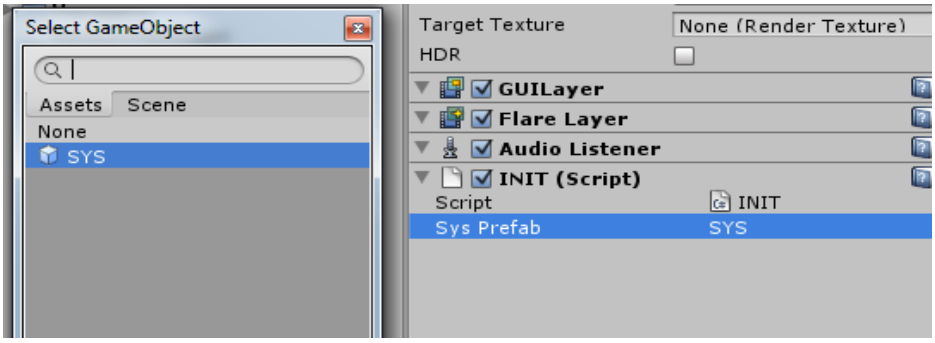


Client

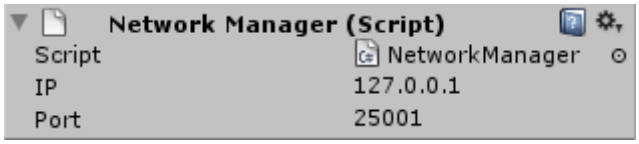
Setup

1. Import “MailClient” package to your client project.
2. Attach "INIT" script to any game object from Scripts folder.
3. Select SYS prefab.

Notice: It’s necessary to do these steps otherwise the system won’t work as Sys prefab Network View ID must be 0!

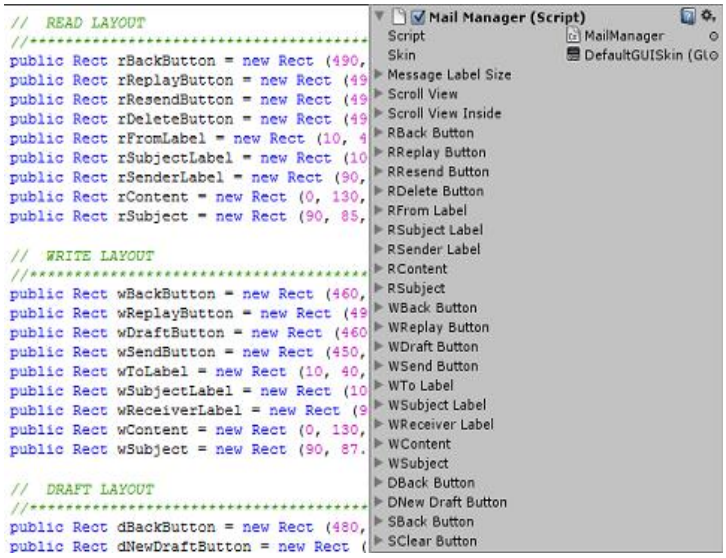


4. Select Sys prefab in project view under prefabs folder. Find attached Network Manager Script, there change default IP address and port in the inspector to match with the server’s IP address. Leave default if server is on same machine for testing purpose.



Interface customization

Can be done in two ways.
Script and in inspector (SYS prefab and GUI skin).



First letter identifies to which window layout rectangle belongs.
R - Read window layout, W - Write layout, D - Draft layout, S - Sent layout.

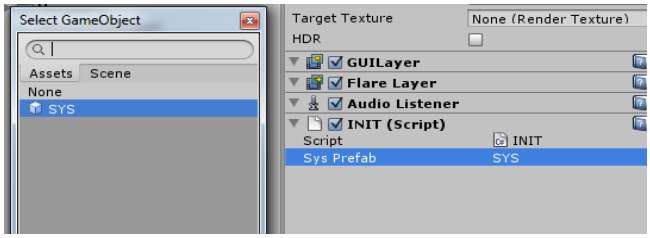
Notice: if you change values in script don't forget to reset the script.

Server

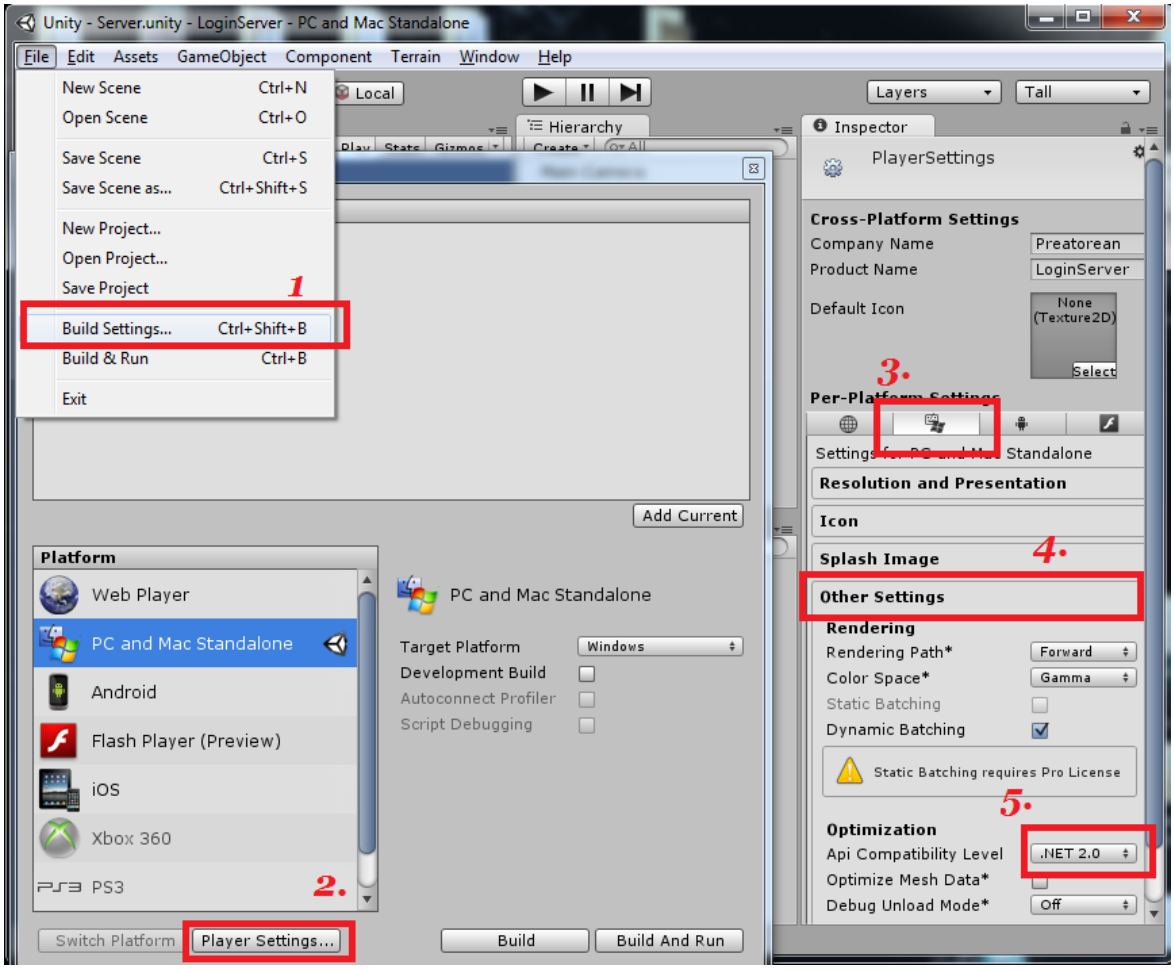
Setup

1. Import “MailServer” package to your server project.
2. Attach "INIT" script to any game object from Scripts folder.
3. Select SYS prefab.

Notice: It’s necessary to do these steps otherwise the system won’t work as Sys prefab Network View ID must be 0!



4. There is 5. Select .NET 2.0. Otherwise you won’t be able to build your server project.



Scripting Client Side

Connect

Connect) : void

Description

Connect to the mail server.

- To access write: Sys.Network.Connect ();

Example:

```
using UnityEngine;
using System.Collections;
public class Demo : MonoBehaviour {


	void Start () {
		Sys.Network.Connect ();
	}
}
```

This script send request to connect to the mail server.

Setup

Setup (ip : string, port : int) : void

Description

Sends call to the server for new user registration.

To access write: Sys.Network.Setup (yourIP : string, yourPort : int);

Example:

```
using UnityEngine;
using System.Collections;
public class Demo : MonoBehaviour {


	void Start () {
```

```
                Sys.Network.Setup ("127.0.0.1",25001);
        }
}
```

This script setups ip and port.

SendMail

SendMail (Receiver : string, Subject : string, Message : string) : void

Description

Sends message.

To access write: Sys.Network.SendMail (Receiver : string, Subject : string, Message : string);

Example:

```
using UnityEngine;
using System.Collections;
public class Demo : MonoBehaviour {

        void Start () {

        Sys.Network.SendMail ("Jonas", "Hello","Hi, join us at www.000.com");

        }
}
```

UName

Sys.Mail.UName : string

Description

Set or get name, it will be used for user name in the server.

To access write: Sys.Mail.UName = yourName : string;

Example:

```
using UnityEngine;
```

```
using System.Collections;
public class Demo : MonoBehaviour {
	// Update is called once per frame
	void OnGUI () {
	Sys.Mail.UName = GUI.TextField (new Rect (Screen.width / 2 - 75, scrH / 2 - 30, 150, 30), Sys.Mail.UName, 25);
	}
}
```

This script displays field to enter the user name.

Enable

Enable : bool

Description

Show or hide mail window.

To access write: Sys.Mail.Enable();

Example:

```
using UnityEngine;
using System.Collections;
public class Demo : MonoBehaviour {
	// Update is called once per frame
	void OnGUI () {
	if (GUI.Button (new Rect (Screen.width / 2 - 100, scrH / 2 + 10, 200, 40), "Show Mail Window")) {
	Sys.Mail.Enable = true;
	}
	}
}
```

Con

Con : bool

Description

Connect to the mail system. Call after user name is entered.

To access write: Sys.Mail.Con();

Example:

```
using UnityEngine;
using System.Collections;
public class Demo : MonoBehaviour {
        // Update is called once per frame
        void OnGUI () {
        if (GUI.Button (new Rect (Screen.width / 2 - 100, scrH / 2 + 10, 200, 40), "Login")) {
        Sys.Mail.Con();
        }
        }
}
```

Scripting Server Side

StartServer

StartServer () : void

Description

Start mail server.

To access write: • Sys.Server.StartServer();

Example:

```
using UnityEngine;
using System.Collections;
public class Demo : MonoBehaviour {

	void OnGUI () {

		if(GUI.Button(new Rect(200,280,150,30),"Start Server")){
		Sys.Server.StartServer();
		}
	}
}
```

This script starts the mail server.

GetDate

GetDate () : string

Description

Use it to get current date (format - MM/dd/yyyy, hh:mm:ss tt):

To access write: Sys.Server.GetDate(); returns [type:string]

SendAllDataOnConnected

SendAllDataOnConnected (name : string) : bool

Description

Use it to enable/disable sending all data at once to user on his connection to the mail system:

To access write: Sys.Server.SendAllDataOnConnected = true/false(type:boolean];

GetUserNetPlayer

GetUserNetPlayer (name : string) : NetworkPlayer

Description

Returns user's unity network player.

AddUser

AddUser (id : int, name : string, info : NetworkMessageInfo) : void

Description

Adds the user to the users list.

To access write: Sys.Server.AddUser (id : int, name : string, info : NetworkMessageInfo);

GetInbox, GetDraft, GetSent

GetInbox (userID : int, info : NetworkMessageInfo) : void

Description

Call to get all messages for specific user.

To access write:

- Sys.Server.GetInbox (userID: int, info : NetworkMessageInfo);
- Sys.Server.GetDraft(userID: int, info : NetworkMessageInfo);
- Sys.Server.GetSent(userID: int, info : NetworkMessageInfo);

GetPlayerID

GetPlayerID (targ : NetworkPlayer) : void

Description

Gets user id.

To access write: Sys.Server. GetPlayerID (targ : NetworkPlayer);