Out of the Park Baseball 2006 Customization Guide

Revision History

Date	Author	Version	Change Reference
6-Jun-2006	Steve Battisti	1.0	Finished names and nations sections, Excel and comma-separated instructions
13-Jun-2006	Steve Battisti	1.1	Finished injuries and logos sections
14-Jun-2006	Steve Battisti	1.2	Finished player photos section
19-Oct-2006	Steve Battisti / Shawn Scefalu	1.3	• Finished ballpark editing section • Added a workaround related to injuries • Added section on editing in-game text, but not play-by-play yet.

Table of Contents

1.0 What is This Guide?

One of the hallmarks of the Out of the Park series has been its great customizability, allowing players to modify the game by importing custom roster sets, creating new “skins” for the game, adding custom ballparks, and so on. OOTPB continues in that tradition, but with even greater flexibility.

This is a guide to customizing OOTP Baseball 2006. If you are looking for the Manual, which contains a relatively basic introduction to the game, or the Game Guide, a much longer but more thorough document covering information about the game, you’re in the wrong place! You can find either of those documents here.

So, what does "customizing" mean? For our purposes, we're going to define "customizations" as "changes to OOTPB made *outside of the game*." So, modifying some of the text files that come with the game is a customization. Changing your game's league settings is not. Got it?

OOTPB does include some in-game editor features, such as the ability for a commissioner to edit a player. Editing in this manner is described in the Game Guide, not in this customization guide. In some cases, such as editing ballparks, there is a little bit of both required. In those cases, we'll cover everything here so that you have the whole picture in one place.

Note: This manual assumes you are familiar with the OOTPB interface. At times we will refer you to specific pages within the game, and in most of these cases we will expect you to know how to find them. If you're having trouble, please consult the Game Guide or Manual!

2.0 Important Warning About Customization

You've heard it before, but we'll say it again. Any time you customize, you risk damaging your game. The only way to be really sure you won't ruin anything is to back up, back up, back up. If you're modifying your game's rosters, back up the entire league. If you're modifying a text file, back up that text file, and so on.

Trust us, the few minutes it will take to do this is well worth it. We don't want to hear any stories about people who blew up a league that took them weeks to create, because they messed up a player import.

Just in case that wasn't overbearing enough, we'll mention it a few more times later on as well.

3.0 Nations, Cities, and Player Origins

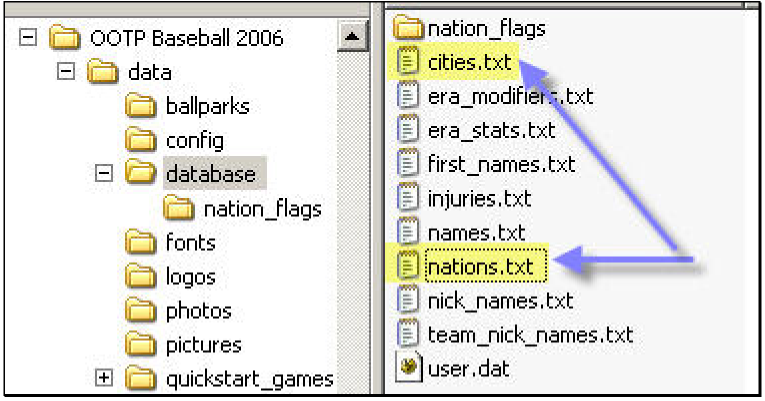
OOTPB comes with a world database that includes over 90,000 cities and all of the nations in the world today. The world database is used in determining player and coach origins, distance between origins and teams for purposes of player contract decisions, and more. But perhaps you don't want to use the real world for your baseball universe. Perhaps you want to add new cities or play in an entirely fictional universe. Luckily, most of the world database is customizable.

See Section 11.0 for more information on editing OOTPB text files using Microsoft Excel.

3.1. Where are the World Database Files?

OOPTB's world database is actually nothing more than two text files: **nations.txt** and **cities.txt**. These are located in the **data\database** directory in your OOTPB installation directory:

Figure 1 World Database File Location



3.2. When are World Database Files Used?

The nations.txt and cities.txt files are used in a number of scenarios in the game:

- In determining the nation for each league in your universe
- In determining team cities based on the league nation
- In determining player and coach origins and names
- In making decisions about whether a player wants to sign a contract, based on the distance between the team's location and the player's origin

3.3. Understanding the Nations.txt File

The nations.txt file is a list of all of the nations used in OOTPB. The version that comes with the game includes every nation in the world today. The nations file is comma-separated, which means it can be edited quite easily. See Section 10.0 for more information about comma-separated files. The nations file stores between 9 and 15 pieces of data for each nation, as follows:

Field #	Req'd	Value	Description	Used in...
1	Y	Nation ID	Unique identifier for the nation. This is a numeric ID, and must be unique. You cannot have two nations with the same Nation ID!	Required, but not visible within the game.
2	Y	Country Long	Long description of the nation	League Nation drop-down
3	Y	Country Short	Short description of the nation	
4	Y	Country Abbreviation	Abbreviation of the nation	Nationality in lists such as the Roster screen.
5	Y	Nationality	Adjective describing people from that country	Nationality select drop-downs Player/coach profile pages
6	N	Region	Region of the world. These are hard-coded in the database, along the lines of the six continent model: 1 = Africa 2 = Asia	Displayed on the Explore World page, but not a factor in any in-game decisions.

			3 = Europe 4 = North America 5 = Oceania 6 = South America The region model is not 100% accurate. Many smaller nations are grouped in with the closest approximate match. *This value can be left blank.*	
7	Y	Baseball Level	Approximate quality of baseball in the nation. There are 5 values: 1 = Nonexistent 2 = Poor 3 = Fair 4 = Good 5 = Excellent	Explore World page. This value is also used in calculating hidden players (when scouting nations) as well as in calculating where foreign players come from, and how talented they will be.
8	Y	Ethnicity/Name Set #1	A number between 0 and 39, representing a name set that should be used when generating names for players from this nation. See Section 4.4 for more information on name sets.	Does not appear in game. Used only during player creation.
9	Y	Ethnicity/Name Set #1 %	The % of players that will be generated using the name set in the previous line. See Section 4.4 for more information on name sets.	Does not appear in game. Used only during player creation.
10-15	N	Ethnicity/Name Sets #2 through #4	Fields 10-15 are three more sets of two fields each, just like fields 8 and 9. Each allows you to specify a name set and a %. This means for each nation, players can be generated from up to 4 name sets/ethnicities. See Section 4.4 for more information on name sets.	Does not appear in game. Used only during player creation.

Some notes about the nations.txt file:

- If you only wish to use one name set, commas are not required after the first name set.
- Spaces are not needed after commas. In fact, it is recommended not to use spaces.
- Values that are marked as "not required" can be left blank. However, you still must include a comma. So, for example, if you want the baseball level to be left blank, the row might look like this:

```
300,My Nation Name,MyNation,MYN,3,,0,100,
```

Note: Nation data is static. That is, baseball quality does not change over time, and so forth.

3.3.1. Example from Nations.txt

Let's look at an example of a line from nations.txt:

```
206,The United States,U.S.A.,USA,American,4,5,0,71,1,27,2,2
```

Here's what this line means in plain English:

- The nation ID is 206
- The long version of the nation's name is "The United States."
- The short version of the nation's name is "U.S.A."
- The abbreviation of the nation is "USA"
- The nationality, or adjective used to describe people from this nation, is "American."
- The nation is on continent 4, North America.
- The baseball level in the U.S.A. is 5, Excellent.
- 71% of players from the U.S.A. will be named using name set 0. These are modern U.S. names, as described in the name set list in Section 4.4.
- 27% of the players from the U.S.A. will be named using name set 1. These are Hispanic names, as described in the name set list in Section 4.4.
- 2% of the players from the U.S.A. will be named using name set 2. These are Japanese names, as described in the name set list in Section 4.4.

Note: Remember, name sets only define the player's name. All of these players will be of U.S. nationality. So, you could get a U.S. player named Julio Lopez.

3.4. Understanding the Cities.txt File

Cities.txt is a list of almost 95,000 real-world cities, including almost every city in the world with a population of greater than 1,000. Just like nations.txt, the cities file is comma-separated. See Section 10.0 for more information about comma-separated files. The cities file stores 6 pieces of data for each city, as follows:

Field #	Req'd	Value	Description	Used in...
1	Y	City Name	Name of the city	Player origins, team locations
2	Y	Population	Population of the city	Determining likelihood of a teams and players being created in that city
3	Y	Nation ID	Numeric ID for the city's nation. This corresponds to the nation ID in the nations.txt file	
4	Y	Region/State	Name of the region or state in which the city is located.	Player origins, team locations
5	Y	City Latitude	The latitude of the city.	Used in contract negotiations, when a player is deciding whether the team offering a contract is close enough to the player's home.
6	Y	City Longitude	The longitude of the city.	Used in contract negotiations, when a player is deciding whether the team offering a contract is close enough to the player's home.

Note: City data is static. That is, populations do not fluctuate over time.

3.4.1. Example from Cities.txt

Let's look at an example of a line from cities.txt:

```
Baltimore,610892,206,Maryland,39.3,-76.61
```

Here's what this line means in plain English:

- The city name is Baltimore.
- The population of the city is 610,892.
- The city is a part of nation 206, which is the U.S.A., according to nations.txt.
- The city is a part of the state Maryland.
- The city is located at longitude 39.3, latitude -76.61

3.5. How do the World Files Work in OOTPB?

Now that you've had a description of how the files are laid out, how exactly do they work in the game? Let's walk through the creation of a league, and see how these files come into play:

Each league that you create in OOTP 2006 must be based in a single nation, the "League Nation." The list of nations available to you when you create a new league is populated using the nations in your nations.txt file. So, if you have edited your nations.txt to include only 20 nations, there will be just 20 nations to choose from here.

During league setup, the game randomly assigns cities to your teams. To do this, OOTPB chooses cities from your cities.txt file, using only cities from the league nation. Although the algorithm isn't precise, generally speaking, the higher the level of the league, the larger the cities that will be chosen.

Note: After league creation, you can change the nation and city for a single team by going to the Team Options & Ballpark page. This allows you, for example, to create a league with teams in multiple nations, such as having one team from Canada in a U.S.-based league. However, the overall league will still be based in the U.S.

OK, now you have a league, based in a certain nation, with teams based in cities from that nation. Once you click on Start Game, OOTPB has to create players for your league or leagues. Each league has a "Foreign Player Percent" option in the league setup that defines how many players in a league will come from nations other than the league nation.

3.5.1. Player Generation without Foreign Players

Let's assume for a moment that you have set foreign players to 0%, just for simplicity. When players are created for your league, each player is assigned an origin from the cities.txt file. However, they will only be chosen from cities in your league nation. So, if you have a league based in Canada, all players would be selected from Canadian cities.

Note: Players are more likely to be generated in cities with higher populations.

So now we have a pool of players for our league, each with a new origin from cities.txt. What shall we name these future Hall of Famers? That's where the names and nations files come back into play. The nations.txt file has fields that allow you to specify percentages for name sets, as described in Section 3.3. OOTPB will assign names to your players based on the name set settings for the league nation. For example, by default the U.S. is set to generate 71% U.S. names, 27% Hispanic names, and 2% Japanese names.

Note: Remember, at this point we are just defining names, and not nationalities. This is a league with 0% foreign players. We're saying that 27% of American players will have Hispanic names, not that 27% of our league will be players from Hispanic countries!

So there you have it; the game has now generated a full pool of players for our league, and assigned them appropriate origins and names.

3.5.2. Player Generation with Foreign Players

So, how does it work if you have turned up the foreign player % in your league? Let's say that you have set foreign players to 20% in a league based in the U.S. When players are created for your league, 80% of the players will be handled exactly as in the example above. These are the American players. But what about the other 20%?

When generating foreign players, the first thing OOTP does is to look at the **baseball level** of the all nations other than the league nation. The baseball level is defined in nations.txt, and can be a value from 0 (nonexistent) to 5 (excellent).

The higher the baseball quality in a country, the greater the chance of a foreign player being generated from that country. Similarly, the higher the baseball quality, the better chance of a player from that country being very good.

A rough breakdown of the percentages is below:

Baseball Quality	% of Foreigners
5 / Excellent	About 70%
4 / Good	About 15%
3 / Average	About 10%
2 / Fair	About 5/10th of 1%
1/ Poor	About 1/10th of 1%
0 / Nonexistent	0

This means that about 70% of foreign players generated will come from nations with a baseball quality rating of "excellent." About 15% will come from nations with a baseball quality of "good," and so forth. No players are generated from countries with nonexistent baseball quality.

Note:The baseball quality is also a factor when the game creates "hidden players" in nations without leagues, to be found by GMs scouting other nations.

3.6. Example: Editing Nations.txt

You can edit nations.txt with any text editor or Microsoft Excel. For more information on editing using Microsoft Excel, see Section 11.0.

Let's add a new fictional nation called "Markusonia" to our nations list. Markusonia will be a German- and English-speaking, baseball-crazy nation in the middle of the ocean near Australia.

1. Open nations.txt using a text editor.
2. Scroll to the bottom of the file and create a new row after the row for Nauru.
3. Enter the value 227 followed by a comma. We are assigning our new country an ID of 227, because it is the next unique number available. It could actually be any unique number that hasn't already been used in the file. Your new line looks like this so far:

```
227,
```

4. Next, type the following: "Markusonia,Markusonia,MKS,Markusonian,". This is assigning our new country's long description, short description, abbreviation, and nationality. Make sure your abbreviation does not match an existing abbreviation. The easiest way to do this is to just search the nations.txt file for the abbreviation you wish to use. So far, your new line should look like this:

```
227,Markusonia,Markusonia,MKS,Markusonian,
```

5. Next, type the following: "5,5,". We are assigning Markusonia to the Oceania continent. We are also setting the baseball quality level to 5, because of course baseball talent is high in Markusonia. Your line now looks like this:

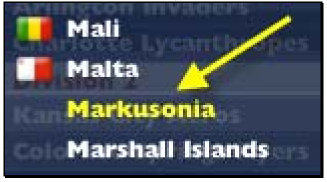
```
227,Markusonia,Markusonia,MKS,Markusonian,5,5,
```

6. Next, type the following: "7,60,0,40,,,,". This is saying that 60% of the player names in Markusonia should be taken from name set 7, German. The other 40% should be taken from name set 0, modern U.S. names. We are including four commas after the last value to represent the two sets of ethnicities that we are not using. Your line should now look like this:

```
227,Markusonia,Markusonia,MKS,Markusonian,5,5,7,60,0,40,,,,
```

7. Save the file, and quit the text editor. The next time you launch OOTPB, Markusonia will now be an option in any nation drop-down:

Figure 2 Custom Country in Nation Drop-Down



<u>Important Notes About Adding Nations</u>

- Note, however, that Markusonia doesn't have any cities yet! If you're going to add nations, be sure to add cities so that the game can add teams or give players appropriate origins! See Section 3.7 for an example of adding cities.
- OOPTB requires that at least 2 nations be in the nations.txt file. If you only wish to have one nation, create a dummy nation with a baseball level of zero. It won't prevent it from displaying in-game, but no players will be generated from that location.
- Do not use commas within any of the fields in the file, such as in a place name
- Do not put spaces between fields of information!

3.7. Example: Editing Cities.txt

You can edit cities.txt with any text editor or Microsoft Excel. For more information on editing cities.txt using Microsoft Excel, see Section 11.3.

Above we created a new nation called "Markusonia." Now let's add a few cities to Markusonia.

1. Open cities.txt using a text editor.
2. Scroll to the bottom of the file and create a new row after the row for Zyuzelskiy.

3. Type the following in your new row: "Markusonia, 5000000,". This is creating a new city called Markusonia (the capital city for Markusonia), with a population of 5 million. Do not put any commas in the population even if the number is difficult to read! Doing so will cause the row not to work. Your new row should look like this so far:

```
Markusonia,5000000,
```

4. Type the following in your new row: "227,". This is saying that Markusonia is a part of nation ID 227, which we identified as the nation of Markusonia in nations.txt. Your new row should look like this so far:

```
Markusonia,5000000,227,
```

5. Type the following in your new row: "Markusonia,". This is saying that Markusonia is a part of the region called "Markusonia," (just like "New York, New York). Your new row should look like this so far:

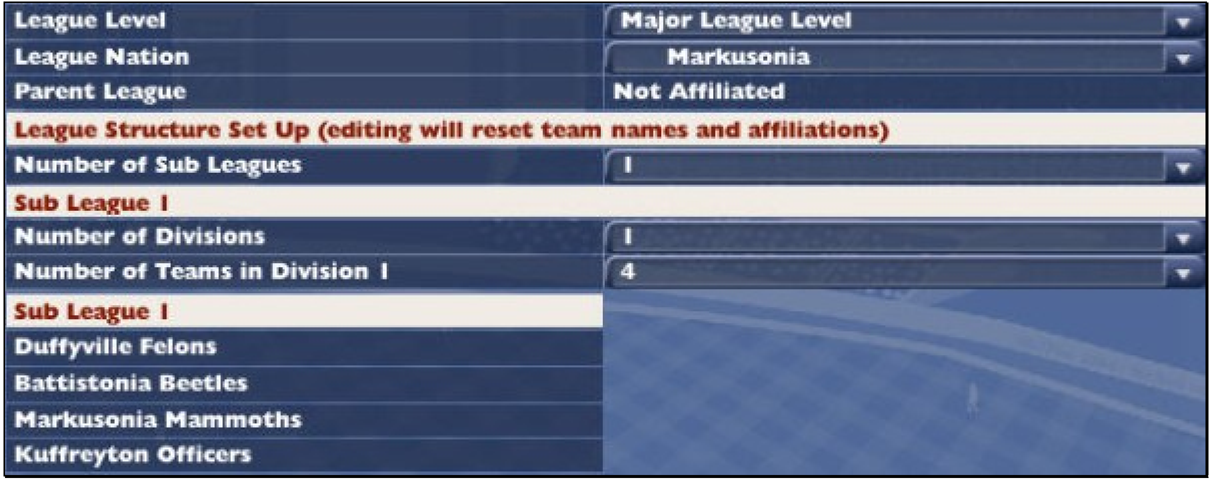
```
Markusonia,5000000,227,Markusonia
```

6. Type the following in your new row: "-16.26,156.47". This is saying that Markusonia is located at longitude -16.26, latitude 156.47. (Note that latitude and longitude can be negative numbers. A negative number for latitude makes the latitude south of the equator, such as Australia. A negative number for longitude makes the longitude west of the Prime Meridian, such as the United States.) Your row of data should now look like this:

```
Markusonia,5000000,227,Markusonia, -16.26,156.47
```

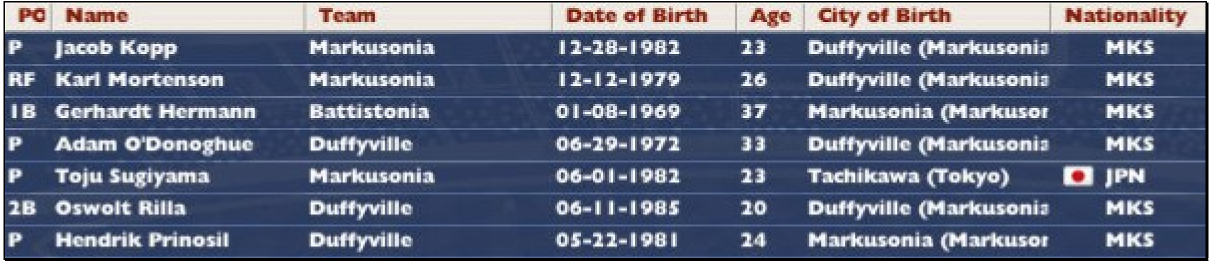
7. Repeat this process for as many cities as you wish.
8. Save and close the file. The next time you open OOTPB, these new cities can be used for team and player origins. If you have combined this with nation changes, you could get screens like the following:

Figure 3 League Based on Custom Nation



League Level	Major League Level
League Nation	Markusonia
Parent League	Not Affiliated
League Structure Set Up (editing will reset team names and affiliations)	
Number of Sub Leagues	1
Sub League 1	
Number of Divisions	1
Number of Teams in Division 1	4
Sub League 1	
Duffyville Felons	
Battistonia Beetles	
Markusonia Mammoths	
Kuffreyton Officers	

Figure 4 Players from Custom Origins



PO	Name	Team	Date of Birth	Age	City of Birth	Nationality
P	Jacob Kopp	Markusonia	12-28-1982	23	Duffyville (Markusonia	MKS
RF	Karl Mortenson	Markusonia	12-12-1979	26	Duffyville (Markusonia	MKS
1B	Gerhardt Hermann	Battistonia	01-08-1969	37	Markusonia (Markusor	MKS
P	Adam O'Donoghue	Duffyville	06-29-1972	33	Duffyville (Markusonia	MKS
P	Toju Sugiyama	Markusonia	06-01-1982	23	Tachikawa (Tokyo)	● JPN
2B	Oswolt Rilla	Duffyville	06-11-1985	20	Duffyville (Markusonia	MKS
P	Hendrik Prinosil	Duffyville	05-22-1981	24	Markusonia (Markusor	MKS

Important Notes About Cities

- For each city, make sure that you have a nation in nations.txt with a matching nation ID! See Section 3.6 for an example of adding nations.
- Do not use commas within any of the fields in the file, such as in the population!
- Do not put spaces between fields of information!
- Note that latitude and longitude can be negative numbers. A negative number for latitude makes the latitude south of the equator, such as Australia. A negative number for longitude makes the longitude west of the Prime Meridian, such as the United States.
- You may find this site helpful for real-world latitudes and longitudes: http://www.mapquest.com/maps/latlong.adp

3.8. Nation Flags

Nations can also have an associated flag, a small graphic that appears in the game next to the name of the country. These flags can be customized. Customizing nation flags is covered in Section **Error! Reference source not found.**.

3.9. For More Information...

If any of this was unclear, or if you have additional questions, visit the Names & Nations forum in the OOTPB community.

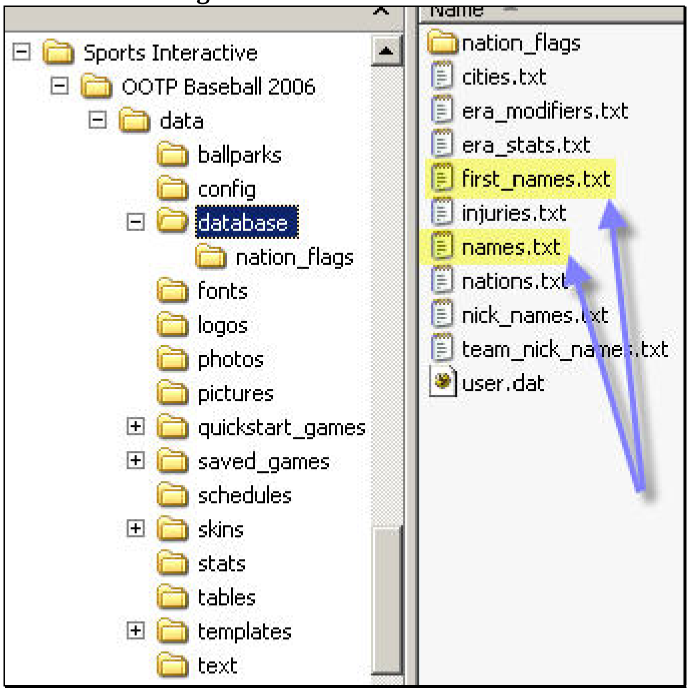
4.0 Names

So, you want to play a Finnish league, but don't want players named "Zack Maloney"? Names are one of the most frequently customized aspects of OOTPB, and they're easy, too. See Section 11.0 for more information on editing OOTPB text files using Microsoft Excel.

4.1. Where are the Names Files?

Names in OOTPB boil down to two text files: **firstnames.txt** and **names.txt**. These are located in the **data\database** directory in your OOTPB installation directory:

Figure 5 Names File Location



Unsurprisingly, **first_names.txt** includes first (or "given") names, and **names.txt** includes last names (or "surnames").

4.2. When are Name Files Used?

The name files come into play only once: at the creation of a game. At that time, all of the names in the names files are converted into a .dat file that is stored in the saved game directory. The names.dat file is then used by the game to dole out names during 2 scenarios in the game: when new players are created, and when new coaches are created.

New players are created at the start of a game, and during any subsequent amateur drafts. In leagues without amateur drafts, some free agents are created annually. New coaches are created at the start of the game, and occasionally thereafter to replace those that have retired.

Note: This means that you cannot change the names files used by your saved game after the initial game creation.Once a game is created, it will always use the same names.

4.3. Understanding the Name Files

The name files are comma-separated files, which means they can be edited quite easily. See Section 10.0 for more information about comma-separated files. The names files store three pieces of data for each name as follows:

Field #	Value	Description
1	Name or First Name	In the names.txt file, these are all surnames. In first_names.txt, they are all first names.
2	Frequency	The usage frequency for this name, within its name set.
3	Name Set ID	The unique identifier of the name set for this name, a number between 0 and 39.

Note: Although most of the names in the files are sorted alphabetically, they do not necessarily need to be that way.

4.4. About Name Sets

A "name set" is a group of names, usually belonging to one ethnic origin rather than a specific nation. For example, there is a Japanese name set, a French name set, and so on. Each name set has a unique numeric ID number, starting at 0 and ending at 39. OOTPB supports up to 40 name sets. The names files that come with the game already have 40 defined name sets, as follows:

ID	Nationality
0	U.S. (Modern)
1	Hispanic
2	Japanese
3	South Korean
4	Chinese
5	Portuguese
6	Dutch
7	German
8	French
9	Italian
10	Norwegian
11	Finnish
12	Swedish
13	Russian
14	Czech/Slavic
15	Polish
16	Scottish
17	Arabic
18	Irish
19	African
20	UK
21	Albanian
22	Serbian
23	Greek
24	Turkish
25	Danish
26	Armenian
27	Indian
28	Australian
29	Azerbaijan
30	Polynesian
31	Austrian
32	Canadian
33	French-Canadian
34	Filipino
35	Romanian
36	Indonesian
37	Pakistani
38	Vietnamese

39	U.S. (Historical)

So, let's look at an example from the first_names.txt file:

```
John,192,0
```

This means that the first name John has a frequency of 192, and a name set ID of 0. Looking at the chart, we see that this is part of the modern U.S. name set. Name sets get associated with nations in the game, so that you can define what types of names are generated for what nations. We'll cover that in more detail in a bit.

Although there are 40 possible name sets, these are all stored in the two names files. So, the first_names.txt file has first names for all 40 name sets. The same goes for names.txt and surnames.

Note: If a name set ID is used in nations.txt for which no names exist, then players will receive a name of "Joe Nobody."

4.5. About Name Frequency

Each name in a name file gets a frequency value, the second value in each row of data. This is just a numeric value, where higher numbers mean more frequent names usage. For example, if your U.S. name set looked like this:

```
David,50,0
John,50,0
Michael,50,0
Steve,25,0
Tom,25,0
```

This would tell us that the names David, John, and Michael would be used with roughly the same frequency (50). Steve and Tom would have about the same frequency as well (25), which would be about half as frequent as David, John, and Michael. So, if your league had 40 players, you might expect to see 10 Davids, 10 Johns, 10 Michaels, 5 Steves, and 5 Toms.

Three important notes about frequency:

1. All names should have a frequency. If you're not sure about frequency, or just want all names to be used equally, simply set all frequencies to 1. This is done in many of the name sets that come with the game.
2. Frequency is NOT a percent. There is no maximum value. You could rate your names on a 1-100 scale, 1-1000, or whatever scale you like.
3. Frequencies are specific to a name set. For example, you could use a 1-10 scale for U.S. names, a 1-100 scale for French and 1-1000 for Arabic. As long as you are consistent within a name set, the game will utilize the frequencies correctly.

4.6. Foreign Characters

Name files do support non-English characters. However, not all of the in-game fonts have all foreign characters, so there may be some places in the game that cannot display accents or other foreign characters correctly. The scoreboard in play-by-play is one such area. Double-byte characters (Japanese, Chinese, etc.) are not supported.

4.7. Linking Names and Nations

As discussed in Section **Error! Reference source not found.** on nations and cities, the nations.txt file includes fields that define which name sets are used for each nation. You can assign up to four different name sets to a single nation. This means that when the game creates players from a certain nation, it can choose names from up to four different name sets, allowing you to represent diverse populations in a nation without getting mixed names like Chin-min Ramirez.

For example, let's look at this line from nations.txt:

```
206,The United States,U.S.A.,USA,American,4,5,0,71,1,27,2,2
```

The name sets begin in the eighth field, the characters listed in blue above. Each name set is paired with a %. So, "0,71" means "name set 0, 71%," which, translated into English, means:

"71% of the names for players from the United States should be created using Name Set 0 (U.S. Modern)"

If we add in the next two pairs, we end up with this information about the U.S:

"71% of the names for players from the United States should be created using Name Set 0 (U.S. Modern)"
"27% of the names for players from the United States should be created using Name Set 1 (Hispanic)"
"2% of the names for players from the United States should be created using Name Set 2 (Japanese)"

A few important notes here:

1. The percentages for each nation should add up to 100.
2. These settings define the *names* for the players from each nation. The *origin* of these players will match the country, however. In other words, in our above example, we're saying that 27% of the U.S. players will have Hispanic *names*, but will be *U.S. citizens*.

For more information on player origins, see Section **Error! Reference source not found.** on nations and cities.

4.8. Example: Creating a Custom Names File

Let's say you decide that you want to create an all-Hobbit league, based in Finland. Unfortunately, Hobbit is not one of the 40 name sets that come with the game. Here's a quick example of how you could do this:

1. Close OOTPB, if it's open.
2. Choose one of the 40 name sets that you don't need. Let's say you decide you don't need Vietnamese names (Name Set ID = 38)
3. Create a list of Hobbit names in text format, in the format described above (name,frequency,name set). Use the ID # you chose in Step 1 above for the name set in each row (in our example, 38). Create two separate files, one for first names, and one for last names. So, your data might look like this:

```
Frodo,1,38
Samwise,1,38
```

4. Back up the names.txt and firstnames.txt files in your OOTPB directory.
5. Open names.txt. Find and delete all of the rows of data that end with 38. (These will be near the end, since the files are in order by name set.)
6. Copy and paste your Hobbit last names into the names.txt file and save.
7. Repeat Steps 4 and 5 for your first names.

8. Back up the nations.txt file in your OOTPB directory.
9. Open nations.txt and find the row for Finland.
10. Go to the eighth field of information and change the values from "11,92" (92% Finnish) to "38,100" (100% Hobbit).
11. Go to the tenth field of information and delete the values "12,8," (8% Swedish). Leave the commas, though. The line should look like this:

```
69,Finland,Finland,FIN,Finnish,3,2,38,100,,,
```

12. Save the file.
13. Close all of the open files, and open OOTPB.
14. Create a new game with a fictional league. Choose Finland for the League Nation, and set the Foreign Player Percentage to 0.
15. Start the game, and go to **League Menu | Transactions | Show All Players** tab. You will see that all of the players have Hobbit names. If you check a player's profile, you will see that their nationality is still Finland.

That's it!

4.9. For More Information...

If any of this was unclear, or if you have additional questions, visit the Names & Nations forum in the OOTPB community.

5.0 Injuries

Not happy with the run-of-the-mill rotator cuff injuries? Want to pretend that the plague has infested your league? Hey injuries are customizable in OOTPB, too! See Section 11.0 for more information on editing OOTPB text files using Microsoft Excel.

5.1. Where is the Injury File?

Just one text file controls the injury data in OOTPB: **injuries.txt**. This file is located in the **data\database** directory in your OOTPB installation directory:

5.2. When is the Injury File Used?

Like the names files, the injury data is transferred to a file called injury.dat in your saved game directory when the game is created, so after the creation of a game, you cannot modify the injury data being used for that game.

Injury data is used, unsurprisingly, any time an injury occurs in your league.

5.3. Changing Injury Files After the Start of a League

Since the game compiles the injury.txt file into a .dat file, you cannot change injury data after the inception of your league. However, there is a workaround for this issue. If you wish to change your injury data *after* the creation of a league, you can do the following:

1. Back up your game files.
2. Modify the injury.txt file as desired.
3. Create a brand new game file and save the game. This will be just a temporary game.

4. Copy the injury.dat file from your new saved game directory into the saved game directory for the game whose injuries you wished to modify.
5. Start up your original saved game. It should now use updated injury data. You can throw away the temporary game you created. There's no need to keep it.

5.4. Understanding the Injury File

Like most OOTPB data files, injury.txt is comma-separated, which means it can be edited quite easily. See Section 10.0 for more information about comma-separated files. The injury file stores twelve pieces of data for each injury, as follows:

Field #	Value	Description
1	Minimum Injury Time	The minimum number of days the player will be injured with this injury.
2	Maximum Injury Time	The maximum number of days the player will be injured with this injury.
3	DTD %	This number should be between 0 and 100. If this number is greater than 0, this injury will be considered a "day-to-day" injury. Furthermore, the % of loss in player performance will be around the number entered here. So, if this is set to 10, then the injury is day-to-day, and will cause somewhere around a 10% loss in performance. There are some variables involved, so it may not be exactly 10%.
4	Season Ending Boolean	If this is 0, then there is no chance for this injury to be season-ending. If it is 1, there is a chance for the injury to be season-ending.
5	Career Ending Possible Boolean	If this is 0, then there is no chance for this injury to be career-ending. If it is 1, there is a chance for the injury to be career-ending.
6	While Pitching/Throwing Boolean	If this is 1, then this injury may occur while the player is pitching or throwing. Otherwise it is 0.
7	While Running/Sliding Boolean	If this is 1, then this injury may occur while the player is running or sliding. Otherwise it is 0.
8	Collision Boolean	If this is 1, then this injury may occur when the player is in a collision. Otherwise it is 0.
9	HBP/Fight Boolean	If this is 1, then this injury may occur when the player is hit by pitch, or is in a fight. Otherwise it is 0.
10	Illness Boolean	If this is 1, then this injury may occur due to illness. Otherwise it is 0.
11	Body Part Code	The body part code is a number from 1 to 4, representing broad categories, as follows: 1 – Arm (from the shoulder to the fingers) 2 – Legs (from the abdomen to the feet) 3 – Back (back, rib cage) 4 – Other (head, illnesses)
12	Diagnosis	Description of the injury.

Note: The order of the injuries in the injuries.txt file is not important.

5.5. Example: Adding a New Injury

Let's say you decide that you want to introduce the possibility that some of your players will contract terminal illnesses. Here's a quick example of how you could do this. Let's create an option of "mononucleosis":

1. Close OOTPB, if it's open.
2. Back up the injuries.txt file in your OOTPB directory.
3. Open injuries.txt.
4. Scroll to the end of the file. Create a new row, and type the following: "15,90," This is saying that players who contract mononucleosis will be out somewhere between 15 and 90 days. Your new row should look like this so far:

```
15,90,
```

5. Type the following in your new row: "0,". This is saying that the injury is not a day-to-day injury. Your new row should look like this so far:

```
15,90,0,
```

6. Type the following in your new row: "1,0,". This is saying that mononucleosis might be season-ending, but cannot be career-ending. Your new row should look like this so far:

```
15,90,0,1,0,
```

7. Type the following in your new row: "0,0,0,0,1,". This is saying that mononucleosis cannot occur in throwing, running, in collisions, or when hit by a pitch. It can only happen due to illness. Your new row should look like this so far:

```
15,90,0,1,0,0,0,0,0,1,
```

8. Type the following in your new row: "4,Mononucleosis". This is saying that mononucleosis is part of the body part 4 category ("other"), and that it is called, of course, "Mononucleosis." Your new row should look like this:

```
15,90,0,1,0,0,0,0,0,1,4,Mononucleosis
```

9. Save the file and quit. In the next OOTPB game you create, mononucleosis will be an option in your game!

6.0 Team and League Logos

OOTPB has built-in functionality that generates new logos for all teams in your game universe. You can also change the colors used in auto-created logos through the Team & Ballpark Options page, described more fully in the Game Guide. In addition, you can create your own logos that will appear both in the game and on any game reports that you generate.

This section covers the creation of custom team logos and custom league logos. There are also many other graphics customization options in OOTPB. You can create custom ballpark graphics, covered in Section 9.0. Custom player photos are covered in Section 7.0. Additionally, customization of a number of other graphic elements is described in Section **Error! Reference source not found.**, modification of the game's "skin." This includes custom nation flags, changing the SION Network logo, modifying the basic graphic elements of the game, and so forth.

Customizing logos requires a decent understanding of graphics editing. While this guide will attempt to "dumb down" some of the more complicated aspects of graphics editing, we assume that you have access to a graphics editing tool such as Adobe Photoshop, and have some understanding of graphics editing.

Warning! *As with all customizations, you should always close OOTPB when making changes to files.*

6.1. Where are Team and League Logos Stored?

If you have installed OOPTB in the default directory, team and league logos are stored in the following directory:

C:\Program Files\Sports Interactive\OOTP Baseball 2006\data\saved_games\Test1.lg\news\html\images

In this case, "Test1.lg" is the name of your game universe.

This directory also contains a number of other graphics that are described in Section **Error! Reference source not found.**, modification of the game's "skin."

Note: Team and league logos "local" to a game universe. That is, they are not shared among all game universes on your computer.

6.2. Logo Naming Convention

Team and league logos must follow a specific file naming convention, as follows:

Logo	Naming Convention	Example
Team	city name_team name	Long Beach_Singing Trouts
League	leaguename	Dominican Winter League

Note: Spaces are allowed in logo file names.

6.3. Logo Image Size and Format

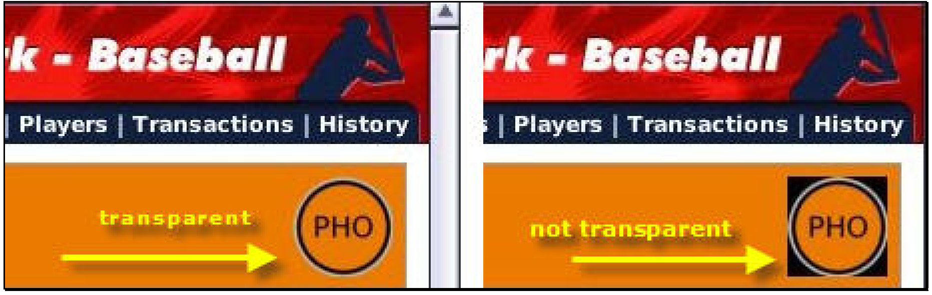
Logo files must be in either .JPG or .PNG format. .PNG format has the advantage of allowing transparency, which is not supported in .JPG files in OOTPB. .PNG images must be in RBG color mode. Logo images must be 150 pixels x 150 pixels. The game actually uses the same logo in many different sizes throughout the game. However, they are resized automatically by the game, so there is no need to create logos in multiple sizes.

Note: If you choose to create custom .JPG format logos, you must delete the .PNG format logos that were created by the game, or the the game will use those instead.

6.3.1. Creating Logos with Transparency

Many logo customizers quickly run into the issue of transparency. Transparency is a confusing topic for many people that want custom logos in OOTPB. If transparency is not handled correctly, you can end up with logos that don't look like you expect, such as graphics displaying with black backgrounds, instead of assuming whatever background color is on the screen. Below are two samples. The one on the right does not use transparency, and as a result ends up looking peculiar. The one on the left uses transparency:

Figure 6 Logo Transparency


rk - Baseball
Players | Transactions | History
transparent
PHO
rk - Baseball
Players | Transactions | History
not transparent
PHO

This section is a bit technical. It's intended to give people who are familiar with graphics editing the information they need to create OOPTB logos.

In previous versions of OOTP, an RGB setting of 0,0,0 was treated as transparent. This is not the case in OOTPB 2006. In OOTPB 2006, in order for an RGB 0,0,0 area to be used as transparent, it needs to be a mask embedded in the image’s alpha channel. Because .JPG files do not have alpha channels, this effectively means that you cannot use .JPG format for logos that need transparency. This also means that you can use RGB 0,0,0 in either a transparent or a non-transparent role.

Although the steps for creating a mask in an alpha channel vary depending on your graphics software, here are the general steps required to create a logo that has some transparent areas:

1. Create your 150x150 image.
2. Paint any areas you want to be transparent in RGB 0,0,0. For example, if you want a circular logo, paint the areas outside of the circle using RGB 0,0,0.
3. Create a mask from the image, selecting only RGB 0,0,0 for the mask. That is, the mask contains only the RGB 0,0,0 areas; all other RGB values are in the image layer. The steps required here may vary widely depending on which graphics software you use.
4. (Optional) If you look at the image's properties you should be able to see that the alpha channel now has contents. This is the mask. Most graphics programs will now let you display the image with the mask. If so, you should now see your RGB 0,0,0 areas as transparent.
5. Save the image in PNG format only.

This should create a logo image with some transparency.

***Warning!** .JPG files are merged images so no alpha channels will be saved. Therefore, all RGB 0,0,0 areas in .JPG files will be treated as non-transparent.*

6.4. Team Logo Colors

A team's logo colors (background and text) can also be adjusted from the Team & Ballpark Options page within the game. Colors are defined using hexidecimal codes, and not RGB codes that are often used in the graphics industry. Below are some links to sites that list hex color codes, or that can convert

http://www.w3schools.com/html/html_colors.asp
http://www.yvg.com/twrs/RGBConverter.html
http://drpeterjones.com/colorcalc/
http://wellstyled.com/tools/colorscheme2/index-en.html

Note: When entering new color codes for your team within OOTPB, be sure to hit the Enter key after each change, or else the change will not be saved.

6.4.1. Helpful Hex Codes

Below is a list of a few helpful hex codes for some very common colors:

Color	Hex Code
White	FFFFFF
Black	000000
Navy Blue	000080
Medium Blue	0000CD
Light Blue	1E90FF
Green	006400
Yellow	FFFF00
Gold	FFD700
Orange	FFA500
Red	FF0000
Purple	A020F0

6.5. Creating Custom Logos

The steps to creating and using a logo are fairly straightforward:

1. Close OOTPB.
2. If you are planning on overwriting existing team logos, first back up or delete those logos from the directory, according to the file location defined in Section 6.1.
3. Create a new team or league logo using the logo information in Section 6.3.
4. Save the file as a .JPG or .PNG file (.PNG only if using transparency.) Use the correct name, according to the naming conventions defined in Section 6.2.
5. Copy the file to the correct directory, according to the file location defined in Section 6.1. If necessary, overwrite the existing logo files.
6. Open OOTPB. Your team or league should now be using your custom logo.

Note: If you choose to create custom .JPG format logos, you must delete the .PNG format logos that were created by the game, or the the game will use those instead.

6.6. Converting Existing Logos

Many players have existing logos in a format other than .JPG or .PNG. For example, previous versions of OOTP used .BMP file, which is not supported in OOTPB 2006. Luckily, converting these files to the OOTPB format is not difficult.

To use existing logos with OOTPB, you must convert the images to either .JPG or .PNG format. Renaming the file does not convert an image; you must convert the image using a graphics program. There are many graphics programs capable of converting image formats. Irfanview is a free graphic program that allows you to convert many images at once, which makes it great for this sort of job.

Here is an example of the steps required to convert existing logos:

1. Create a folder to hold your original logos. We'll call it **C:\Original Logos**.
2. Create a folder to hold your converted logos. We'll call it **C:\Converted Logos**.
3. Copy your .BMP format images to C:\Original Logos.
4. Start up your image converting program (Irfanview, for example).
5. Select all of the images in C:\Original Logos.

6. In your image converting program, select the option to convert the files to .PNG format.
7. Set the option to save the converted files to C:\Converted Logos.
8. Run the conversion process.
9. Open C:\Converted Logos and rename the images to the naming convention described in Section 6.2.
10. Copy the renamed .PNG images in C:\Converted Logos and paste them into the images directory specified in Section 6.1, replacing the default logos.
12. Run OOTP Baseball 2006. Your teams should now be using their new logos.

6.7. Using the Same Logos in Multiple Game Universes

In general, team and league logos are specific to a single game universe. You can easily share logos among game universes by copying the logos from one directory to another. However, OOTPB also has some functionality that can be helpful if you plan on using the same logos across many game universes.

OOTPB has a master logo directory located here:

C:\Program Files\Sports Interactive\OOTP Baseball 2006\data\logos

Any logos in this directory will automatically be copied to any newly created game universe. So, for example, let's say you have created a game universe with custom logos, and you want to use those same logos for several additional game universes. Copy all of your custom logos into the directory listed above. Then, when you create a new game universe, all of the logos will be copied automatically.

6.8. For More Information

If any of this was unclear, or if you have additional questions, visit the Logos, Graphics, and HTML forum in the OOTPB community. The forum is also stocked full of custom logos that our talented community members have already created!

Additionally, many logo-makers offer their services to create custom logos for players who are not as accomplished at creating logos. Feel free to post requests!

7.0 Player and Coach Photos

OOTPB allows you to add player or coach (including all personnel) photos to the game. These photos are used in numerous places in the game: the player profile (for players not on a team), news articles about the player, HTML reports about the player, and in the SION broadcast screen.

7.1. Where are Photos Stored?

If you have installed OOPTB in the default directory, all player and coach photos are stored in the following directory:

C:\Program Files\Sports Interactive\OOTP Baseball 2006\data\saved_games\Test1.lg\news\html\images

In this case, "Test1.lg" is the name of your game universe.

7.2. Photo Naming Convention

Photo files must follow one of two file naming conventions, as follows:

Type	Naming Convention	Examples

Name	Firstname_lastname	Dwight_Evans David_O'Malley Jorge_De la Rosa
ID	Player_#	Player_2344

Either format is acceptable, and you can mix formats. That is, you can have some photos in the name format, and some photos in the ID format.

Each format has advantages and disadvantages. Many players may find that they need to use a combination of the two to get accurate pictures for each player and coach in their game. See below for more information.

Note: The name format will always take precedence over the player ID format if both exist in the directory.

7.2.1. Using the Name Format

Here are a few things you should consider when using the name format for player or coach photos:

Advantages

- Name files are much easier to create and name, and file names are intuitive.
- Names containing apostrophes ("Joe O'Malley") or spaces ("Jorge De la Rosa") can use the firstname_lastname format. So, Joe_O'Malley.jpg and Jorge_de la rosa.jpg are both valid photo names.

Disadvantages

- Names containing accents ("José Cruz") cannot use the firstname_lastname format. For players or coaches with accents, you must use the ID format.
- If 2 people have the exact same name, each person will have the same photo.

7.2.2. Using the ID Format

Here are a few things you should consider when using the ID format for player or coach photos:

Advantages

- You don't have to worry about accents in player names. IDs handle any kind of name equally well.
- Even if players have identical names, they can have distinct photos when using the ID format.

Disadvantages

- Players and coaches both currently use the "player_#" format. However, coaches and players have distinct numbering systems, so that you can have a "coach #1" as well as a "player #1". This means that if you create a photo named "player_1.jpg", it will be used for both *player* number 1 and for *coach* number 1. (*Note: This is anticipated to change in patch #2, planned for release in July. At that time, you should be able to use coach_# instead of player_#, thus eliminating this issue.*)
- Player IDs are not terribly intuitive.

To find a player or coach ID, you need to export database files from your game. Some information on this can be found in the Game Guide, but the simplest way to look up a player or coach ID number is as follows:

1. Open your game in OOTPB.
2. Select **Database Info** from the **Game Menu**.
3. Select **Export data to CSV files** from the **Tools** action menu in the top right.

4. Quit OOTPB.
5. Navigate to C:\Program Files\Sports Interactive\OOTP Baseball 2006\data\saved_games\Test1.lg\import_export (using your game's specific directory).
6. Open players.csv (or coaches.csv) in Microsoft Excel, Notepad, or any other program.
7. Search for the player in this file, using their name. The very first number in the row for that player is the player's ID.

Once you have found the player or coach ID, you can then use this ID in a photo name. For example, if you find that your player's ID is 2243, you could create a custom photo called **player_2243.jpg**.

7.3. Photo Image Size and Format

Photo files must be in .JPG or .PNG format. Photo images must be no larger than 90 pixels wide by 135 pixels high. There is no limit on the size of the file in bytes, although about 5 KB is recommended.

7.4. Where are Photos Used?

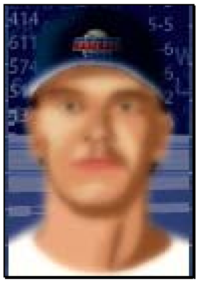
Player and coach photos are used in multiple places within the game:

- The Player or Coach Profile page
- The Player or Coach in-game report
- Other in-game reports, such as the team home page
- The SION Broadcast screen in-game

7.5. Replacing the Default Player Photo

If you are not currently using custom photos, you're probably well-acquainted with the default player photo in OOTPB:

Figure 7 OOTPB Default Player Photo



The default player photo appears for any player who is not currently on a team. If a player is currently on a team, the player's team logo is used instead of the default player photo.

You can change the default player photo. The easiest way to change the default player photo is as follows:

1. Close OOPTB, if it is open.
2. Find the image you wish to use. Like player photos, this image should be 90 pixels wide by 135 pixels high, but this image should be in .jpg format, not .png.
3. Name your new file "default_player_photo.jpg."
4. Find the file with the same name in the images directory in the skin of your choice. For example, if you use the brown-green skin, you would go to C:\Program Files\Sports Interactive\OOTP Baseball 2006\data\skins\brown green\images.
5. Replace the default_player_photo.jpg file in that directory with your new file.

6. This next piece is critical to having the change work. Navigate to the C:\Program Files\Sports Interactive\OOTP Baseball 2006\data\skins\brown green\ directory, or the identical folder for whichever skin you are using. Rename the **skin.bin** file to **skin.old**.
7. Open OOTPB. The game will take some time to load, as it is now rebuilding the skin. This may take 5-10 minutes, depending on your computer.
8. Once the game opens, your new default player photo should be in effect! If you changed the file in multiple skins, repeat Step 4-6 for each skin that you want to change.

This change will give you a new default player photo any time you use this skin. The change is specific to the skin, so no matter which saved game you load, as long as you use the same skin, the game should use your new default player photo.

7.6. Converting Existing Photos

Previous versions of OOTP used .BMP format for player photos. In order to use these photos in OOTPB 2006, you must convert the images to .JPG format. Renaming the file does not convert an image; you must convert the image using a graphics program. There are many graphics programs capable of converting image formats. Irfanview is a free graphic program that allows you to convert many images at once, which makes it great for this sort of job.

Here is an example of the steps required to convert existing photos:

1. Create a folder to hold your original photos. We'll call it **C:\Original Photos**.
2. Create a folder to hold your converted photos. We'll call it **C:\Converted Photos**.
3. Copy your .BMP format images to C:\Original Photos.
4. Start up your image converting program (Irfanview, for example).
5. Select all of the images in C:\Original Photos.
6. In your image converting program, select the option to convert the files to .JPG format.
7. Set the option to save the converted files to C:\Converted Photos.
8. Run the conversion process.
9. Open C:\Converted Photos and rename the images to the naming convention described in Section 7.2.
10. Copy the renamed .JPG images in C:\Converted Photos and paste them into the images directory specified in Section 7.1.
12. Run OOTP Baseball 2006. Your players should now be using their new photos.

7.7. For More Information

If any of this was unclear, or if you have additional questions, visit the Rosters, Photos, and Quick-Starts forum in the OOTPB community. Many users post links to photos sets they have already created, so you may be able to accomplish your goals without all of the work that goes into creating player photos!

8.0 In-Game Text

One of the neatest features of OOTPB 2006 is customizable text. Much of the text used in the game can be customized by editing a single XML file. The in-game text is sometimes referred to as "the xml," "the text engine," "the in-game text," or "the text database." This section will cover customization of text in detail.

8.1. What Can and Cannot be Customized?

While there is great flexibility in the OOTPB text engine, not everything can be customized. Below is a list of everything that can be customized in the game:

- Private contract negotiation messages (player refuses deal, accepts deal, etc.)
- Game news articles (shutouts, hitting for the cycle, perfect game, etc.)
- League event messages (start of the season, start of the playoffs, Rule 5 Draft, etc.)
- League news articles (award announcements, manager firings/hirings, public contract news articles, etc.)
- Manager news articles (talent drop/increase messages, scouting reports, etc.)
- Play-by-play commentary

It's important to understand that you can customize *text*, but not the *logic* that decides how and when the text is used. For example:

- You can change the text of a message that appears when a player wins the MVP award, but you cannot change the logic that causes the game to generate that message.
- You can change the text that is used during a double-play ground ball, but you cannot change the logic that determines when a double-play ground ball occurs.
- You can change the text in a scouting report, but you cannot change the information that the game provides, such as providing a list of 20 players instead of 10.

8.1.1. Exceptions

As with most things, there are exceptions in OOTPB; certain pieces of text that cannot be customized in the current version. Here are some of those things:

- Some messages and news articles, such as arbitration results notifications and trade offers /responses.
- Some play-by-play text, such as game introductions, certain plays in the field, and statistical notes such as "That's his 10th homer of the season."

8.2. Where is In-Game Text Stored?

All in-game text is located in a single XML file, **english.xml**. If you have installed OOPTB in the default directory, this file is stored in the following directory:

C:\Program Files\Sports Interactive\OOTP Baseball 2006\data\text\english.xml

8.3. What is an XML File?

Without getting too technical, an XML file is a specially-formatted text file somewhat similar to HTML. However, XML is fundamentally designed to carry data, while HTML is designed to display data. That's an awfully broad description. If you're at least familiar with HTML, you could find more information on XML here: http://www.w3schools.com/xml/default.asp. There are many other resources on the Internet to learn more about XML.

For purposes of editing OOTPB text, however, perhaps more important than a deep understanding of XML is simply understanding how to tweak the OOTPB XML file.

XML files consist of "elements" that are defined by text enclosed in the characters <>. Each element represents a different piece of data. All elements in an XML file must have a "closing tag," which is marked by the characters </>. So, you may see text like this in the game:

```
<TEXT_CATEGORY id="13">
	<TEXT_OBJECT id="28" text="[%personname L] advances to second base"/>
</TEXT_CATEGORY>
```

The first line here means that you have an element called TEXT_CATEGORY, and this particular element has an id of 13. The last line is the closing tag for the TEXT_CATEGORY element. In between is a completely separate child element called TEXT_OBJECT, which is a subset of TEXT_CATEGORY.

We'll go into more detail on the specific elements used in the OOTPB XML file later on.

Warning! *In XML, elements are case-sensitive. That is, you cannot write <TEXT_CATEGORY> in one tag, and then write </Text_Category> for the closing tag!*

At the heart of it, XML files are simply text files that can be edited with any program that can edit text files. Luckily, every operating system has at least one text editing program by default; Windows has Notepad and Word Pad, Macs have Write, and so forth.

8.4. Understanding the Text Database

There are four important concepts to understand when editing the text database: categories, objects, tokens, and conditions. In addition, there are a number of other small rules and tricks that are important in editing the text database. This section will describe these concepts in detail.

8.4.1. Categories

Text in the game is divided into "categories," each of which is called by the game's code whenever certain situations occur. Simply put, each category represents an "event" in your baseball universe. OOTPB 2006 has over 350 categories. Here are some examples of text categories:

Category	Used...
GAME_NEWS_SHUTOUT	When a player pitches a shutout
PLAYER_CONTRACT_REACTION_TOO_FAR_FROM_FAMILY	When a player turns down a contract offer because he feels it is too far from his home
BATTER_OF_YEAR_SUBJECT	As the subject header when a player wins the batter of the year award
ADVANCE_BASE_2	In play-by-play, when a player advances safely to second base

In the XML file, categories are represented by the TEXT_CATEGORY element. Each element has a unique numeric identifier, like this:

```
<TEXT_CATEGORY id="13">
</TEXT_CATEGORY>
```

Categories themselves don't actually contain any text. They're simply, um, categories, that are then filled with text "objects," described in the next section.

A complete list of the categories, IDs, and their usage in Microsoft Excel format can be downloaded here. In this document, the categories have been further divided into logical groups: Game News, Manager News, Game Play-by-Play, League News, League Events, Player Season Descriptions, and Contract Negotiations. However, these groups are not actually used in the XML file, nor anywhere else in the game.

Remember that each TEXT_CATEGORY element must have the closing tag </TEXT_CATEGORY>.

Note: Since categories are only used when called by the game's code, please note that there will be no way for players to add new categories to the game.

8.4.2. Objects

As we mentioned, categories don't actually contain text. However, each category contains one or more "objects". Each object represents one possible text phrase to be used when that category is needed. So, objects all belong to a single category.

For example, for the ADVANCE_BASE_2 category, some of the objects might be phrases like this:

```
"Wouk advances to second"
"Wouk reaches second safely"
"Wouk goes to second"
```

In the XML file, objects are represented by the TEXT_OBJECT element. TEXT_OBJECTS are nested inside a TEXT_CATEGORY element, and each TEXT_OBJECT must have a unique numeric identifier, like this:

```
<TEXT_CATEGORY id="13">
   <TEXT_OBJECT id="28" text="[%personname L] advances to second
base"/>
</TEXT_CATEGORY>
```

Looking at the example above, we can see that TEXT_OBJECT # 28 is part of TEXT_CATEGORY # 13.

Remember that each TEXT_OBJECT element must end with a closing tag. In this case, instead of having a wholly separate tag, the tag is closed by ending the element with the characters />.

The more objects exist for a given category, the less likely you are to see the same phrase repeated over and over again. By default, every text object that you create can potentially be used by the game. For that reason, it is important to make objects very generic, unless you are using conditions, described below. For example, let's say you create the following text object (tokens removed for simplicity):

```
After hitting .300 this month, Davis wins the Batter of the Month
award. He really has a great career ahead of him!
```

If this object gets used with a young player, it will sound fine. But if the object is used when a 40-year old wins the award, it will sound awkward, because he doesn't have a great career ahead of him! So, it's important to keep things generic if you aren't using conditions.

Unlike categories, objects can be customized to your heart's content. There is no limit to the number of objects allowed per category, or total number of objects overall. The only requirement is that each must have a unique numeric ID. That is, you cannot have two TEXT_OBJECTS in the XML file with the same ID, even if they are in different categories.

8.4.3. Tokens

If you examine the XML file, you will see many phrases with text enclosed in brackets with a % sign, like this:

```
[%personname L]
```

These are known as "tokens," and they represent variables. For those of you unfamiliar with programming, tokens are used to represent pieces of information that will be pulled from the OOTPB game. There are over 300 tokens in OOTPB.

For example, let's say we're writing a few text objects to be used when a player makes a catch in the outfield. We don't know which player will be making the catch. So, how can we ensure that the game will show the correct player's name? The answer is, we use a token. By putting the [%game fielder of] token into a text object, OOTPB will know to substitute the name of the player actually making the catch in the game.

Tokens are essential to editing text objects. Nearly every text object in the game uses at least one token. Tokens can be mixed and matched to make creative text objects. For example, you might see an object that looks like this:

```
The [%subleaguename] honored [%personlink] with the [%batteraward]
earlier today. [%leagueyear] was a career year for the [%teamlink]
[%playerposition], who posted a batting average of [%batting avg],
with [%batting rbi word] and [%batting hr] dingers for good measure.
```

Wow! That's confusing! But when all of those tokens are converted into real values by the game, you can end up with a message like this:

```
The National League honored Markus Heinsohn with the Super Slugger
Award earlier today. 2006 was a career year for the Berlin first
baseman, who posted a batting average of .355, with 145 RBI and 45
dingers for good measure.
```

A complete list of the available tokens and their usage in Microsoft Excel format can be downloaded here. There are also a few important things to remember regarding tokens:

- As with text categories, new tokens cannot be added by players, although more tokens will certainly be added in subsequent versions of the game, to allow even greater flexibility.
- Only certain tokens can be used in each category. See the spreadsheet for details on which tokens are available to each category.\
- Tokens must be placed in brackets [ ], and not parentheses or braces. They also must start with a % sign. If a token is not spelled correctly according to the token list, it will not work.
- Statistical tokens are dynamic, depending on the nature of the category. For example, if you use the [%batting hr] token in a Player of the Week award message, then the token will display the number of home runs the player hit *that week*. If you use the exact same token in a Player of the Month award message, it will display the number of home runs the player hit in *that month*.

8.4.3.1. Hyperlink Tokens

Some tokens for player and team names include the word "link." These tokens have special behavior in OOTPB. When a link token is used, that team or player name will be hyperlinked in the game. For example, if you use [%personlink], the result might be the name of a player, Steve Wilson. However, it will also be a hyperlink to Steve Wilson's player page within the game. We encourage you to use link tokens in all news messages.

However, there are a few special rules around link tokens:

- Link tokens cannot be used in categories with the word SUBJECT in the title
- Link tokens cannot be used in play-by-play

8.4.4. Conditions

By default, every text object that you create can potentially be used by the game, so most objects should be fairly generic in nature. You don't want your article talking about a "young star," when it could be talking about an over-the-hill veteran. That alone would have made OOTPB's in-game text severely limiting. Luckily, however, OOTPB also includes "conditions."

Conditions are applied to text objects, and influence when that object can be used. For instance, you can have an object that refers to a “long-time veteran," which only gets used if the player in question has been playing professional ball for more than 10 years. Some examples of conditions are as follows:

Condition	Used...
PERSON_AGE_MIN	When an object should only be used for players over a certain age
PLAYER_MLB_YEARS_MAX	When an object should only be used for players with less than X years of major league experience.
PLAYER_IS_UPCOMING_FREE_AGENT	When you want the object to consider whether the player in question will be a free agent.
GAME_BASES_LOADED	In play-by-play, when an object should only be used when the bases are loaded.

In the XML file, conditions are represented by the TEXT_CONDITION element. TEXT_CONDITION are nested inside a TEXT_OBJECT element, like this:

```
<TEXT_OBJECT id="2237" text="The [%personage]-year-old veteran
[%playerposition] has a lifetime average of [%batting avg].">
   <TEXT_CONDITION id="7" value="34"/>
</TEXT_OBJECT>
```

Conditions also contain an ID and a value. In the above example, TEXT_CONDITION #7 is the "PERSON_AGE_MIN condition. Since the value is 34, this means that this text object will only be used if the player is at least 34 years old. True/false conditions like PLAYER_IS_UPCOMING_FREE_AGENT usually have a value of 1 if true, and 0 if false.

A complete list of the available conditions, IDs, values, and their usage in Microsoft Excel format can be downloaded here. There are also a few important things to remember regarding conditions. As with text categories, new conditions cannot be added by players, although more conditions will probably be added in subsequent versions of the game, to allow even greater flexibility.

A text object can have multiple conditions assigned. For example, you could combine the following conditions into one object:

CATEGORY = PLAYER_CONTRACT_EXTENSION_RUMOR
PERSON_AGE_MIN = 34
PLAYER_IS_UPCOMING_FREE_AGENT = 1
CONTRACT_OFFER_IS_EXTENSION = 1

OBJECT:
Cagey veteran Will Smith today downplayed rumors that he was negotiating an extension with the Phillies. He repeatedly dodged the issue, turning aside all questions with a laugh. The 34-year old shortstop is in the last year of his contract, and is hitting .340 on the season, with 25 bombs. The Phillies need to wrap this guy up, or perhaps start looking for a younger solution!

Here is another example of a text object with conditions:

Example 2
CATEGORY = PLAYER_CONTRACT_REACTION_NEGATIVE
PLAYER_GREED_MIN = 4
CONTRACT_OFFER_IS_EXTENSION = 0 (free agent contract offer)

OBJECT:
Are you kidding me with that offer? Money talks, and I'm hearin' nothin' over here. Pony up some more cash, and some more respect, or go look for another shortstop!

As you can see, conditions make the text engine much more flexible. Of course, the more specific your conditions are, the less likely they are to get used.

8.4.5. (nl) – Creating New Lines

If you examine the XML file, you may see these characters frequently, such as in this object:

```
<TEXT_OBJECT id="11679" text="[%game batter] rounds
third...(nl)returns to the bag..."/>
```

Many text objects in OOTPB will be one paragraph only. However, you can create multiple paragraphs within a single object by entering the text (nl), including the parentheses. (nl) stands for "new line," and unsurprisingly, the result in OOTPB is that any time you use the (nl) command, a new line is created in the text. It often looks best to use two (nl)s in succession, so that a complete blank line appears between the two paragraphs. For example:

```
[%personlink] won the award today!(nl)(nl)"I'm so happy!" he said.
```

This text object results in the following text on-screen:

```
Steve Battisti won the award today!

"I'm so happy!" he said.
```

A few more notes about (nl):

- (nl) cannot be used in SUBJECT categories
- (nl) is case-sensitive. You cannot write (Nl) or (NL), or the text will simply appear in-game.

8.4.6. Other XML File Notes

In addition to the five topics covered above, there are a few more notes about the behavior of the XML file:

- Be careful when writing text objects in Microsoft Word. Word has a tendency to replace apostrophes and quotation marks with it's special "curly quotes." Curly quotes are not supported by OOTPB, and will result in garbled characters in your news messages. Similarly, Word also likes to turn ellipses (...) into a special ellipsis character, instead of three separate periods. This ellipsis character is also not supported. These can both be toggled on and off in Word's Auto Correct options.
- There are a handful of other functions within the XML file that affect the usage chance for certain options and the mutual exclusivity of objects. These, however, all pertain to game play-by-play commentary and are covered in that section.

8.5. How the Game Processes Text Objects

So, after all of that, how does OOTPB actually decide which text object to use? We'll talk about game play-by-play more later on, but for now, here's the general process the game follows:

1. When you start OOTPB, all text objects are randomized within their categories. (So, if there are 40 messages in the “Player of the Week” category, they are all shuffled.)
2. In the game, an event occurs that generates a message. For example, it just turned Monday, and the "Player of the Week" award is announced.
3. OOTPB looks for the appropriate text category in the XML file. In this case, it first needs a subject for the message, so it chooses PLAYER_OF_THE_WEEK_HITTER_SUBJECT.
4. OOTPB then looks at the shuffled list of objects in this category, and chooses the first option.
5. After an object is selected, it checks for any conditions attached to the object. If there is a condition, and the condition is not met, OOTPB discards the object and selects the next object. If the condition is met, or if there is no condition, then the game uses this object. The subject of the message has been decided.
6. Next, OOTPB needs a body for the message, so it looks in the PLAYER_OF_THE_WEEK_HITTER category and repeats Steps 4 and 5 above, selecting a message body. Now our news article has both a subject and a body.
7. OOTPB also may check the BATTER_SEASON_DESCRIPTION category to get a summary of the player's performance so far this year, and append it to the message body.
8. The game then displays the news article in the game.
9. The next time an article is required, this entire process is repeated. If it's another PLAYER_OF_THE_WEEK_HITTER category, then the game will pick up where it left off in the shuffled list of objects. Once OOTPB has gone through all of the options in a category, it will start again from the top in the same order, as long as OOTPB is not closed. Any time you close and re-open OOTPB, the objects are randomized again.

Let's walk through an example, with tokens removed to make it easier to follow. Let's say that 28-year old hitter Will Smith wins the Player of the Week award, and OOTPB is trying to generate that news article.

First it chooses the first PLAYER_OF_THE_WEEK_HITTER_SUBJECT object in the randomized list:

```
"Smith Wins Player of the Week in NL"
```

There is no condition assigned to this object, so that option "passes" and gets selected as the subject header. Next, the game chooses the first PLAYER_OF_THE_WEEK_HITTER object:

```
CONDITION: PLAYER_AGE_MIN = 33
"Veteran Will Smith slammed 12 RBI this past week, winning the Player
of the Week Award."
```

OK, there's a condition assigned to this object. This object should only be used if the player in question is at least 33 years old. Well, Will Smith is 28, so the condition fails. OOTPB discards this option and picks the next object in the randomized list:

```
"Will Smith pounded 12 RBI this past week to win the Player of the Week
Award. What a great job!"
```

There is no condition here, so this object is accepted. The game then moves on to pick a BATTER_SEASON_DESCRIPTION. It chooses this:

```
CONDITION: PLAYER_AGE_MAX = 32
"Smith is having a great season so far, hitting .360 with 20 HR and 65
RBI. He still has a lot of years left!"
```

This object also has a condition, but this time the condition says that the player must be under 32 years old, which Smith is, so the object is accepted. The game has now selected the 3 pieces it needs, and creates the news article as follows:

```
Subject: Smith Wins Hitter of the Month in NL

Will Smith pounded 40 RBI in this month to win the Hitter of the Month
Award. What a great job!

Smith is having a great season so far, hitting .360 with 20 HR and 65
RBI. He still has a lot of years left!
```

8.6. Play-by-Play Commentary

Sorry, not quite yet. :(

8.7. Editing OOTPB's XML File

The steps to actually editing the XML file are simple:

1. Shut down OOTPB.
2. Make a backup copy of your english.xml file.
3. Open english.xml in your favorite text editor.
4. Add, edit, or delete as many text objects as you wish, using the information on categories, objects, tokens, and conditions from above.
5. Save the english.xml file.
6. Start up OOTPB. Your new text will be used as your game proceeds.

Additionally, OOPTB community member **halos17** has created a neat tool for Windows users, that has a slick graphical interface for editing OOTPB's text. You can download the program here. We highly recommend this program for Windows users. It has a slick interface that makes editing the XML file much easier than editing manually.

8.8. XML Debug Options

OOTPB also has a handful of game engine options that are very helpful to players who wish to edit in-game text. While there would not usually be a need to change these during normal game play, they are invaluable for testing text.

There is a configuration file called **app** that holds the debug settings. If you installed OOTPB in the default location, this will be in c:\program files\sports interactive\OOTP Baseball 2006\data\config. You can open this file with Notepad or any text editor.

Warning! *You should always close OOTPB before editing this file.*

In the file, there are two options related to text editing:

add_debug_info_to_in_game_commentary

If you change the value for this entry from 0 to 1, when you play the game, the Category ID and the Text Object ID will be displayed for each text object that appears in the game, like this:

[330-6442] missed that offering by Dave McNiven, a fastball low...

In this case, the 330 is the category ID, and the 6442 is the text object ID. You can then find these within the english.xml file by searching for "id="6442"", for example.

This is incredibly helpful if you see an option that you don't like, because you can easily find it and eliminate it from the XML file.

This setting affects both game play-by-play and news articles.

dump_in_game_commentary_to_file
If you change the value for this entry from 0 to 1, the game will automatically dump the full play-by-play for all games simmed into a file in_game.log in the c:\program files\sports interactive\OOTP Baseball 2006\data\saved_games\X.lg folder (for example \data\saved_games\NewGame 321.lg\in_game.log").

It's important to note that the game will spit out PbP for every game on the docket, not just the one you are simming, so don't auto-play a long period of time with this turned on. This option also causes play-by-play not to display on the screen, and it slows down auto-play tremendously. So, it's really exclusively for debugging.

8.9. Using Different XML Files for Different Games

The english.xml file is shared across all games on your computer. That is, you cannot have one set of text for your historical league, and another set of text for your fictional league. However, because the english.xml file is loaded each time you start the game, you can always swap different copies of the files in and out. For example, if you have one version that has old-school play-by-play, you could swap that in when simming your 1950s league, and then replace it with a more modern-sounding file when playing a new fictional league.

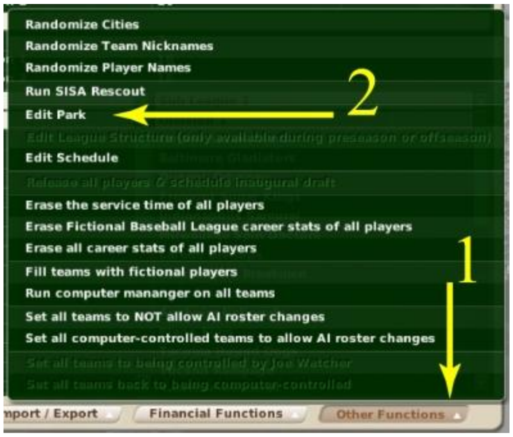
8.10. For More Information

If any of this was unclear, or if you have additional questions, visit the In-Game Text and PbP Commentary forum in the OOTPB community.

9.0 Ballparks/Stadiums

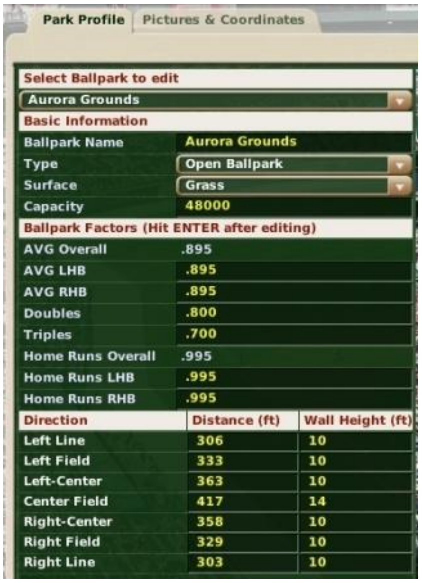
One of the long-time features of the OOTP series is the ability to add your own ballparks to your leagues. This section details the ballpark editor and its features.

The ballpark editor is accessed through the game setup screen (select “Game Setup” on the “Game” menu). Once at the game setup screen, select the “League Setup” tab and look to the bottom-right corner of the screen to the button marked “Other Functions.” On the pop-up menu, you will find the “Edit Park” option.


Randomize Cities
Randomize Team Nicknames
Randomize Player Names
Run SISA Rescout
Edit Park
2
Edit League Structure (only available during preseason or offseason)
Edit Schedule
Release all players & schedule inaugural draft
Erase the service time of all players
Erase Fictional Baseball League career stats of all players
Erase all career stats of all players
Fill teams with fictional players
1
Run computer mananger on all teams
Set all teams to NOT allow AI roster changes
Set all computer-controlled teams to allow AI roster changes
Set all teams to being controlled by Joe Watcher
Set all teams back to being computer-controlled
nport / Export
Financial Functions
Other Functions

9.1. A Tour of the Ballpark (Editor)

There are two tabs that make up the ballpark editor: park profile, and pictures & coordinates. The options on the park profile tab control the in-game performance of the ballpark while the options on the pictures and coordinates tab provide the control over the cosmetic aspects of the ballpark.



Select Ballpark to edit	
Aurora Grounds	
Basic Information	
Ballpark Name	Aurora Grounds
Type	Open Ballpark
Surface	Grass
Capacity	48000
Ballpark Factors (Hit ENTER after editing)	
AVG Overall	.895
AVG LHB	.895
AVG RHB	.895
Doubles	.800
Triples	.700
Home Runs Overall	.995
Home Runs LHB	.995
Home Runs RHB	.995

Direction	Distance (ft)	Wall Height (ft)
Left Line	306	10
Left Field	333	10
Left-Center	363	10
Center Field	417	14
Right-Center	358	10
Right Field	329	10
Right Line	303	10

At the top of the Park Profile tab is the ballpark selection drop-down list. From this list, you can select any ballpark that has already been defined in the current baseball universe. One a ballpark has been selected, the information pertaining to that particular park fills in the remainder of the form.

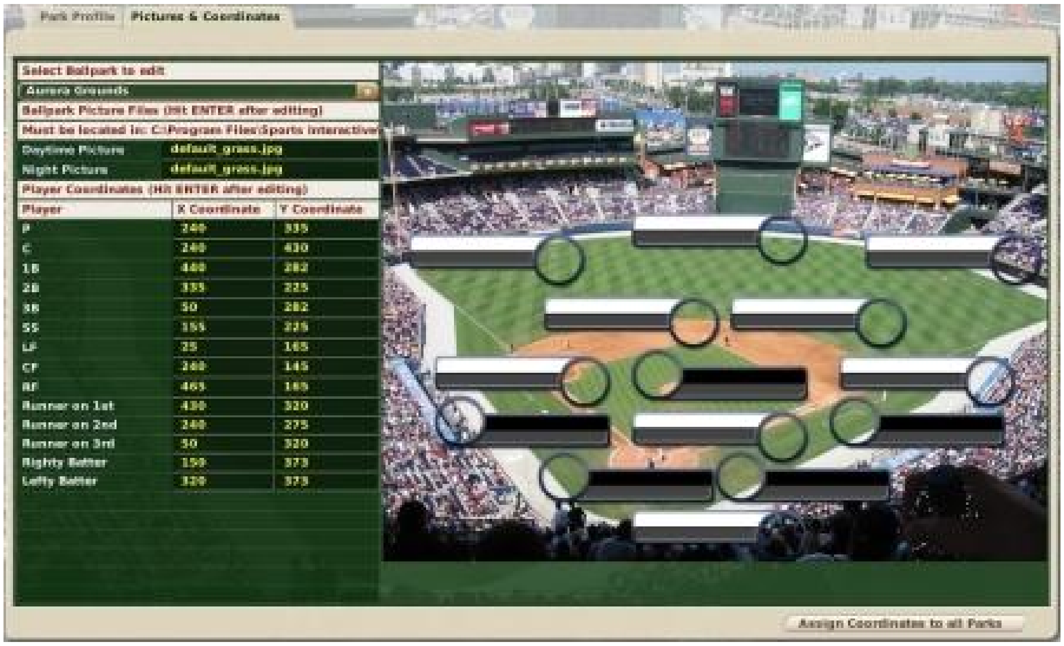
Under the Basic Information heading, you find the name of the ballpark along with other general information. This information can be changed at any time. Changes to the ballpark name and type are purely cosmetic.

The Ballpark Factors (located under the Basic Information section) control how the ballpark affects statistics. The batting average and homerun factors are split for left and right hand batters (LHB/RHB). The doubles and triples factors are not split and apply equally to left and right handed batters. If the values are set to 1.000, then the park plays according to the league average for that statistic. Values above 1.000

mean that more of that particular outcome will occur in games played that park than on the average. Similarly, values below 1.000 tend to suppress the corresponding result. For example, a pitchers park will tend to have batting average (AVG LHB & AVG RHB) and homerun (Home Runs LHB & Home Runs RHB) factors below 1.000.

Use caution when altering these numbers because the boxes will accept any numeric value, positive or negative. Numbers outside of the range from 0.500 to 1.500 will result in extreme statistical deviations for that park. In addition, the average value for each factor should be 1.000 when taken across all ballparks in the league. If the average value is too far from 1.000, then the league averages will not be the league averages. So, again, use caution.

At the bottom of the park profile tab is the fence distance and height table. Here, you can set the fence distances and heights for 7 different positions between the foul lines. The fence distances are purely cosmetic and used for the play-by-play text generation. So, use caution here, or you may end up with 1,000-foot homeruns or 100-foot homeruns. The homerun factors determine the likelihood of hitting homeruns in the ballpark, not the fence distances. On the other hand, the fence heights do have some role in the outcomes calculated by the game. If the fence distance is short enough (under 11 feet), then an outfielder has a chance to save a homerun that is just over the wall.



Select Ballpark to edit

Aurora Grounds

Ballpark Picture Files (Hit ENTER after editing)

Must be located in: C:\Program Files\Sports Interactive

Daytime Picture: default_grass.jpg

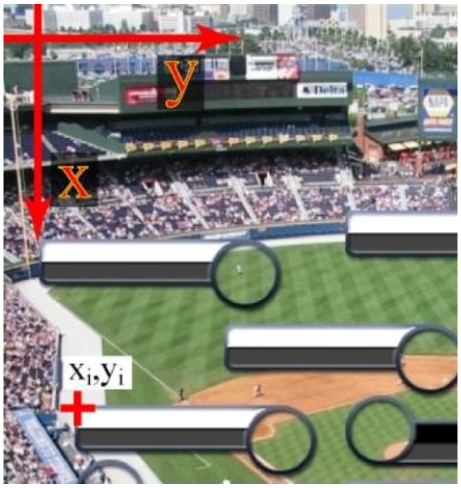
Night Picture: default_grass.jpg

Player Coordinates (Hit ENTER after editing)

Player	X Coordinate	Y Coordinate
P	240	335
C	240	430
1B	440	282
2B	335	225
3B	50	282
SS	155	225
LF	25	165
CF	240	145
RF	465	165
Runner on 1st	430	320
Runner on 2nd	240	275
Runner on 3rd	50	320
Righty Batter	150	373
Lefty Batter	320	373

At the top of the Pictures & Coordinates tab, we see the ballpark selection drop box again. Underneath the ballpark selection tab are places to specify the filename of the daytime and nighttime pictures for the ballpark. Daytime pictures are used for games that start prior to 7:00 PM on the schedule. That’s right! You can have different pictures based on the time of day. Pictures should be 640x480 pixels and saved in .jpg (like the default picture) or .png formats. Any ballpark pictures that you add to the game should be stored in the ..\data\ballparks directory (e.g., the default location is C:\Program Files\Sports Interactive\OOTP Baseball 2006\data\ballparks). If you want to see the daytime or nighttime picture displayed on the screen, then click in the text box and press ENTER. The appropriate ballpark picture should be displayed.

Below the ballpark picture filenames are the player coordinate settings. These are used to adjust the location of the fielder and offensive player displays on the ballpark picture. The coordinate system has the x-y origin in the upper left corner, as shown in the figure below. The coordinates for each of the players are established on the upper left corner of the label.



The default picture views the diamond from above and behind home plate. This is the "traditional" OOTP layout. Other, more creative, layouts may be used with adjusted coordinates.

In the lower-right is a button labeled "Assign Coordinates to all Parks." Use this button if the coordinates you have set for the current ballpark (identified in the select ballpark dropdown list) are appropriate for all of the ballparks in your baseball universe. Don't worry about pressing this button by accident. The game will present a confirmation message that gives you the opportunity to cancel the assignment of the current coordinates to all parks.

9.2. Ballpark Reconstruction

There is only one ballpark for each team in the game universe. New ballparks cannot be added to the baseball universe without adding a team to that saved game. So, in order to add a "new" ballpark to the game, you must edit the team's old ballpark to conform to the new specifications. There is no known way around this limitation at this time.

Once the ballpark is edited, you may want to assign a new name to the park. In addition to the park editor, ballpark names may be assigned through the team's Options and Ballpark page (accessible through a tab on the team's home page).

Multiple teams may be assigned to the same ballpark using the Commissioner Mode. While acting as the commissioner, the assigned ballpark for a team may be changed using the team's Options and Ballpark page.

9.3. For More Information

If you have questions, or if you have other information to share, you can go to the Logo, Graphics and HTML area of the OOTP community forums.

10.0 About Comma-Separated Files

Comma-separated (or comma-delimited) files are simple text files. They can be edited with just about any program that can read text files, including most word processing programs, Notepad, Write, and Microsoft Excel.

In a comma-separated file, each row represents a single thing, such as a nation or a name in OOTPB. Each row holds multiple pieces of information about that one thing. These pieces of information (or "values") are separated by commas. Hence, "comma-separated," because commas separate the pieces of data.

For example, in the OOTPB names files, we track three pieces of data for each name: the name, the frequency, and the name set ID. These pieces of data are separated by commas. So, the first names file begins with two lines like this:

```
A.C.,1,0
A.J.,1,0
Aaron,4,0
```

Since the commas represent divisions between different pieces of information, we could describe these two lines in plain English in the following way:

The first name "A.C." has a frequency of 1, and belongs to name set ID 0.
The first name "A.J." has a frequency of 1, and belongs to name set ID 0.

Most of OOTPB's text files are comma-separated, and behave in a similar fashion.

Note: Spaces are not needed after commas in comma-separated files. In fact, it is recommended not to use spaces.

11.0 Editing OOPTB Text Files in Microsoft Excel

Most of the text files used in OOTPB, such as the cities.txt file, are comma-delimited files that can easily be read into a Microsoft Excel spreadsheet. This is often a much easier way to edit these files, but some extra steps are required to get the file back into the proper format to be used in the game. Below are instructions for editing OOTPB's text files:

These instructions work for any of the OOTPB text files stored in the **data\database** directory, or C:\Program Files\Sports Interactive\OOTP Baseball 2006\data\database if you installed the game in the default directory.

Note: These instructions were written using Microsoft Excel 2003. They should work for most other versions of Excel, although the steps may vary slightly.

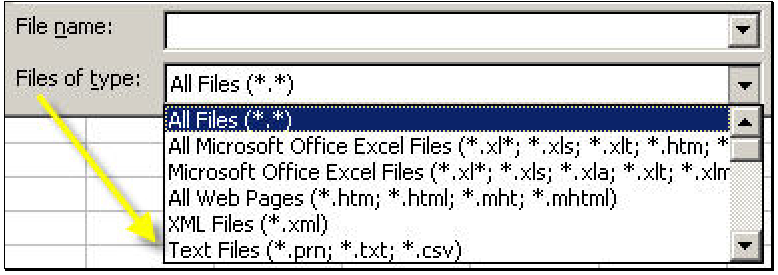
Warning! *Remember to make a backup copy of any file before modifying it!*

11.1. Opening Files in Microsoft Excel

Use these steps to open an OOTPB text file in Microsoft Excel:

1. Open Microsoft Excel.
2. Select **File | Open**.
3. Navigate to **C:\Program Files\Sports Interactive\OOTP Baseball 2006\data\database**.
4. In the "Files of Type" drop-down box, select "Text Files (*.prn, *.txt, *.csv)".

Figure 8 Importing into Excel



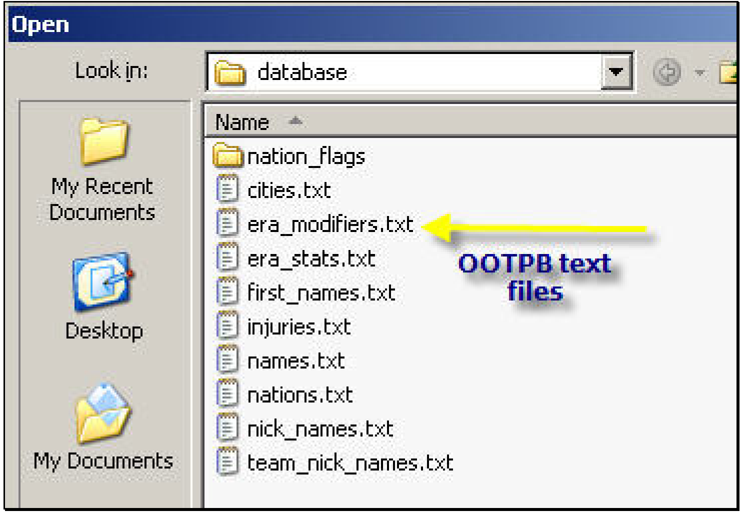
File name:

Files of type: All Files (*.*)

All Files (*.*)
All Microsoft Office Excel Files (*.xl*; *.xls; *.xlt; *.htm; *
Microsoft Office Excel Files (*.xl*; *.xls; *.xla; *.xlt; *.xlm
All Web Pages (*.htm; *.html; *.mht; *.mhtml)
XML Files (*.xml)
Text Files (*.prn; *.txt; *.csv)

5. The text files in the folder should now be visible:

Figure 9 Importing into Excel



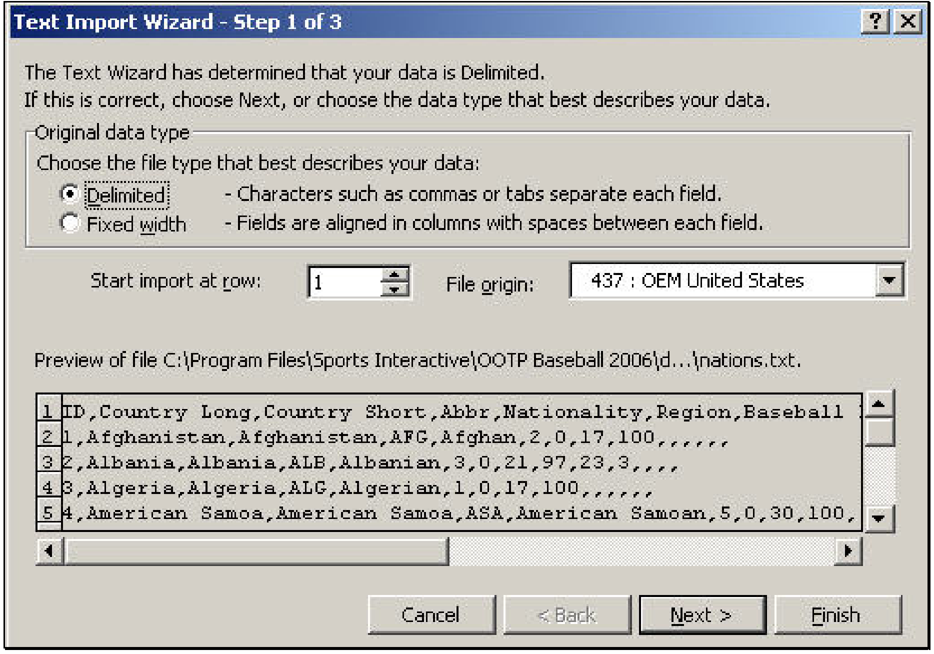
Open

Look in: database

OOTPB text files

6. Click the name of the file you wish to edit, and click Open. In this case, we will edit **nations.txt**.
7. You may receive an error saying, "Excel has detected that 'nations.txt' is a SYLK file, but cannot load it." This is fine; you can just click OK on this error message.
8. Excel will now begin the process of converting the file into Excel format using the Text Import Wizard:

Figure 10 Excel Text Import Wizard 1



The Text Wizard has determined that your data is Delimited.
If this is correct, choose Next, or choose the data type that best describes your data.

Original data type
Choose the file type that best describes your data:

● Delimited - Characters such as commas or tabs separate each field.
○ Fixed width - Fields are aligned in columns with spaces between each field.

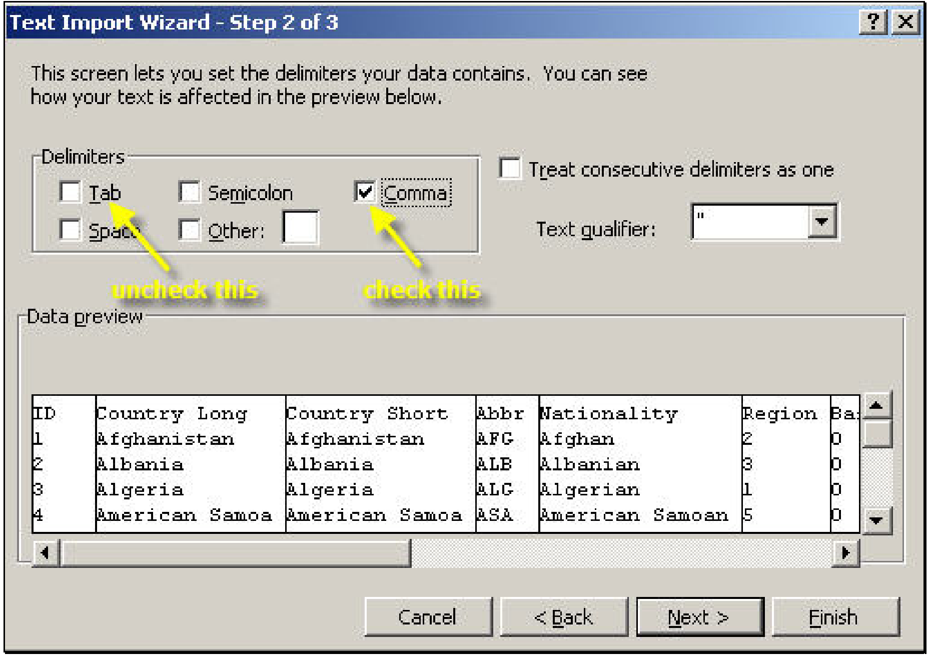
Start import at row: 1 File origin: 437 : OEM United States

Preview of file C:\Program Files\Sports Interactive\OOTP Baseball 2006\d...\nations.txt.

```
1 ID,Country Long,Country Short,Abbr,Nationality,Region,Baseball 
2 1,Afghanistan,Afghanistan,AFG,Afghan,2,0,17,100,,,,,,
3 2,Albania,Albania,ALB,Albanian,3,0,21,97,23,3,,,,
4 3,Algeria,Algeria,ALG,Algerian,1,0,17,100,,,,,,
5 4,American Samoa,American Samoa,ASA,American Samoan,5,0,30,100,
```

9. Click Next on this screen without changing any of the default values.
10. On the second page of the wizard, you must uncheck the Tab checkbox and check the Comma checkbox. This is to tell Excel that we're importing a comma-delimited file.

Figure 11 Excel Text Import Wizard 2



This screen lets you set the delimiters your data contains. You can see how your text is affected in the preview below.

Delimiters
□ Tab □ Semicolon ☑ Comma
□ Space □ Other:

□ Treat consecutive delimiters as one

Text qualifier: "

Data preview

ID	Country Long	Country Short	Abbr	Nationality	Region	Ba
1	Afghanistan	Afghanistan	AFG	Afghan	2	0
2	Albania	Albania	ALB	Albanian	3	0
3	Algeria	Algeria	ALG	Algerian	1	0
4	American Samoa	American Samoa	ASA	American Samoan	5	0

11. Once you have changed the checkboxes, click Finish. There is a third step to the wizard, but we don't need to make any changes there.

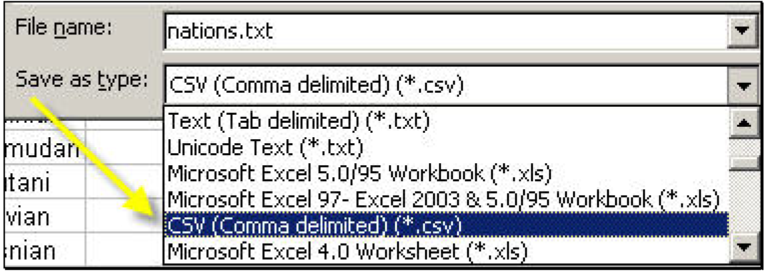
12. The file will now be open in Excel. Make any changes you wish, as necessary, and then follow the steps in the next section to get the file back into the format OOTPB needs.

11.2. Saving Files from Excel into OOTPB Required Format

Use these steps to save a file that you have edited in Microsoft Excel:

1. You must first have the edited file open in Microsoft Excel.
2. Select **File | Save As**.
3. From the "Save as Type" drop-down, select "CSV (Comma-delimited) (*.csv)":

Figure 12 Microsoft Excel Save as Type Box

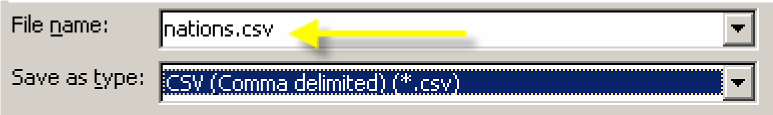


File name: nations.txt

Save as type: CSV (Comma delimited) (*.csv)

Text (Tab delimited) (*.txt)
Unicode Text (*.txt)
Microsoft Excel 5.0/95 Workbook (*.xls)
Microsoft Excel 97- Excel 2003 & 5.0/95 Workbook (*.xls)
CSV (Comma delimited) (*.csv)
Microsoft Excel 4.0 Worksheet (*.xls)
mudan
itani
vian
nian

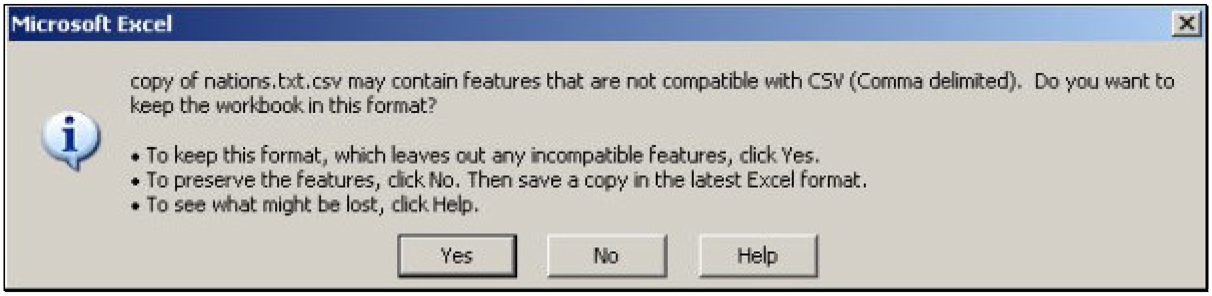
4. This will change the name of the file to nations.csv:



File name: nations.csv

Save as type: CSV (Comma delimited) (*.csv)

5. Click Save. You may see the following error message. Just click Yes:

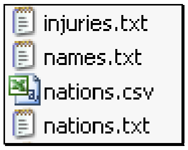


Microsoft Excel

copy of nations.txt.csv may contain features that are not compatible with CSV (Comma delimited). Do you want to keep the workbook in this format?

- To keep this format, which leaves out any incompatible features, click Yes.
- To preserve the features, click No. Then save a copy in the latest Excel format.
- To see what might be lost, click Help.

6. Close Microsoft Excel.
7. Open My Computer (or the Finder on a Mac) and navigate to the **database\data** directory. There should be two nations files now, nations.txt and nations.csv. *Note: If the file names do not display the file extension (.csv or .txt) after the file name, go to Tools|Folder Options. Click on the View tab and uncheck the box that says "Hide extensions for known file types." The extensions should now display.*



injuries.txt
names.txt
nations.csv
nations.txt

8. Rename nations.txt to nations.txt.old, or some other name.
9. Rename nations.csv to nations.txt.
10. Your changes will take effect the next time you load OOTPB.

11.3. Editing Cities.txt in Microsoft Excel

Cities.txt has too many rows of data to be edited in Excel. You can still accomplish this, however, by breaking the cities.txt file into two separate text files. Edit them as two separate Excel spreadsheets, and then merge the results back into a single .txt file.

12.0 Credits

Special thanks to everyone who contributed to this guide:

Contributor	Area
Steve Battisti	General writing and editing
Raidergoo	Names and nations editing instructions
Jdew	Graphics editing instructions
Endgame	Player photo instructions
Halos17	XML editor
Scefalu	Ballpark editing