**Multiplayer Assembly Package  
Tutorials  
News**

This tutorial will teach you how to create a GUI in-game that downloads your game news from a web source. I cover my method for both publishing the news via a PHP admin panel and the TS code necessary to download and display this news

**What is it?**At some point, you might want to push a major update or inform players of an upcoming content expansion pack. The only problem is, not all of your users will view your web site or forums for that manner to see this info. The best way to reach this audience, is to directly display it to them the instant they sign on to your game. This tutorial will go over how to display a news gui the instant your players log-in that displays the latest news for your project.

Inside the Tutorial Files folder (which if you didn’t know, contains the code for all of these tutorials), you will see a NewsDlg.gui file. Place this file in your art/gui/ folder. Then open a new TorqueScript file and name it news.cs placing it in the scripts/client/ folder.

//news.cs

//Phantom139

//Download and populate the newsDlg. Also controls the commands

//for this specific GUI control.

//STYLE:

//$MAP$NEWS Title\tContent

function addNewsData(%data) {

%this = newsControl;

// Create the article

%article = new ScriptObject() {

headline = getField(%data, 0);

content = strReplace(getField(%data, 1), "[n]", "\n");

};

// Add the article

%this.article[%this.last] = %article;

%this.last++;

}

function newsDlg::onWake(%this) {

if(!isObject(newsControl)) {

MessagePopup("News", "Please Wait, Connecting to News Server");

schedule(500, 0, downloadNews);

return;

}

//check the news control

%this.populate();

//select the last article

Headlines.setSelectedRow(0);

newsContent.setText(newsControl.article[0].content);

}

function newsDlg::populate(%this) {

if(!isObject(newsControl)) {

return;

}

for(%i = 0; newsControl.article[%i] !$= ""; %i++) {

%headline = newsControl.article[%i].headline;

//%content = newsControl.article[%i].content;

//Add the headline to the text.

Headlines.addRow(%i, %headline);

}

}

function Headlines::onSelect(%id, %text) {

//echo(%id @" "@%text);

newsContent.setText(newsControl.article[%text].content);

}

function downloadNews() {

if(!isObject(newsControl)) {

new ScriptObject(newsControl) {

last = 0;

};

}

else {

newsControl.delete();

new ScriptObject(newsControl) {

last = 0;

};

}

Headlines.clear();

if(isObject(NewsDownloader)) {

NewsDownloader.delete();

}

%theNews = new TCPObject(NewsDownloader);

%separator = getRandomSeperator(16);

%header = assembleHTTP1\_1Header("www.mywebsite.com", "/path/to.news.php", "POST", "MAP Client", "Content-Type: multipart/form-data; boundary="@%separator@"\r\n");

%dispo = makeDisposition(%separator, newsReq, 1, 1);

%header = %header @ "Content-Length: " @ strLen(%dispo) @ "\r\n\r\n";

%theNews.request = %header @ %dispo;

%theNews.connect("www.mywebsite.com:80");

}

function NewsDownloader::onConnected(%this) {

%this.timeout = schedule(7500, 0, TimeoutNewsDownload);

%this.send(%this.request);

}

function NewsDownloader::onLine(%this, %line) {

closeMessagePopup();

if(strstr(%line, "$MAP") != -1) {

%line = strReplace(%line, "$MAP", "");

%response = stripchars(%line, "$\n");

switch$(firstWord(%response)) {

case "INTERNAL":

MessageBoxOK("News", "Could Not Download The News: \n"@%response);

return;

case "NEWS":

//add the news to the news control

%response = strReplace(%response, "NEWS ", "");

addNewsData(%response);

}

}

}

function NewsDownloader::onDisconnect(%this) {

cancel(%this.timeout);

%this.delete();

for(%i = 0; newsControl.article[%i] !$= ""; %i++) {

%headline = newsControl.article[%i].headline;

//%content = newsControl.article[%i].content;

//Add the headline to the text.

Headlines.addRow(%i, %headline);

}

//select the last article

Headlines.setSelectedRow(0);

newsContent.setText(newsControl.article[0].content);

}

function NewsDowloader::onConnectFailed(%this) {

TimeoutNewsDownload();

}

function NewsDowloader::onDNSFailed(%this) {

TimeoutNewsDownload();

}

function TimeoutNewsDownload() {

if(isObject(NewsDownloader)) {

NewsDownloader.disconnect();

}

closeMessagePopup();

MessageBoxOk("Connection Timed Out", "Unable to connect to the news server.");

}

Make all of the adjustments above to reflect your website, and where you will put a news.php file (we’re heading there next). Be sure to execute the news.cs file and the new gui dialog, and then in the location where the client logs in, after a successful login, call: canvas.pushDialog(NewsDlg);

Now, head over to your website, and create a new file named news.php. Make sure the path is the same as the one you had earlier in the TS code:

<?php

// News Data

// Accesses the MySQL Table to send out News to the game clients

require("path/to/sql.php");

function obtainNews() {

$sql = sqlLogin("host\_newsUsr", "sqlPassowrd", "gameNews");

$Highest = 0;

$result = sqlCall($sql, "SELECT \* FROM News");

while($row = $result->fetch\_array(MYSQLI\_ASSOC)) {

$PiD = $row["postIndex"];

$PTitle[$PiD] = $row["postTitle"];

$PBy[$PiD] = $row["postAuthor"];

$PDate[$PiD] = $row["postDate"];

$PCont[$PiD] = $row["postContent"];

if($PiD > $Highest) {

$Highest = $PiD;

}

}

//loop backwards...

for($x = $Highest; $x > 0; $x--) {

echo "$"."MAP"."$"."NEWS ".$PTitle[$x]."\t"."<Font:Arial:20><Just:Center>".$PTitle[$x]."[n]<Color:6B238E><Font:Arial:18>Author: ".$PBy[$x]." | ".$PDate[$x]."[n]<Just:Left><Color:000000><Font:Arial:16>".$PCont[$x]."\n";

}

$sql->close();

}

obtainNews();

?>

Make the necessary MySQL adjustments, and yes. You get to now make a brand spanking new MySQL database and table on your server with the following fields:

Database: gameNews

Table: News

Fields:

postIndex – int(10)

postTitle – text()

postContent – text()

postAuthor – text()

postDate – text()

Save all files. Now to add news, you can use the phpMyAdmin interface to manually enter new fields to the table, or be fancy and write a php “admin panel” to handle that fun for you, I leave that to you.

All in all, this is a quick and easy way to get news to your players when you need to!