



Database Structure

Printable View

Page 1 of 2 **1** 2 Last

Show 40 post(s) from this thread on one page

archibalduk

16-06-14, 08:43 PM

Database Structure

Could anybody point me in the direction of the database structure information? SI used to publish this information on each version of Champ Man, but I can't find the 01/02 structure anywhere (I've tried the forums here, the SI Games website and Google). It's usually in the form of .h files (C/C++ header files).

I also have a second question which I may as well ask here too: Has anybody ever attempted to write an editor that allows you to edit the database using spreadsheets (and thereby allowing lots of people to work on separate spreadsheets and import them into the database)? From what I could tell from the Downloads section, nobody has.

I have been a big CM fan since CM2 97/98 and used to while away hours on the game. Since the release of FM, I've played less and less because it didn't hold the same magic for me as CM2 and 3 and, to a lesser extent, CM4. Also, I became hopelessly addicted to Eastside Hockey Manager (I'm a big hockey fan too - and in fact have been running my own EHM website for the past decade). I've just picked up CM01/02 again and have started falling in love with it all over again. I have written a few DB and saved game editors and a start date patch for EHM and am thinking about an editor I might start on for CM01/02. It would basically be the same as the editor I'm continuing to develop for EHM where data can be imported and exported into the DB. We have been doing this with EHM for a few seasons now and it has made it so much easier to update the database (because lots of users can edit the DB via spreadsheets rather than relying on one person to do all of the work in a Pre-Game Editor). So that's why I'm asking about the DB structure and whether an editor like this already exists. If it doesn't, then I may start work on one.

P.s. I just want to say how incredible this site is and how in awe of the brilliant databases, patches, editors, etc people here have created.

djole2mcloud

16-06-14, 09:24 PM

there are some tools developed by John Locke, and it is possible to insert data from .csv files... now, update teams avoid this tools cause they can cause some troubles... i agree, they are not perfect, but... personally, i use it on every of my patches cause it allows me to insert huge amount of data, players, staff with full attributes within few minutes...

Dermotron

16-06-14, 09:24 PM

There is a basic, but fussy, version of an importer in Flex 2 but it lacks a small bit of refinement (positions only import in 20's, contracts show in a strange format). It's also a bit cumbersome. It will import csv files but it's as quick to add the players manually.

An import/export tool that allowed you to edit the exported data and re-import the same file would be very handy.

In terms of export there is Data Exporter in Downloads > Tools. Takes about 30-40 minutes to export the data from the March 2014 Data Update

That's from a purely Updating and DB maintenance POV. As djole says even the basic one is help for making patches.

djole2mcloud

16-06-14, 09:38 PM

i remember JL wanted to create new improved version of data inserter... resolving all strange issues with this version... it is a pity he never finished it... but i reckon he had some better things to do...

archibalduk

16-06-14, 09:50 PM

Ok, so it sounds like it might be of some use. I have all of the code in my EHM tool which I could fairly easily port over to Champ Man. All I need is the DB structure. Is there anywhere I can find this? It should look a bit like this: <http://www.ehmtheblueline.com/forums...t=8381#p131548>

Quote:

Originally Posted by **Dermotron**

In terms of export there is Data Exporter in Downloads > Tools. Takes about 30-40 minutes to export the data from the March 2014 Data Update

Do you mean 30 - 40 minutes to export the data into spreadsheets?! Maybe I've misunderstood, but if that's the case then that's extremely slow.

Dermotron

16-06-14, 10:04 PM

Yeah about that time. It's generally a 40mb text file so plenty of columns and about 120k rows

archibalduk

16-06-14, 10:14 PM

Quote:

*Originally Posted by **Dermotron** [↗](#)*

Yeah about that time. It's generally a 40mb text file so plenty of columns and about 120k rows

It'll depend on what is in the file and what processing is required in order to generate it. That said, we can extract the player histories from EHM (250k rows at around 100mb) in under a minute.

milo

16-06-14, 10:32 PM

in the bottom of your link i see "posted by graeme kelly", he is the one who made a save game editor for cm0102, i suppose that if what you want was possible he would have made it like he did for your hockey game...

archibalduk

16-06-14, 10:47 PM

Quote:

*Originally Posted by **milo** [↗](#)*

in the bottom of your link i see "posted by graeme kelly", he is the one who made a save game editor for cm0102, i suppose that if what you want was possible he would have made it like he did for your hockey game...

Yeah Graeme works for SI Games. He used to post the CM and EHM database structures each year. EHM was originally based on CM4 and so they have a lot of the DB structure in common - but they will still be very different (and even more different to the CM3 series).

I thought maybe with all the editors here that one of the authors would be able to share the structure (assuming of course that those authors still visit these forums and see this thread). In the mean time I've posted on the SI Forums. If that doesn't work then I'll drop Graeme a PM.

milo

16-06-14, 10:59 PM

i really hope that can be possible, i'm one of the updaters and i know how much work every release needs :D

silly question: in your link i also see the data are taken from .dat files, since we have those files, can't we extract them from there?

djole2mcloud

16-06-14, 11:24 PM

there was a thread long time ago...written code of a DB structure...i don't know what was the name of the thread,but inside was,i think,just right what u need... maybe some of admins can find that thread...

Jesus

16-06-14, 11:36 PM

Incredibly interesting thread this. A shining light in the darkness. Welcome to the forum Archi :wave:

Zé

16-06-14, 11:37 PM

Maybe it is my search skills but I cannot see that any structures were ever released? If you do have some it would be some valuable information for the relevant sub sections on the forum.

Sent from my RM-914_eu_euro1_327 using Tapataalk

djole2mcloud

16-06-14, 11:41 PM

well,there was a thread,maybe on old server...something about source code,champions league format...some of the creators of the cm0102 register himself on forum and posted a few posts about DB structure...based on that John Locke created some of his flex tools...

archibalduk

17-06-14, 07:38 AM

Quote:

*Originally Posted by **milo** [↗](#)*

silly question: in your link i also see the data are taken from .dat files, since we have those files, can't we extract them from there?

No, I'm afraid it's not that simple. The structure within each file will be different and wouldn't load in my EHM tool.

The structure was definitely released. I can see links on the SI Forums to it. Unfortunately all of the links are dead because I think the posts have been archived.

I'll try to get hold of Graeme Kelly. Perhaps he'll be able to dig out a copy.

milo

17-06-14, 10:16 AM

great :ok:

another silly question: can't you check, examine the extraction tool made by john locke via olly or something similar, to see if the code you are looking for it's somewhere there, or even fix the tool to make it work like your ehm one?

djole2mcloud

17-06-14, 11:00 AM

Can u update competition and staff histories via your tool?

Sent from my HUAWEI G510-0200 using Tapatalk

Craig Forrest

17-06-14, 01:18 PM

Hey an EHM fan.... original free EHM or the overly-complicated and buggy 2005/2007 versions? :P

I think I'm a member of the blueline too..... not very active though.... only a handful of posts if I remember correctly

Any help you can give in terms of your knowledge would be awesome :ok:

Go Flyers :D

Hugo9

17-06-14, 07:09 PM

Code:

```
//-----  
// Project: Championship Manager 3 Program (and subsequent updates)  
// Sports Interactive  
// Copyright © 1997. All Rights Reserved.  
//  
// SUBSYSTEM: Fix Application  
// FILE: database.h  
// AUTHOR: Marc Vaughan  
//  
// OVERVIEW  
// ~~~~~  
// Header file for the database access functions.  
//  
//-----  
  
// Ifdef wrapping to prevent multiple inclusions  
#ifndef DATABASE_H  
#define DATABASE_H  
  
//Added by Graeme Kelly  
typedef struct  
{  
short day; // days from Jan 1st.  
short year;  
long LeapYear; // =1 if year is a leapyear, 0 otherwise.  
} CM_DATE;  
  
// ----- //  
// Custom Include Files //  
// ----- //  
#include "date.h"  
#include "utils.h"  
#include "zipdir.h"  
#include "club.h"  
#include "langlib.h"  
  
// Define for initialising club rep table  
#define INITIALISE_TABLE TRUE  
  
// Invalid date def  
#define INVALID_DATE CM_DATE( 31, 1900 )  
  
// Languages  
#define DB_ALBANIAN 1  
#define DB_ARABIC 2  
#define DB_BULGARIAN 3  
#define DB_CHINESE 4  
#define DB_CROATIAN 5  
#define DB_CZECH 6  
#define DB_DUTCH 7  
#define DB_ENGLISH 8  
#define DB_FINNISH 9  
#define DB_FRENCH 10  
#define DB_GERMAN 11  
#define DB_GREEK 12  
#define DB_HEBREW 13  
#define DB_HUNGARIAN 14  
#define DB_ITALIAN 15
```

```

#define DB_JAPANESE 16
#define DB_KOREAN 17
#define DB_POLISH 18
#define DB_PORTUGUESE 19
#define DB_ROMANIAN 20
#define DB_RUSSIAN 21
#define DB_NORWEGIAN 22
#define DB_SERBIAN 23
#define DB_SPANISH 24
#define DB_TURKISH 25
#define DB_BURMESE 26
#define DB_MALAY 27
#define DB_PERSIAN 28
#define DB_SLOVAK 29
#define DB_SWEDISH 30
#define DB_ICELANDIC 31
#define DB_DANISH 32
#define DB_AFRIKAANS 33
#define DB_ALBANIAN 34
#define DB_AMHARIC 35
#define DB_ARMENIAN 36
#define DB_AZERI 37
#define DB_BANGLA 38
#define DB_ESTONIAN 39
#define DB_GEORGIAN 40
#define DB_HINDI 41
#define DB_JAVANESE 42
#define DB_KHMER 43
#define DB_KYRGYZ 44
#define DB_LAO 45
#define DB_LATVIAN 46
#define DB_LETBURGESCH 47
#define DB_LITHUANIAN 48
#define DB_MACEDONIAN 49
#define DB_MALAGASY 50
#define DB_MALTESE 51
#define DB_SINHALESE 52
#define DB_SLOVENIAN 53
#define DB_SOMALI 54
#define DB_SWAHILI 55
#define DB_TAJIK 56
#define DB_TAMIL 57
#define DB_THAI 58
#define DB_UZBEK 59
#define DB_VIETNAMESE 60

#define DB_NUMBER_OF_CM3_LANGUAGES 60

// Formation definitions
#define FORMATION_NOT_SET 0
#define FORMATION_532SW 1
#define FORMATION_532 2
#define FORMATION_442 3
#define FORMATION_433 4
#define FORMATION_424 5
#define FORMATION_352 6
#define FORMATION_41212 7
#define FORMATION_451 8
#define FORMATION_343 9

#define DB_NUMBER_OF_FORMATION_SHAPES 10

// NationGroupDefines
#define EEC_COUNTRY 2
#define NON_EEC_COUNTRY 1

// Selected leagues defs
#define NOT_SELECTED 0x0000
#define SELECTED_BACKGROUND 0x0001
#define SELECTED_FOREGROUND 0x0002
#define ALL_DIVISIONS_ACTIVE 0x0004

// Compatibility defs //
#define cm3_second_name_list cm3_second_names_list
#define CM3_PLAYER CM3_PLAYERS
#define CM3_NON_PLAYER CM3_NON_PLAYERS
#define CM3_CLUB_COMP CM3_CLUB_COMPS
#define cm3_club_tbl cm3_clubs_tbl
#define cm3_club_tbl_sz cm3_clubs_tbl_sz
#define cm3_player_tbl cm3_players_tbl
#define CM3_NAME CM3_NAMES
#define CENTER CENTRE
#define CM3_YOUTH_PLAYER CM3_YOUTH_PLAYERS

// -----//
// Constants //
// ----- //

#define INDEX_FILENAME "index.dat"

#define DB_CONTINENTS_VERSION 0x01
#define DB_NATIONS_VERSION 0x02
#define DB_CITIES_CORRUPT_ON_MAC 0x01
#define DB_CITIES_VERSION 0x02
#define DB_STADIUMS_VERSION 0x01
#define DB_CLUBS_VERSION 0x02
#define DB_STAFF_VERSION 0x02
#define DB_NON_PLAYERS_VERSION 0x02
#define DB_PLAYERS_VERSION 0x02
#define DB_STAFF_PREFERENCES_VERSION 0x01
#define DB_STAFF_HISTORY_VERSION 0x01
#define DB_CLUB_COMPS_VERSION 0x02
#define DB_CLUB_COMP_HISTORY_VERSION 0x01
#define DB_STAFF_COMPS_VERSION 0x02
#define DB_STAFF_COMP_HISTORY_VERSION 0x01
#define DB_OFFICIALS_VERSION 0x01
#define DB_COLOURS_VERSION 0x01

```

```

#define DB_NAMES_VERSION 0x01

// Primary colour definitions
#define PRIMARY_BLACK 1
#define PRIMARY_WHITE 2
#define PRIMARY_GREY 3
#define PRIMARY_YELLOW 4
#define PRIMARY_GREEN 5
#define PRIMARY_BLUE 6
#define PRIMARY_PURPLE 7
#define PRIMARY_RED 8
#define PRIMARY_ORANGE 9
#define PRIMARY_BROWN 10

// Condition level below which a player is considered unable to play
#define CONDITION_INJURED 60

// max number of friendly competitions
#define MAX_FRIENDLY_COMPETITIONS 127

// text lengths
#define LONG_TXT_LENGTH 101
#define STANDARD_TXT_LENGTH 51
#define SHORT_TXT_LENGTH 26

// club staff table sizes
#define DIRECTOR_SIZE 3
#define SQUAD_SIZE 50
#define SCOUT_SIZE 7
#define COACH_SIZE 5
#define PHYSIO_SIZE 3

// max number of human players
#define MAX_HUMAN_PLAYERS 16

// Maximum number of tactics in training at once
#define MAX_TACTIC_TRAINING 4

// Maximum size of a squad for a match
#define TEAM_SZ 20

// Ability defs
#define LOWER_TECHNICAL_ABILITY_LIMIT -125
#define UPPER_TECHNICAL_ABILITY_LIMIT 125
#define LOWER_NON_TECHNICAL_ABILITY_LIMIT 1
#define UPPER_NON_TECHNICAL_ABILITY_LIMIT 20

// ----- //
// Enum definitions //
// ----- //

// Home and away team definitions
#define HOME_TEAM 0
#define AWAY_TEAM 1

// nation region
#define REGION_AFRICA 1
#define REGION_ASIA 2
#define REGION_BALKAN 3
#define REGION_BRAZIL 4
#define REGION_BRITAIN 5
#define REGION_CARIBBEAN 6
#define REGION_CENTRAL_AMERICA 7
#define REGION_EASTERN_EUROPE 8
#define REGION_FRANCE 9
#define REGION_GERMANY 10
#define REGION_HOLLAND 11
#define REGION_ITALY 12
#define REGION_JAPAN 13
#define REGION_MIDDLE_EAST 14
#define REGION_NORTH_AFRICA 15
#define REGION_NORTH_AMERICA 16
#define REGION_NORTHERN_EUROPE 17
#define REGION_OCEANIA 18
#define REGION_RUSSIA 19
#define REGION_SCANDINAVIA 20
#define REGION_SOUTH_AFRICA 21
#define REGION_SOUTH_AMERICA 22
#define REGION_SOUTHERN_EUROPE 23
#define REGION_SPAIN 24
#define FIRST_REGION 1
#define LAST_REGION 24

// nation actual region
#define ACTUAL_REGION_AFRICA 1
#define ACTUAL_REGION_ASIA 2
#define ACTUAL_REGION_CARIBBEAN 3
#define ACTUAL_REGION_CENTRAL_AMERICA 4
#define ACTUAL_REGION_CENTRAL_EUROPE 5
#define ACTUAL_REGION_EASTERN_EUROPE 6
#define ACTUAL_REGION_MIDDLE_EAST 7
#define ACTUAL_REGION_NORTH_AFRICA 8
#define ACTUAL_REGION_NORTH_AMERICA 9
#define ACTUAL_REGION_OCEANIA 10
#define ACTUAL_REGION_SCANDINAVIA 11
#define ACTUAL_REGION_SOUTH_AMERICA 12
#define ACTUAL_REGION_SOUTHERN_EUROPE 13
#define ACTUAL_REGION_UK_AND_IRELAND 14
#define FIRST_ACTUAL_REGION 1
#define LAST_ACTUAL_REGION 14

// club professional status
#define PROFESSIONAL 1
#define SEMI_PRO 2
#define AMATEUR 3

// states of development

```

```

#define DEVELOPED_STATE 1
#define DEVELOPING_STATE 2
#define THIRD_WORLD_STATE 3

// database days of the week
#define DB_SUN 1
#define DB_MON 2
#define DB_TUE 3
#define DB_WED 4
#define DB_THUR 5
#define DB_FRI 6
#define DB_SAT 7

// staff job for club/nation
#define JOB_INVALID_JOB 0
#define JOB_CHAIRMAN 1
#define JOB_MANAGING_DIRECTOR 2
#define JOB_GENERAL_MANAGER 3
#define JOB_DIRECTOR_OF_FOOTBALL 4
#define JOB_MANAGER 5
#define JOB_ASSISTANT_MANAGER 6
#define JOB_RESERVE_TEAM_MANAGER 7
#define JOB_COACH 8
#define JOB_SCOUT 9
#define JOB_PHYSIO 10
#define JOB_PLAYER 11
#define JOB_PLAYER_MANAGER 12
#define JOB_PLAYER_ASSISTANT_MANAGER 13
#define PLAYER_RESERVE_TEAM_MANAGER 14
#define JOB_PLAYER_COACH 15
#define JOB_PLAYER_RETIRED 16
#define JOB_MEDIA_PUNDIT 17

// staff playing squad
#define INVALID_SQUAD 0
#define CLUB_SENIOR_SQUAD 0x01
#define CLUB_RESERVE_SQUAD 0x02
#define NATION_MAIN_SQUAD 0x04
#define NATION_B_SQUAD 0x08

// staff classification
#define INVALID_CLASSIFICATION 0
#define NON_PLAYER 1
#define PLAYER 2
#define NON_PLAYER_AND_PLAYER 3
#define YOUTH_PLAYER 4
#define NEW_HUMAN_MANAGER 5
#define SPARE_NON_PLAYER 6

// name tables
#define FIRSTNAME 1
#define SECONDNAME 2
#define COMMONNAME 3

// Player form definitions
#define VERY_POOR 0
#define POOR 1
#define AVERAGE 2
#define GOOD 3
#define VERY_GOOD 4
#define SUPERB 5

// Index file ids //
#define NUM_INDEX_TABLES 22
#define CLUB_TABLE 0
#define NATION_CLUBS_TABLE 1
#define COLOUR_TABLE 2
#define CONTINENT_TABLE 3
#define NATION_TABLE 4
#define STADIUM_TABLE 5
#define STAFF_TABLE 6
#define OFFICIALS_TABLE 7
#define YOUTH_PLAYER_TABLE 8 // Not used
#define NON_PLAYER_TABLE 9
#define PLAYER_TABLE 10
#define STAFF_COMP_TABLE 11
#define CLUB_COMP_TABLE 12
#define FIRST_NAME_TABLE 13
#define SECOND_NAME_TABLE 14
#define COMMON_NAME_TABLE 15
#define NATION_COMP_TABLE 16
#define STAFF_HISTORY_TABLE 17
#define STAFF_COMP_HISTORY_TABLE 18
#define CLUB_COMP_HISTORY_TABLE 19
#define NATION_COMP_HISTORY_TABLE 20
#define CITY_TABLE 21
#define STAFF_PREFERENCES_TABLE 22 // MUX with youth player table

#define test_for_bits_set( info, bits ) ( ( info & bits ) == bits )

// type definitions
#ifndef PLATFORM_MAC
typedef struct cm3_continents CM3_CONTINENTS;
typedef struct cm3_nations CM3_NATIONS;
typedef struct cm3_cities CM3_CITIES;
typedef struct cm3_stadiums CM3_STADIUMS;
typedef struct cm3_clubs CM3_CLUBS;
typedef struct cm3_staff CM3_STAFF;
typedef struct cm3_non_players CM3_NON_PLAYERS;
typedef struct cm3_players CM3_PLAYERS;
typedef struct cm3_staff_preferences CM3_STAFF_PREFERENCES;
typedef struct cm3_staff_history CM3_STAFF_HISTORY;
typedef struct cm3_club_comps CM3_CLUB_COMPS;
typedef struct cm3_club_comp_history CM3_CLUB_COMP_HISTORY;
typedef struct cm3_staff_comps CM3_STAFF_COMPS;
typedef struct cm3_staff_comp_history CM3_STAFF_COMP_HISTORY;
typedef struct cm3_officials CM3_OFFICIALS;

```

```

typedef struct cm3_colours CM3_COLOURS;
typedef struct cm3_names CM3_NAMES;
#else
typedef struct mac_cm3_continents CM3_CONTINENTS;
typedef struct mac_cm3_nations CM3_NATIONS;
typedef struct mac_cm3_cities CM3_CITIES;
typedef struct mac_cm3_stadiums CM3_STADIUMS;
typedef struct mac_cm3_clubs CM3_CLUBS;
typedef struct mac_cm3_staff CM3_STAFF;
typedef struct mac_cm3_non_players CM3_NON_PLAYERS;
typedef struct mac_cm3_players CM3_PLAYERS;
typedef struct mac_cm3_staff_preferences CM3_STAFF_PREFERENCES;
typedef struct mac_cm3_staff_history CM3_STAFF_HISTORY;
typedef struct mac_cm3_club_comps CM3_CLUB_COMPS;
typedef struct mac_cm3_club_comp_history CM3_CLUB_COMP_HISTORY;
typedef struct mac_cm3_staff_comps CM3_STAFF_COMPS;
typedef struct mac_cm3_staff_comp_history CM3_STAFF_COMP_HISTORY;
typedef struct mac_cm3_officials CM3_OFFICIALS;
typedef struct mac_cm3_colours CM3_COLOURS;
typedef struct mac_cm3_names CM3_NAMES;

#include "mac_database.h"
#endif

// ----- //
// Continents //
// ----- //

// This structure contains information about a nations name list beginning and
// end within each of the various names tables.
typedef struct
{
    long first_name_start_index; // Start index for this nations first names
    long second_name_start_index; // Start index for this nations second names
    long common_name_start_index; // Start index for this nations common names
    long number_first_names; // Total number of first names
    long number_second_names; // Total number of second names
    long number_common_names; // Total number of common names
    long first_name_count; // Count of different types of first names
    long second_name_count; // Count of different types of second names
    long common_name_count; // Count of different types of common names
} NATION_NAME_INDEX_ENTRY;

#ifdef DATABASE_CPP
NATION_NAME_INDEX_ENTRY *nation_name_index_tbl = (NATION_NAME_INDEX_ENTRY*)0;
// Size is the same as the cm3_nations_tbl_sz
#else
extern NATION_NAME_INDEX_ENTRY *nation_name_index_tbl;
#endif

// the following structures need to be byte aligned
#ifdef PLATFORM_MAC
#pragma options align=packed
#endif

// ----- //
// Continents //
// ----- //

struct cm3_continents
{
    // original data
    long ContinentID;
    char ContinentName[ SHORT_TXT_LENGTH ];
    char ContinentGenderName;
    char ContinentNameThreeLetter[ 4 ];
    char ContinentNameContinentality[ SHORT_TXT_LENGTH ];
    char ContinentFederationName[ LONG_TXT_LENGTH ];
    char ContinentGenderFederationName;
    char ContinentFederationNameShort[ SHORT_TXT_LENGTH ];
    char ContinentGenderFederationNameShort;
    SI_DOUBLE ContinentRegionalStrength;
};

// ----- //
// Nations //
// ----- //

struct cm3_nations
{
    // original data
    long NationID;
    char NationName[ STANDARD_TXT_LENGTH ];
    char NationGenderName;
    char NationNameShort[ SHORT_TXT_LENGTH ];
    char NationGenderNameShort;
    char NationNameThreeLetter[ 4 ];
    char NationNameNationality[ SHORT_TXT_LENGTH ];
    CM3_CONTINENTS *NationContinent;
    char NationRegion;
    char NationActualRegion;
    char NationFirstLanguage;
    char NationSecondLanguage;
    char NationThirdLanguage;
    CM3_CITIES *NationCapitalCity;
    char NationStateOfDevelopment;
    char NationGroupMembership;
    CM3_STADIUMS *NationNationalStadium;
    char NationGameImportance;
    char NationLeagueStandard;
    short NationNumberClubs;
    long NationNumberStaff; // Version 0x02 - Added
    short NationSeasonUpdateDay; // Version 0x02 - Added
    short NationReputation; // Version 0x02 - Changed char->short
    CM3_COLOURS *NationForegroundColour1;

```

```

CM3_COLOURS *NationBackgroundColour1;
CM3_COLOURS *NationForegroundColour2;
CM3_COLOURS *NationBackgroundColour2;
CM3_COLOURS *NationForegroundColour3;
CM3_COLOURS *NationBackgroundColour3;
SI_DOUBLE NationFIFACoefficient;
SI_DOUBLE NationFIFACoefficient91;
SI_DOUBLE NationFIFACoefficient92;
SI_DOUBLE NationFIFACoefficient93;
SI_DOUBLE NationFIFACoefficient94;
SI_DOUBLE NationFIFACoefficient95;
SI_DOUBLE NationFIFACoefficient96;
SI_DOUBLE NationUEFACoefficient91;
SI_DOUBLE NationUEFACoefficient92;
SI_DOUBLE NationUEFACoefficient93;
SI_DOUBLE NationUEFACoefficient94;
SI_DOUBLE NationUEFACoefficient95;
SI_DOUBLE NationUEFACoefficient96;
CM3_NATIONS *NationRivals1;
CM3_NATIONS *NationRivals2;
CM3_NATIONS *NationRivals3;

// runtime data
char NationLeagueSelected;
long NationShortlistOffset; // Version 0x02 - Added
char NationGamesPlayed; // Version 0x02 - Moved to runtime
};

// WEATHER STRUCTURE DEFINITIONS //
// (Have Fun PJ) //
// This structure defines the current weather settings for an area.

#define SPECIAL_HAIL 1
#define SPECIAL_SLEET 2
#define SPECIAL_SNOW 3

typedef struct
{
char wind;
char precipitation;
char temperature;
char special;
} WEATHER_CONDITION;

// Wind definition structure, this structure should give the percentage scale
// upon which each of the various wind type will happen within a season.
// Eg. calm - 3 // 0 - 3 % means calm weather occurs
// breezy - 20
// gusty - 75
// strong - 90
// gale - 100
typedef struct
{
char calm;
char breezy;
char gusty;
char strong;
char gale;
} WIND;

// Precipitation weather structure, this structure gives the percentage chance
// of each of the various types of precipitations happening.
typedef struct
{
char dry;
char wet;
char drizzle;
char shower;
char down_pour;
} PRECIPITATION;

// Temperature weather structure, this structure gives the percentage chance
// of each of the various types of precipitations happening.
typedef struct
{
char freezing;
char cold;
char mild;
char fine;
char warm;
char hot;
char very_hot;
} TEMPERATURE;

// This structure defines a seasons weather conditions within CM3.
typedef struct
{
short WeatherSeasonStartDay;
WIND WeatherSeasonWind;
PRECIPITATION WeatherSeasonPrecipitation;
TEMPERATURE WeatherSeasonTemperature;
} WEATHER_SEASON;

// CITY_WEATHER : This structure defines the weather for a particular city.
// This information should be editable through the data editor
// (Thanks PJ)

// Season definitions ... for indexing into CityWeatherSeason
#define SPRING 0
#define SUMMER 1
#define AUTUMN 2
#define WINTER 3

typedef struct
{
long CityWeatherID; // ID of the city (auto generated by editor)

```



```

char CityWeatherName[ SHORT_TXT_LENGTH ]; // Name of the city which is modelled (editable)
WEATHER_SEASON CityWeatherSeason[ 4 ]; // Seasonal model for this city (editable)
WEATHER_CONDITION CityWeatherCondition; // Current Weather in this city (not editable)
} CITY_WEATHER;

// ----- //
// Cities //
// ----- //

struct cm3_cities
{
// original data
long CityID;
char CityName[ SHORT_TXT_LENGTH ];
char CityGenderName;
CM3_NATIONS *CityNation;
SI_DOUBLE CityLatitude;
SI_DOUBLE CityLongitude;
char CityAttraction;
long CityWeather;
// CITY_WEATHER *CityWeatherCity; // Weather city to use for this cities weather
};

// ----- //
// Stadiums //
// ----- //

struct cm3_stadiums
{
// original data
long StadiumID;
char StadiumName[ STANDARD_TXT_LENGTH ];
char StadiumGenderName;
CM3_CITIES *StadiumCity;
long StadiumCapacity;
long StadiumSeatingCapacity;
long StadiumExpansionCapacity;
CM3_STADIUMS *StadiumNearbyStadium;
char StadiumCovered;
char StadiumUnderSoilHeating;
};

// ----- //
// Clubs //
// ----- //

struct cm3_clubs
{
// original data
long ClubID;
char ClubName[ STANDARD_TXT_LENGTH ];
char ClubGenderName;
char ClubNameShort[ SHORT_TXT_LENGTH ];
char ClubGenderNameShort;
CM3_NATIONS *ClubNation;
CM3_CLUB_COMPS *ClubDivision;
CM3_CLUB_COMPS *ClubLastDivision;
char ClubLastPosition;
CM3_CLUB_COMPS *ClubReserveDivision;
char ClubProfessionalStatus;
long ClubCash;
CM3_STADIUMS *ClubStadium;
char ClubOwnStadium;
CM3_STADIUMS *ClubReserveStadium;
char ClubHomeMatchDay;
long ClubAttendance;
long ClubMinAttendance;
long ClubMaxAttendance;
char ClubTraining;
short ClubReputation; // Version 0x02 - Changed char->short
char ClubPLC;
CM3_COLOURS *ClubForegroundColour1;
CM3_COLOURS *ClubBackgroundColour1;
CM3_COLOURS *ClubForegroundColour2;
CM3_COLOURS *ClubBackgroundColour2;
CM3_COLOURS *ClubForegroundColour3;
CM3_COLOURS *ClubBackgroundColour3;
CM3_STAFF *ClubFavouriteStaff1;
CM3_STAFF *ClubFavouriteStaff2;
CM3_STAFF *ClubFavouriteStaff3;
CM3_STAFF *ClubDislikedStaff1;
CM3_STAFF *ClubDislikedStaff2;
CM3_STAFF *ClubDislikedStaff3;
CM3_CLUBS *ClubRivals1;
CM3_CLUBS *ClubRivals2;
CM3_CLUBS *ClubRivals3;
CM3_STAFF *ClubChairman;
CM3_STAFF *ClubDirectorList[ DIRECTOR_SIZE ];
CM3_STAFF *ClubManager;
CM3_STAFF *ClubAssistantManager;
CM3_STAFF *ClubSquadList[ SQUAD_SIZE ];
CM3_STAFF *ClubCoachList[ COACH_SIZE ];
CM3_STAFF *ClubScoutList[ SCOUT_SIZE ];
CM3_STAFF *ClubPhysioList[ PHYSIO_SIZE ];

// runtime data
long ClubEuroFlag;
char ClubEuroSeeding;
CM3_STAFF *ClubTeamSelected[ TEAM_SZ ];
long ClubTacticTraining[ MAX_TACTIC_TRAINING ];
long ClubTacticSelected;
char ClubHasLinkedClub;
};

```

```

// ----- //
// Staff //
// ----- //

struct cm3_staff
{
long StaffID;
char *StaffFirstName;
char *StaffSecondName;
char *StaffCommonName;
CM_DATE StaffDateOfBirth;
short StaffYearOfBirth;
CM3_NATIONS *StaffNation;
CM3_NATIONS *StaffSecondNation;
unsigned char StaffInternationalApps;
unsigned char StaffInternationalGoals;
CM3_CLUBS *StaffNationContracted;
char StaffJobForNation;
CM_DATE StaffDateJoinedNation;
CM_DATE StaffContractExpiresNation;
CM3_CLUBS *StaffClubContracted;
char StaffJobForClub;
CM_DATE StaffDateJoinedClub;
CM_DATE StaffContractExpiresClub;
long StaffEstimatedWage;
long StaffEstimatedValue;
char StaffAdaptability;
char StaffAmbition;
char StaffDetermination;
char StaffLoyalty;
char StaffPressure;
char StaffProfessionalism;
char StaffSportsmanship;
char StaffTemperament;
char StaffPlayingSquad;
char StaffClassification;
char StaffClubValuation;
CM3_PLAYERS *StaffPlayerData;
CM3_STAFF_PREFERENCES *StaffPreferences; // Version 0x02 - New ptr type
CM3_NON_PLAYERS *StaffNonPlayerData;

// Runtime data //
char StaffSquadSelectedFor; // should be StaffEuroSquadFlag - Kev
};

// ----- //
// Non Players //
// ----- //

struct cm3_non_players
{
long StaffNonPlayerID;
short StaffNonPlayerCurrentAbility;
short StaffNonPlayerPotentialAbility;
short StaffNonPlayerHomeReputation; // Version 0x02 - Changed char->short
short StaffNonPlayerCurrentReputation; // Version 0x02 - Changed char->short
short StaffNonPlayerWorldReputation; // Version 0x02 - Changed char->short
char StaffNonPlayerAttacking;
char StaffNonPlayerBusiness;
char StaffNonPlayerCoaching;
char StaffNonPlayerCoachingGks;
char StaffNonPlayerCoachingTechnique;
char StaffNonPlayerDirectness;
char StaffNonPlayerDiscipline;
char StaffNonPlayerFreeRoles;
char StaffNonPlayerInterference;
char StaffNonPlayerJudgement;
char StaffNonPlayerJudgingPotential;
char StaffNonPlayerManHandling;
char StaffNonPlayerMarking;
char StaffNonPlayerMotivating;
char StaffNonPlayerOffside;
char StaffNonPlayerPatience;
char StaffNonPlayerPhysiotherapy;
char StaffNonPlayerPressing;
char StaffNonPlayerResources;
char StaffNonPlayerTactics;
char StaffNonPlayerYoungsters;
CM3_STAFF *StaffNonPlayerGoalkeeper;
CM3_STAFF *StaffNonPlayerSweeper;
CM3_STAFF *StaffNonPlayerDefender;
CM3_STAFF *StaffNonPlayerDefensiveMidfielder;
CM3_STAFF *StaffNonPlayerMidfielder;
CM3_STAFF *StaffNonPlayerAttackingMidfielder;
CM3_STAFF *StaffNonPlayerAttacker;
CM3_STAFF *StaffNonPlayerWingBack;
char StaffNonPlayerFormationPreferred;
};

// ----- //
// Players //
// ----- //

struct cm3_players
{
// original data
long StaffPlayerID;
char StaffPlayerSquadNumber;
short StaffPlayerCurrentAbility;
short StaffPlayerPotentialAbility;
short StaffPlayerHomeReputation; // Version 0x02 - Changed char->short
short StaffPlayerCurrentReputation; // Version 0x02 - Changed char->short
short StaffPlayerWorldReputation; // Version 0x02 - Changed char->short
char StaffPlayerGoalkeeper;
char StaffPlayerSweeper;
char StaffPlayerDefender;
char StaffPlayerDefensiveMidfielder;

```

```

char StaffPlayerMidfielder;
char StaffPlayerAttackingMidfielder;
char StaffPlayerAttacker;
char StaffPlayerWingBack;
char StaffPlayerRightSide;
char StaffPlayerLeftSide;
char StaffPlayerCentral;
char StaffPlayerFreeRole;
char StaffPlayerAcceleration;
char StaffPlayerAggression;
char StaffPlayerAgility;
char StaffPlayerAnticipation;
char StaffPlayerBalance;
char StaffPlayerBravery; // Value
char StaffPlayerConsistency;
char StaffPlayerCorners;
char StaffPlayerCrossing;
char StaffPlayerDecisions;
char StaffPlayerDirtiness;
char StaffPlayerDribbling;
char StaffPlayerFinishing;
char StaffPlayerFlair;
char StaffPlayerFreeKicks;
char StaffPlayerHandling;
char StaffPlayerHeading;
char StaffPlayerImportantMatches;
char StaffPlayerInjuryProneness;
char StaffPlayerJumping;
char StaffPlayerLeadership;
char StaffPlayerLeftFoot;
char StaffPlayerLongShots;
char StaffPlayerMarking;
char StaffPlayerMovement;
char StaffPlayerNaturalFitness;
char StaffPlayerOneOnOnes;
char StaffPlayerPace;
char StaffPlayerPassing;
char StaffPlayerPenalties;
char StaffPlayerPositioning;
char StaffPlayerReflexes;
char StaffPlayerRightFoot;
char StaffPlayerStamina;
char StaffPlayerStrength;
char StaffPlayerTackling;
char StaffPlayerTeamwork;
char StaffPlayerTechnique;
char StaffPlayerThrowins;
char StaffPlayerVersatility;
char StaffPlayerVision;
char StaffPlayerWorkRate;

// runtime data
char StaffPlayerMorale;
};

// ----- //
// Player details //
// ----- //

struct cm3_staff_preferences
{
    long StaffPreferencesID;
    CM3_CLUBS *StaffFavouriteClubs1;
    CM3_CLUBS *StaffFavouriteClubs2;
    CM3_CLUBS *StaffFavouriteClubs3;
    CM3_CLUBS *StaffDislikedClubs1;
    CM3_CLUBS *StaffDislikedClubs2;
    CM3_CLUBS *StaffDislikedClubs3;
    CM3_STAFF *StaffFavouriteStaff1;
    CM3_STAFF *StaffFavouriteStaff2;
    CM3_STAFF *StaffFavouriteStaff3;
    CM3_STAFF *StaffDislikedStaff1;
    CM3_STAFF *StaffDislikedStaff2;
    CM3_STAFF *StaffDislikedStaff3;
};

// ----- //
// Staff History //
// ----- //

struct cm3_staff_history
{
    long StaffHistoryID;
    CM3_STAFF *StaffHistoryStaff;
    short StaffHistoryYear;
    CM3_CLUBS *StaffHistoryClub;
    char StaffHistoryOnLoan;
    unsigned char StaffHistoryApps;
    unsigned char StaffHistoryGoals;
};

// ----- //
// Club Competitions //
// ----- //

struct cm3_club_comps
{
    long ClubCompID;
    char ClubCompName[ STANDARD_TXT_LENGTH ];
    char ClubCompGenderName;
    char ClubCompNameShort[ SHORT_TXT_LENGTH ];
    char ClubCompGenderNameShort;
    char ClubCompNameThreeLetter[ 4 ];
    char ClubCompScope;
    char ClubCompSelected;
    CM3_CONTINENTS *ClubCompContinent;
    CM3_NATIONS *ClubCompNation;
};

```

```

CM3_COLOURS *ClubCompForegroundColour;
CM3_COLOURS *ClubCompBackgroundColour;
short ClubCompReputation; // Version 0x02 - Changed char->short
};

// ----- //
// Club Competition History //
// ----- //

struct cm3_club_comp_history
{
long ClubCompHistoryID;
CM3_CLUB_COMPS *ClubCompHistoryClubComp;
short ClubCompHistoryYear;
CM3_CLUBS *ClubCompHistoryWinners;
CM3_CLUBS *ClubCompHistoryRunnersUp;
CM3_CLUBS *ClubCompHistoryThirdPlaced;
CM3_CLUBS *ClubCompHistoryHosts;
};

// ----- //
// Staff Competitions //
// ----- //

struct cm3_staff_comps
{
long StaffCompID;
char StaffCompName[ STANDARD_TXT_LENGTH ];
char StaffCompGenderName;
char StaffCompNameShort[ SHORT_TXT_LENGTH ];
char StaffCompGenderNameShort;
CM3_CONTINENTS *StaffCompContinent;
CM3_NATIONS *StaffCompNation;
CM3_COLOURS *StaffCompForegroundColour;
CM3_COLOURS *StaffCompBackgroundColour;
short StaffCompReputation; // Version 0x02 - Changed char->short
};

// ----- //
// Staff Competition History //
// ----- //

struct cm3_staff_comp_history
{
long StaffCompHistoryID;
CM3_STAFF_COMPS *StaffCompHistoryStaffComp;
short StaffCompHistoryYear;
char *StaffCompHistoryFirstPlacedFirstName;
char *StaffCompHistoryFirstPlacedSecondName;
CM3_STAFF *StaffCompHistoryFirstPlacedIndex;
long StaffCompHistoryFirstPlacedInfo;
char *StaffCompHistorySecondPlacedFirstName;
char *StaffCompHistorySecondPlacedSecondName;
CM3_STAFF *StaffCompHistorySecondPlacedIndex;
long StaffCompHistorySecondPlacedInfo;
char *StaffCompHistoryThirdPlacedFirstName;
char *StaffCompHistoryThirdPlacedSecondName;
CM3_STAFF *StaffCompHistoryThirdPlacedIndex;
long StaffCompHistoryThirdPlacedInfo;
};

// ----- //
// Officials //
// ----- //

struct cm3_officials
{
long OfficialID;
char *OfficialFirstName;
char *OfficialSecondName;
CM_DATE OfficialDateOfBirth;
short OfficialYearOfBirth;
CM3_NATIONS *OfficialNation;
CM3_CITIES *OfficialCity;
short OfficialCurrentAbility;
short OfficialPotentialAbility;
short OfficialReputation;
char OfficialAllowingFlow;
char OfficialDiscipline;
char OfficialImportantMatches;
char OfficialPressure;
char OfficialRefereeing;
char OfficialRunningLine;
char OfficialTimekeeping;
};

// ----- //
// Colours //
// ----- //

struct cm3_colours
{
long ColourID;
char ColourName[ STANDARD_TXT_LENGTH ];
unsigned char ColourRedIntensity;
unsigned char ColourGreenIntensity;
unsigned char ColourBlueIntensity;
};

// ----- //
// Names //
// ----- //

// The cm3_names structure contains the information upon a name which is used
// by either the staff or officials within championship manager 3.
struct cm3_names
{

```

```

char Name[ STANDARD_TXT_LENGTH ];
long NameID;
long Nation;
char count;
};

// reset alignment
#ifdef PLATFORM_MAC
#pragma options align=reset
#endif

// ----- //
// Variables //
// ----- //

#define MAX_INDEX_NAME 50

// Index file struct //
typedef struct
{
char filename[ MAX_INDEX_NAME + 1 ];
long file_id;
long table_sz;
long offset;
long version;
} INDEX_TABLE;

#ifdef DATABASE_CPP
INDEX_TABLE index_info[ NUM_INDEX_TABLES ];

static BOOL game_format_data = FALSE;
static long num_of_friendlies = 0,
num_of_humans = 0;

// Table sizes //
long cm3_continents_tbl_sz = 0;
long cm3_nations_tbl_sz = 0;
long cm3_cities_tbl_sz = 0;
long cm3_stadiums_tbl_sz = 0;
long cm3_clubs_tbl_sz = 0;
long cm3_nation_clubs_tbl_sz = 0;
long cm3_staff_tbl_sz = 0;
long cm3_non_players_tbl_sz = 0;
long cm3_players_tbl_sz = 0;
long cm3_staff_preferences_tbl_sz = 0;
long cm3_staff_history_tbl_sz = 0;
long cm3_club_comps_tbl_sz = 0;
long cm3_club_comp_history_tbl_sz = 0;
long cm3_nation_club_comps_tbl_sz = 0;
long cm3_nation_club_comp_history_tbl_sz = 0;
long cm3_staff_comps_tbl_sz = 0;
long cm3_staff_comp_history_tbl_sz = 0;
long cm3_officials_tbl_sz = 0;
long cm3_colours_tbl_sz = 0;
long cm3_first_names_list_sz = 0;
long cm3_second_names_list_sz = 0;
long cm3_common_names_list_sz = 0;

char use_real_players = 1;

// Table ptrs //
CM3_CONTINENTS *cm3_continents_tbl = NULL;
CM3_NATIONS *cm3_nations_tbl = NULL;
CM3_CITIES *cm3_cities_tbl = NULL;
CM3_STADIUMS *cm3_stadiums_tbl = NULL;
CM3_CLUBS *cm3_clubs_tbl = NULL;
CM3_CLUBS *cm3_nation_clubs_tbl = NULL;
CM3_STAFF *cm3_staff_tbl = NULL;
CM3_NON_PLAYERS *cm3_non_players_tbl = NULL;
CM3_PLAYERS *cm3_players_tbl = NULL;
CM3_STAFF_PREFERENCES *cm3_staff_preferences_tbl = NULL;
CM3_STAFF_HISTORY *cm3_staff_history_tbl = NULL;
CM3_CLUB_COMPS *cm3_club_comps_tbl = NULL;
CM3_CLUB_COMP_HISTORY *cm3_club_comp_history_tbl = NULL;
CM3_CLUB_COMPS *cm3_nation_club_comps_tbl = NULL;
CM3_CLUB_COMP_HISTORY *cm3_nation_club_comp_history_tbl = NULL;
CM3_STAFF_COMPS *cm3_staff_comps_tbl = NULL;
CM3_STAFF_COMP_HISTORY *cm3_staff_comp_history_tbl = NULL;
CM3_OFFICIALS *cm3_officials_tbl = NULL;
CM3_COLOURS *cm3_colours_tbl = NULL;
CM3_NAMES *cm3_first_names_list = NULL;
CM3_NAMES *cm3_second_names_list = NULL;
CM3_NAMES *cm3_common_names_list = NULL;
#else
// Table sizes //
extern long cm3_continents_tbl_sz;
extern long cm3_nations_tbl_sz;
extern long cm3_cities_tbl_sz;
extern long cm3_stadiums_tbl_sz;
extern long cm3_clubs_tbl_sz;
extern long cm3_nation_clubs_tbl_sz;
extern long cm3_staff_tbl_sz;
extern long cm3_non_players_tbl_sz;
extern long cm3_players_tbl_sz;
extern long cm3_staff_preferences_tbl_sz;
extern long cm3_staff_history_tbl_sz;
extern long cm3_club_comps_tbl_sz;
extern long cm3_club_comp_history_tbl_sz;
extern long cm3_nation_club_comps_tbl_sz;
extern long cm3_nation_club_comp_history_tbl_sz;
extern long cm3_staff_comps_tbl_sz;
extern long cm3_staff_comp_history_tbl_sz;
extern long cm3_officials_tbl_sz;
extern long cm3_colours_tbl_sz;
extern long cm3_first_names_list_sz;
extern long cm3_second_names_list_sz;
extern long cm3_common_names_list_sz;

```

```

// Table sizes //
extern CM3_CONTINENTS *cm3_continents_tbl;
extern CM3_NATIONS *cm3_nations_tbl;
extern CM3_CITIES *cm3_cities_tbl;
extern CM3_STADIUMS *cm3_stadiums_tbl;
extern CM3_CLUBS *cm3_clubs_tbl;
extern CM3_CLUBS *cm3_nation_clubs_tbl;
extern CM3_STAFF *cm3_staff_tbl;
extern CM3_NON_PLAYERS *cm3_non_players_tbl;
extern CM3_PLAYERS *cm3_players_tbl;
extern CM3_STAFF_PREFERENCES *cm3_staff_preferences_tbl;
extern CM3_STAFF_HISTORY *cm3_staff_history_tbl;
extern CM3_CLUB_COMPS *cm3_club_comps_tbl;
extern CM3_CLUB_COMP_HISTORY *cm3_club_comp_history_tbl;
extern CM3_CLUB_COMPS *cm3_nation_club_comps_tbl;
extern CM3_CLUB_COMP_HISTORY *cm3_nation_club_comp_history_tbl;
extern CM3_STAFF_COMPS *cm3_staff_comps_tbl;
extern CM3_STAFF_COMP_HISTORY *cm3_staff_comp_history_tbl;
extern CM3_OFFICIALS *cm3_officials_tbl;
extern CM3_COLOURS *cm3_colours_tbl;
extern CM3_NAMES *cm3_first_names_list;
extern CM3_NAMES *cm3_second_names_list;
extern CM3_NAMES *cm3_common_names_list;

extern char use_real_players;
#endif

// Language header structure ..
typedef struct
{
long club_tbl_sz;
long continent_tbl_sz;
long nation_comp_tbl_sz;
long nation_tbl_sz;
long cities_tbl_sz;
long club_comp_tbl_sz;
long stadiums_tbl_sz;
long staff_comp_tbl_sz;
} LANG_HEADER;

// Structure containing club language information
typedef struct
{
char ClubName[ STANDARD_TXT_LENGTH ];
char GenderClubName;
char ClubNameShort[ SHORT_TXT_LENGTH ];
char GenderClubNameShort;
} CLUB_LANG;

// Structure containing continent language information
typedef struct
{
char ContinentName[ SHORT_TXT_LENGTH ];
char GenderContinentName;
char ContinentNameThreeLetter[ 4 ];
char ContinentNameNationality[ SHORT_TXT_LENGTH ];
char ContinentFederationName[ LONG_TXT_LENGTH ];
char GenderContinentFederationName;
char ContinentFederationNameShort[ SHORT_TXT_LENGTH ];
char GenderContinentFederationNameShort;
} CONTINENT_LANG;

// Structure containing nation competition language information
typedef struct
{
char NationCompName[ STANDARD_TXT_LENGTH ];
char GenderNationCompName;
char NationCompNameShort[ SHORT_TXT_LENGTH ];
char GenderNationCompNameShort;
char NationCompNameThreeLetter[ 4 ];
} NATION_COMP_LANG;

// Structure containing nation language information
typedef struct
{
char NationName[ STANDARD_TXT_LENGTH ];
char GenderNationName;
char NationNameShort[ SHORT_TXT_LENGTH ];
char GenderNationNameShort;
char NationNameThreeLetter[ 4 ];
char NationNameNationality[ SHORT_TXT_LENGTH ];
} NATION_LANG;

// Structure containing city language information
typedef struct
{
char CityName[ STANDARD_TXT_LENGTH ];
char GenderCityName;
} CITY_LANG;

// Structure containing stadium language information
typedef struct
{
char StadiumName[ STANDARD_TXT_LENGTH ];
char GenderStadiumName;
} STADIUM_LANG;

// Structure containing club comp language information
typedef struct
{
char ClubCompName[ STANDARD_TXT_LENGTH ];
char GenderClubCompName;
char ClubCompNameShort[ SHORT_TXT_LENGTH ];
char GenderClubCompNameShort;
char ClubCompNameThreeLetter[ 4 ];
} CLUB_COMP_LANG;

```

```

// Structure containing staff language information
typedef struct
{
char StaffCompName[ STANDARD_TXT_LENGTH ];
char GenderStaffCompName;
char StaffCompNameShort[ SHORT_TXT_LENGTH ];
char GenderStaffCompNameShort;
} STAFF_COMP_LANG;

// Class containing English language text for database
class ENGLISH_LANG
{
public:

ENGLISH_LANG( void );
~ENGLISH_LANG( void );

// access functions
char *get_nation_name( CM3_NATIONS *p_nation );
char *get_nation_name_three_letter( CM3_NATIONS *p_nation );
char *get_club_comp_name( CM3_CLUB_COMPS *p_club_comp );
char *get_continent_name( CM3_CONTINENTS *p_continent );
char *get_staff_comp_name( CM3_STAFF_COMPS *p_staff_comp );
char *get_club_name( CM3_CLUBS *p_club );
char *get_club_name_short( CM3_CLUBS *p_club );
char *get_stadium_name( CM3_STADIUMS *p_stadium );

private:
BOOL read_english_database( void );

LANG_HEADER lang_header;
CLUB_LANG *club_lang_data;
CONTINENT_LANG *continent_lang_data;
NATION_COMP_LANG *nation_comp_lang_data;
NATION_LANG *nation_lang_data;
CITY_LANG *city_lang_data;
STADIUM_LANG *stadium_lang_data;
CLUB_COMP_LANG *club_comp_lang_data;
STAFF_COMP_LANG *staff_comp_lang_data;
};

#ifdef DATABASE_CPP
LANGUAGE current_database_language = NO_LANGUAGE_SET;
long *club_remapping_indices = NULL;
#else
extern LANGUAGE current_database_language;
extern long *club_remapping_indices;
#endif

```

i knew it,i saw it somewhere... bravo Hugo9!

Quote:

*Originally Posted by **Hugo9***

Code:

```

//-----
// Project: Championship Manager 3 Program (and subsequent updates)
// Sports Interactive
// Copyright © 1997. All Rights Reserved.
//
// SUBSYSTEM: Fix Application
// FILE: database.h
// AUTHOR: Marc Vaughan
//
// OVERVIEW
// ~~~~~
// Header file for the database access functions.
//
//-----

// Ifdef wrapping to prevent multiple inclusions
#ifndef DATABASE_H
#define DATABASE_H

//Added by Graeme Kelly
typedef struct
{
short day; // days from Jan 1st.
short year;
long LeapYear; // =1 if year is a leapyear, 0 otherwise.
} CM_DATE;

// ----- //
// Custom Include Files //
// ----- //
#include "date.h"
#include "utils.h"
#include "zipdir.h"
#include "club.h"
#include "langlib.h"

```

```

// Define for initialising club rep table
#define INITIALISE_TABLE TRUE

// Invalid date def
#define INVALID_DATE CM_DATE( 31, 1900 )

// Languages
#define DB_ALBANIAN 1
#define DB_ARABIC 2
#define DB_BULGARIAN 3
#define DB_CHINESE 4
#define DB_CROATIAN 5
#define DB_CZECH 6
#define DB_DUTCH 7
#define DB_ENGLISH 8
#define DB_FINNISH 9
#define DB_FRENCH 10
#define DB_GERMAN 11
#define DB_GREEK 12
#define DB_HEBREW 13
#define DB_HUNGARIAN 14
#define DB_ITALIAN 15
#define DB_JAPANESE 16
#define DB_KOREAN 17
#define DB_POLISH 18
#define DB_PORTUGUESE 19
#define DB_ROMANIAN 20
#define DB_RUSSIAN 21
#define DB_NORWEGIAN 22
#define DB_SERBIAN 23
#define DB_SPANISH 24
#define DB_TURKISH 25
#define DB_BURMESE 26
#define DB_MALAY 27
#define DB_PERSIAN 28
#define DB_SLOVAK 29
#define DB_SWEDISH 30
#define DB_ICELANDIC 31
#define DB_DANISH 32
#define DB_AFIKAANS 33
#define DB_ALBANIAN 34
#define DB_AMHARIC 35
#define DB_ARMENIAN 36
#define DB_AZERI 37
#define DB_BANGLA 38
#define DB_ESTONIAN 39
#define DB_GEORGIAN 40
#define DB_HINDI 41
#define DB_JAVANESE 42
#define DB_KHMER 43
#define DB_KYRGYZ 44
#define DB_LAO 45
#define DB_LATVIAN 46
#define DB_LETBURGESH 47
#define DB_LITHUANIAN 48
#define DB_MACEDONIAN 49
#define DB_MALAGASY 50
#define DB_MALTESE 51
#define DB_SINHALESE 52
#define DB_SLOVENIAN 53
#define DB_SOMALI 54
#define DB_SWAHILI 55
#define DB_TAJIK 56
#define DB_TAMIL 57
#define DB_THAI 58
#define DB_UZBEK 59
#define DB_VIETNAMESE 60

#define DB_NUMBER_OF_CM3_LANGUAGES 60

// Formation definitions
#define FORMATION_NOT_SET 0
#define FORMATION_532SW 1
#define FORMATION_532 2
#define FORMATION_442 3
#define FORMATION_433 4
#define FORMATION_424 5
#define FORMATION_352 6
#define FORMATION_41212 7
#define FORMATION_451 8
#define FORMATION_343 9

#define DB_NUMBER_OF_FORMATION_SHAPES 10

// NationGroupDefines
#define EEC_COUNTRY 2
#define NON_EEC_COUNTRY 1

// Selected leagues defs
#define NOT_SELECTED 0x0000
#define SELECTED_BACKGROUND 0x0001
#define SELECTED_FOREGROUND 0x0002
#define ALL_DIVISIONS_ACTIVE 0x0004

```



```

// Compatability defs //
#define cm3_second_name_list cm3_second_names_list
#define CM3_PLAYER CM3_PLAYERS
#define CM3_NON_PLAYER CM3_NON_PLAYERS
#define CM3_CLUB_COMP CM3_CLUB_COMPS
#define cm3_club_tbl cm3_clubs_tbl
#define cm3_club_tbl_sz cm3_clubs_tbl_sz
#define cm3_player_tbl cm3_players_tbl
#define CM3_NAME CM3_NAMES
#define CENTER CENTRE
#define CM3_YOUTH_PLAYER CM3_YOUTH_PLAYERS

// -----//
// Constants //
// ----- //

#define INDEX_FILENAME "index.dat"

#define DB_CONTINENTS_VERSION 0x01
#define DB_NATIONS_VERSION 0x02
#define DB_CITIES_CORRUPT_ON_MAC 0x01
#define DB_CITIES_VERSION 0x02
#define DB_STADIUMS_VERSION 0x01
#define DB_CLUBS_VERSION 0x02
#define DB_STAFF_VERSION 0x02
#define DB_NON_PLAYERS_VERSION 0x02
#define DB_PLAYERS_VERSION 0x02
#define DB_STAFF_PREFERENCES_VERSION 0x01
#define DB_STAFF_HISTORY_VERSION 0x01
#define DB_CLUB_COMPS_VERSION 0x02
#define DB_CLUB_COMP_HISTORY_VERSION 0x01
#define DB_STAFF_COMPS_VERSION 0x02
#define DB_STAFF_COMP_HISTORY_VERSION 0x01
#define DB_OFFICIALS_VERSION 0x01
#define DB_COLOURS_VERSION 0x01
#define DB_NAMES_VERSION 0x01

// Primary colour definitions
#define PRIMARY_BLACK 1
#define PRIMARY_WHITE 2
#define PRIMARY_GREY 3
#define PRIMARY_YELLOW 4
#define PRIMARY_GREEN 5
#define PRIMARY_BLUE 6
#define PRIMARY_PURPLE 7
#define PRIMARY_RED 8
#define PRIMARY_ORANGE 9
#define PRIMARY_BROWN 10

// Condition level below which a player is considered unable to play
#define CONDITION_INJURED 60

// max number of friendly competitions
#define MAX_FRIENDLY_COMPETITIONS 127

// text lengths
#define LONG_TXT_LENGTH 101
#define STANDARD_TXT_LENGTH 51
#define SHORT_TXT_LENGTH 26

// club staff table sizes
#define DIRECTOR_SIZE 3
#define SQUAD_SIZE 50
#define SCOUT_SIZE 7
#define COACH_SIZE 5
#define PHYSIO_SIZE 3

// max number of human players
#define MAX_HUMAN_PLAYERS 16

// Maximum number of tactics in training at once
#define MAX_TACTIC_TRAINING 4

// Maximum size of a squad for a match
#define TEAM_SZ 20

// Ability defs
#define LOWER_TECHNICAL_ABILITY_LIMIT -125
#define UPPER_TECHNICAL_ABILITY_LIMIT 125
#define LOWER_NON_TECHNICAL_ABILITY_LIMIT 1
#define UPPER_NON_TECHNICAL_ABILITY_LIMIT 20

// ----- //
// Enum definitions //
// ----- //

// Home and away team definitions
#define HOME_TEAM 0
#define AWAY_TEAM 1

// nation region
#define REGION_AFRICA 1
#define REGION_ASIA 2

```

```

#define REGION_BALKAN 3
#define REGION_BRAZIL 4
#define REGION_BRITAIN 5
#define REGION_CARIBBEAN 6
#define REGION_CENTRAL_AMERICA 7
#define REGION_EASTERN_EUROPE 8
#define REGION_FRANCE 9
#define REGION_GERMANY 10
#define REGION_HOLLAND 11
#define REGION_ITALY 12
#define REGION_JAPAN 13
#define REGION_MIDDLE_EAST 14
#define REGION_NORTH_AFRICA 15
#define REGION_NORTH_AMERICA 16
#define REGION_NORTHERN_EUROPE 17
#define REGION_OCEANIA 18
#define REGION_RUSSIA 19
#define REGION_SCANDINAVIA 20
#define REGION_SOUTH_AFRICA 21
#define REGION_SOUTH_AMERICA 22
#define REGION_SOUTHERN_EUROPE 23
#define REGION_SPAIN 24
#define FIRST_REGION 1
#define LAST_REGION 24

// nation actual region
#define ACTUAL_REGION_AFRICA 1
#define ACTUAL_REGION_ASIA 2
#define ACTUAL_REGION_CARIBBEAN 3
#define ACTUAL_REGION_CENTRAL_AMERICA 4
#define ACTUAL_REGION_CENTRAL_EUROPE 5
#define ACTUAL_REGION_EASTERN_EUROPE 6
#define ACTUAL_REGION_MIDDLE_EAST 7
#define ACTUAL_REGION_NORTH_AFRICA 8
#define ACTUAL_REGION_NORTH_AMERICA 9
#define ACTUAL_REGION_OCEANIA 10
#define ACTUAL_REGION_SCANDINAVIA 11
#define ACTUAL_REGION_SOUTH_AMERICA 12
#define ACTUAL_REGION_SOUTHERN_EUROPE 13
#define ACTUAL_REGION_UK_AND_IRELAND 14
#define FIRST_ACTUAL_REGION 1
#define LAST_ACTUAL_REGION 14

// club professional status
#define PROFESSIONAL 1
#define SEMI_PRO 2
#define AMATEUR 3

// states of development
#define DEVELOPED_STATE 1
#define DEVLOPING_STATE 2
#define THIRD_WORLD_STATE 3

// database days of the week
#define DB_SUN 1
#define DB_MON 2
#define DB_TUE 3
#define DB_WED 4
#define DB_THUR 5
#define DB_FRI 6
#define DB_SAT 7

// staff job for club/nation
#define JOB_INVALID_JOB 0
#define JOB_CHAIRMAN 1
#define JOB_MANAGING_DIRECTOR 2
#define JOB_GENERAL_MANAGER 3
#define JOB_DIRECTOR_OF_FOOTBALL 4
#define JOB_MANAGER 5
#define JOB_ASSISTANT_MANAGER 6
#define JOB_RESERVE_TEAM_MANAGER 7
#define JOB_COACH 8
#define JOB_SCOUT 9
#define JOB_PHYSIO 10
#define JOB_PLAYER 11
#define JOB_PLAYER_MANAGER 12
#define JOB_PLAYER_ASSISTANT_MANAGER 13
#define PLAYER_RESERVE_TEAM_MANAGER 14
#define JOB_PLAYER_COACH 15
#define JOB_PLAYER_RETIRED 16
#define JOB_MEDIA_FUNDIT 17

// staff playing squad
#define INVALID_SQUAD 0
#define CLUB_SENIOR_SQUAD 0x01
#define CLUB_RESERVE_SQUAD 0x02
#define NATION_MAIN_SQUAD 0x04
#define NATION_B_SQUAD 0x08

// staff classification
#define INVALID_CLASSIFICATION 0
#define NON_PLAYER 1
#define PLAYER 2
#define NON_PLAYER_AND_PLAYER 3

```

```

#define YOUTH_PLAYER 4
#define NEW_HUMAN_MANAGER 5
#define SPARE_NON_PLAYER 6

// name tables
#define FIRSTNAME 1
#define SECONDNAME 2
#define COMMONNAME 3

// Player form definitions
#define VERY_POOR 0
#define POOR 1
#define AVERAGE 2
#define GOOD 3
#define VERY_GOOD 4
#define SUPERB 5

// Index file ids //
#define NUM_INDEX_TABLES 22
#define CLUB_TABLE 0
#define NATION_CLUBS_TABLE 1
#define COLOUR_TABLE 2
#define CONTINENT_TABLE 3
#define NATION_TABLE 4
#define STADIUM_TABLE 5
#define STAFF_TABLE 6
#define OFFICIALS_TABLE 7
#define YOUTH_PLAYER_TABLE 8 // Not used
#define NON_PLAYER_TABLE 9
#define PLAYER_TABLE 10
#define STAFF_COMP_TABLE 11
#define CLUB_COMP_TABLE 12
#define FIRST_NAME_TABLE 13
#define SECOND_NAME_TABLE 14
#define COMMON_NAME_TABLE 15
#define NATION_COMP_TABLE 16
#define STAFF_HISTORY_TABLE 17
#define STAFF_COMP_HISTORY_TABLE 18
#define CLUB_COMP_HISTORY_TABLE 19
#define NATION_COMP_HISTORY_TABLE 20
#define CITY_TABLE 21
#define STAFF_PREFERENCES_TABLE 22 // MUX with youth player table

#define test_for_bits_set( info, bits ) ( ( info & bits ) == bits )

// type definitions
#ifdef PLATFORM_MAC
typedef struct cm3_continents CM3_CONTINENTS;
typedef struct cm3_nations CM3_NATIONS;
typedef struct cm3_cities CM3_CITIES;
typedef struct cm3_stadiums CM3_STADIUMS;
typedef struct cm3_clubs CM3_CLUBS;
typedef struct cm3_staff CM3_STAFF;
typedef struct cm3_non_players CM3_NON_PLAYERS;
typedef struct cm3_players CM3_PLAYERS;
typedef struct cm3_staff_preferences CM3_STAFF_PREFERENCES;
typedef struct cm3_staff_history CM3_STAFF_HISTORY;
typedef struct cm3_club_comps CM3_CLUB_COMPS;
typedef struct cm3_club_comp_history CM3_CLUB_COMP_HISTORY;
typedef struct cm3_staff_comps CM3_STAFF_COMPS;
typedef struct cm3_staff_comp_history CM3_STAFF_COMP_HISTORY;
typedef struct cm3_officials CM3_OFFICIALS;
typedef struct cm3_colours CM3_COLOURS;
typedef struct cm3_names CM3_NAMES;
#else
typedef struct mac_cm3_continents CM3_CONTINENTS;
typedef struct mac_cm3_nations CM3_NATIONS;
typedef struct mac_cm3_cities CM3_CITIES;
typedef struct mac_cm3_stadiums CM3_STADIUMS;
typedef struct mac_cm3_clubs CM3_CLUBS;
typedef struct mac_cm3_staff CM3_STAFF;
typedef struct mac_cm3_non_players CM3_NON_PLAYERS;
typedef struct mac_cm3_players CM3_PLAYERS;
typedef struct mac_cm3_staff_preferences CM3_STAFF_PREFERENCES;
typedef struct mac_cm3_staff_history CM3_STAFF_HISTORY;
typedef struct mac_cm3_club_comps CM3_CLUB_COMPS;
typedef struct mac_cm3_club_comp_history CM3_CLUB_COMP_HISTORY;
typedef struct mac_cm3_staff_comps CM3_STAFF_COMPS;
typedef struct mac_cm3_staff_comp_history CM3_STAFF_COMP_HISTORY;
typedef struct mac_cm3_officials CM3_OFFICIALS;
typedef struct mac_cm3_colours CM3_COLOURS;
typedef struct mac_cm3_names CM3_NAMES;

#include "mac_database.h"
#endif

// ----- //
// Continents //
// ----- //

// This structure contains information about a nations name list beginning and
// end within each of the various names tables.
typedef struct

```

```

{
long first_name_start_index; // Start index for this nations first names
long second_name_start_index; // Start index for this nations second names
long common_name_start_index; // Start index for this nations common names
long number_first_names; // Total number of first names
long number_second_names; // Total number of second names
long number_common_names; // Total number of common names
long first_name_count; // Count of different types of first names
long second_name_count; // Count of different types of second names
long common_name_count; // Count of different types of common names
} NATION_NAME_INDEX_ENTRY;

#ifdef DATABASE_CPP
NATION_NAME_INDEX_ENTRY *nation_name_index_tbl = (NATION_NAME_INDEX_ENTRY*)0;
// Size is the same as the cm3_nations_tbl_sz
#else
extern NATION_NAME_INDEX_ENTRY *nation_name_index_tbl;
#endif

// the following structures need to be byte aligned
#ifdef PLATFORM_MAC
#pragma options align=packed
#endif

// ----- //
// Continents //
// ----- //

struct cm3_continents
{
// original data
long ContinentID;
char ContinentName[ SHORT_TXT_LENGTH ];
char ContinentGenderName;
char ContinentNameThreeLetter[ 4 ];
char ContinentNameContinentality[ SHORT_TXT_LENGTH ];
char ContinentFederationName[ LONG_TXT_LENGTH ];
char ContinentGenderFederationName;
char ContinentFederationNameShort[ SHORT_TXT_LENGTH ];
char ContinentGenderFederationNameShort;
SI_DOUBLE ContinentRegionalStrength;
};

// ----- //
// Nations //
// ----- //

struct cm3_nations
{
// original data
long NationID;
char NationName[ STANDARD_TXT_LENGTH ];
char NationGenderName;
char NationNameShort[ SHORT_TXT_LENGTH ];
char NationGenderNameShort;
char NationNameThreeLetter[ 4 ];
char NationNameNationality[ SHORT_TXT_LENGTH ];
CM3_COLOURS *NationContinent;
char NationRegion;
char NationActualRegion;
char NationFirstLanguage;
char NationSecondLanguage;
char NationThirdLanguage;
CM3_CITIES *NationCapitalCity;
char NationStateOfDevelopment;
char NationGroupMembership;
CM3_STADIUMS *NationNationalStadium;
char NationGameImportance;
char NationLeagueStandard;
short NationNumberClubs;
long NationNumberStaff; // Version 0x02 - Added
short NationSeasonUpdateDay; // Version 0x02 - Added
short NationReputation; // Version 0x02 - Changed char->short
CM3_COLOURS *NationForegroundColour1;
CM3_COLOURS *NationBackgroundColour1;
CM3_COLOURS *NationForegroundColour2;
CM3_COLOURS *NationBackgroundColour2;
CM3_COLOURS *NationForegroundColour3;
CM3_COLOURS *NationBackgroundColour3;
SI_DOUBLE NationFIFACoefficient;
SI_DOUBLE NationFIFACoefficient91;
SI_DOUBLE NationFIFACoefficient92;
SI_DOUBLE NationFIFACoefficient93;
SI_DOUBLE NationFIFACoefficient94;
SI_DOUBLE NationFIFACoefficient95;
SI_DOUBLE NationFIFACoefficient96;
SI_DOUBLE NationUEFACoefficient91;
SI_DOUBLE NationUEFACoefficient92;
SI_DOUBLE NationUEFACoefficient93;
SI_DOUBLE NationUEFACoefficient94;
SI_DOUBLE NationUEFACoefficient95;
SI_DOUBLE NationUEFACoefficient96;
CM3_NATIONS *NationRivals1;

```

```

CM3_NATIONS *NationRivals2;
CM3_NATIONS *NationRivals3;

// runtime data
char NationLeagueSelected;
long NationShortlistOffset; // Version 0x02 - Added
char NationGamesPlayed; // Version 0x02 - Moved to runtime
};

// WEATHER STRUCTURE DEFINITIONS //
// (Have Fun PJ) //
// This structure defines the current weather settings for an area.

#define SPECIAL_HAIL 1
#define SPECIAL_SLEET 2
#define SPECIAL_SNOW 3

typedef struct
{
char wind;
char precipitation;
char temperature;
char special;
} WEATHER_CONDITION;

// Wind defintion structure, this structure should give the percentage scale
// upon which each of the various wind type will happen within a season.
// Eg. calm - 3 // 0 - 3 % means calm weather occurs
// breezy - 20
// gusty - 75
// strong - 90
// gale - 100
typedef struct
{
char calm;
char breezy;
char gusty;
char strong;
char gale;
} WIND;

// Precipitation weather structure, this structure gives the percentage chance
// of each of the various types of precipitations happening.
typedef struct
{
char dry;
char wet;
char drizzle;
char shower;
char down_pour;
} PRECIPITATION;

// Temperature weather structure, this structure gives the percentage chance
// of each of the various types of precipitations happening.
typedef struct
{
char freezing;
char cold;
char mild;
char fine;
char warm;
char hot;
char very_hot;
} TEMPERATURE;

// This structure defines a seasons weather conditions within CM3.
typedef struct
{
short WeatherSeasonStartDay;
WIND WeatherSeasonWind;
PRECIPITATION WeatherSeasonPrecipitation;
TEMPERATURE WeatherSeasonTemperature;
} WEATHER_SEASON;

// CITY WEATHER : This structure defines the weather for a particular city.
// This information should be editable through the data editor
// (Thanks PJ)

// Season definitions ... for indexing into CityWeatherSeason
#define SPRING 0
#define SUMMER 1
#define AUTUMN 2
#define WINTER 3

typedef struct
{
long CityWeatherID; // ID of the city (auto generated by editor)
char CityWeatherName[ SHORT_TXT_LENGTH ]; // Name of the city which is modelled (editable)
WEATHER_SEASON CityWeatherSeason[ 4 ]; // Seasonal model for this city (editable)
WEATHER_CONDITION CityWeatherCondition; // Current Weather in this city (not editable)
} CITY_WEATHER;

```

```

// ----- //
// Cities //
// ----- //

struct cm3_cities
{
// original data
long CityID;
char CityName[ SHORT_TXT_LENGTH ];
char CityGenderName;
CM3_NATIONS *CityNation;
SI_DOUBLE CityLatitude;
SI_DOUBLE CityLongitude;
char CityAttraction;
long CityWeather;
// CITY_WEATHER *CityWeatherCity; // Weather city to use for this cities weather
};

// ----- //
// Stadiums //
// ----- //

struct cm3_stadiums
{
// original data
long StadiumID;
char StadiumName[ STANDARD_TXT_LENGTH ];
char StadiumGenderName;
CM3_CITIES *StadiumCity;
long StadiumCapacity;
long StadiumSeatingCapacity;
long StadiumExpansionCapacity;
CM3_STADIUMS *StadiumNearbyStadium;
char StadiumCovered;
char StadiumUnderSoilHeating;
};

// ----- //
// Clubs //
// ----- //

struct cm3_clubs
{
// original data
long ClubID;
char ClubName[ STANDARD_TXT_LENGTH ];
char ClubGenderName;
char ClubNameShort[ SHORT_TXT_LENGTH ];
char ClubGenderNameShort;
CM3_NATIONS *ClubNation;
CM3_CLUB_COMPS *ClubDivision;
CM3_CLUB_COMPS *ClubLastDivision;
char ClubLastPosition;
CM3_CLUB_COMPS *ClubReserveDivision;
char ClubProfessionalStatus;
long ClubCash;
CM3_STADIUMS *ClubStadium;
char ClubOwnStadium;
CM3_STADIUMS *ClubReserveStadium;
char ClubHomeMatchDay;
long ClubAttendance;
long ClubMinAttendance;
long ClubMaxAttendance;
char ClubTraining;
short ClubReputation; // Version 0x02 - Changed char->short
char ClubPLC;
CM3_COLOURS *ClubForegroundColour1;
CM3_COLOURS *ClubBackgroundColour1;
CM3_COLOURS *ClubForegroundColour2;
CM3_COLOURS *ClubBackgroundColour2;
CM3_COLOURS *ClubForegroundColour3;
CM3_COLOURS *ClubBackgroundColour3;
CM3_STAFF *ClubFavouriteStaff1;
CM3_STAFF *ClubFavouriteStaff2;
CM3_STAFF *ClubFavouriteStaff3;
CM3_STAFF *ClubDislikedStaff1;
CM3_STAFF *ClubDislikedStaff2;
CM3_STAFF *ClubDislikedStaff3;
CM3_CLUBS *ClubRivals1;
CM3_CLUBS *ClubRivals2;
CM3_CLUBS *ClubRivals3;
CM3_STAFF *ClubChaiman;
CM3_STAFF *ClubDirectorList[ DIRECTOR_SIZE ];
CM3_STAFF *ClubManager;
CM3_STAFF *ClubAssistantManager;
CM3_STAFF *ClubSquadList[ SQUAD_SIZE ];
CM3_STAFF *ClubCoachList[ COACH_SIZE ];
CM3_STAFF *ClubScoutList[ SCOUT_SIZE ];
CM3_STAFF *ClubPhysioList[ PHYSIO_SIZE ];

// runtime data
long ClubEuroFlag;
char ClubEuroSeeding;

```

```

CM3_STAFF *ClubTeamSelected[ TEAM_SZ ];
long ClubTacticTraining[ MAX_TACTIC_TRAINING ];
long ClubTacticSelected;
char ClubHasLinkedClub;

};

// ----- //
// Staff //
// ----- //

struct cm3_staff
{
long StaffID;
char *StaffFirstName;
char *StaffSecondName;
char *StaffCommonName;
CM_DATE StaffDateOfBirth;
short StaffYearOfBirth;
CM3_NATIONS *StaffNation;
CM3_NATIONS *StaffSecondNation;
unsigned char StaffInternationalApps;
unsigned char StaffInternationalGoals;
CM3_CLUBS *StaffNationContracted;
char StaffJobForNation;
CM_DATE StaffDateJoinedNation;
CM_DATE StaffContractExpiresNation;
CM3_CLUBS *StaffClubContracted;
char StaffJobForClub;
CM_DATE StaffDateJoinedClub;
CM_DATE StaffContractExpiresClub;
long StaffEstimatedWage;
long StaffEstimatedValue;
char StaffAdaptability;
char StaffAmbition;
char StaffDetermination;
char StaffLoyalty;
char StaffPressure;
char StaffProfessionalism;
char StaffSportsmanship;
char StaffTemperament;
char StaffPlayingSquad;
char StaffClassification;
char StaffClubValuation;
CM3_PLAYERS *StaffPlayerData;
CM3_STAFF_PREFERENCES *StaffPreferences; // Version 0x02 - New ptr type
CM3_NON_PLAYERS *StaffNonPlayerData;

// Runtime data //
char StaffSquadSelectedFor; // should be StaffEuroSquadFlag - Kev
};

// ----- //
// Non Players //
// ----- //

struct cm3_non_players
{
long StaffNonPlayerID;
short StaffNonPlayerCurrentAbility;
short StaffNonPlayerPotentialAbility;
short StaffNonPlayerHomeReputation; // Version 0x02 - Changed char->short
short StaffNonPlayerCurrentReputation; // Version 0x02 - Changed char->short
short StaffNonPlayerWorldReputation; // Version 0x02 - Changed char->short
char StaffNonPlayerAttacking;
char StaffNonPlayerBusiness;
char StaffNonPlayerCoaching;
char StaffNonPlayerCoachingGks;
char StaffNonPlayerCoachingTechnique;
char StaffNonPlayerDirectness;
char StaffNonPlayerDiscipline;
char StaffNonPlayerFreeRoles;
char StaffNonPlayerInterference;
char StaffNonPlayerJudgement;
char StaffNonPlayerJudgingPotential;
char StaffNonPlayerManHandling;
char StaffNonPlayerMarking;
char StaffNonPlayerMotivating;
char StaffNonPlayerOffside;
char StaffNonPlayerPatience;
char StaffNonPlayerPhysiotherapy;
char StaffNonPlayerPressing;
char StaffNonPlayerResources;
char StaffNonPlayerTactics;
char StaffNonPlayerYoungsters;
CM3_STAFF *StaffNonPlayerGoalkeeper;
CM3_STAFF *StaffNonPlayerSweeper;
CM3_STAFF *StaffNonPlayerDefender;
CM3_STAFF *StaffNonPlayerDefensiveMidfielder;
CM3_STAFF *StaffNonPlayerMidfielder;
CM3_STAFF *StaffNonPlayerAttackingMidfielder;
CM3_STAFF *StaffNonPlayerAttacker;
CM3_STAFF *StaffNonPlayerWingBack;
char StaffNonPlayerFormationPreferred;

```

```

};

// ----- //
// Players //
// ----- //

struct cm3_players
{
// original data
long StaffPlayerID;
char StaffPlayerSquadNumber;
short StaffPlayerCurrentAbility;
short StaffPlayerPotentialAbility;
short StaffPlayerHomeReputation; // Version 0x02 - Changed char->short
short StaffPlayerCurrentReputation; // Version 0x02 - Changed char->short
short StaffPlayerWorldReputation; // Version 0x02 - Changed char->short
char StaffPlayerGoalkeeper;
char StaffPlayerSweeper;
char StaffPlayerDefender;
char StaffPlayerDefensiveMidfielder;
char StaffPlayerMidfielder;
char StaffPlayerAttackingMidfielder;
char StaffPlayerAttacker;
char StaffPlayerWingBack;
char StaffPlayerRightSide;
char StaffPlayerLeftSide;
char StaffPlayerCentral;
char StaffPlayerFreeRole;
char StaffPlayerAcceleration;
char StaffPlayerAggression;
char StaffPlayerAgility;
char StaffPlayerAnticipation;
char StaffPlayerBalance;
char StaffPlayerBravery; // Value
char StaffPlayerConsistency;
char StaffPlayerCorners;
char StaffPlayerCrossing;
char StaffPlayerDecisions;
char StaffPlayerDirtiness;
char StaffPlayerDribbling;
char StaffPlayerFinishing;
char StaffPlayerFlair;
char StaffPlayerFreeKicks;
char StaffPlayerHandling;
char StaffPlayerHeading;
char StaffPlayerImportantMatches;
char StaffPlayerInjuryProneness;
char StaffPlayerJumping;
char StaffPlayerLeadership;
char StaffPlayerLeftFoot;
char StaffPlayerLongShots;
char StaffPlayerMarking;
char StaffPlayerMovement;
char StaffPlayerNaturalFitness;
char StaffPlayerOneOnOnes;
char StaffPlayerPace;
char StaffPlayerPassing;
char StaffPlayerPenalties;
char StaffPlayerPositioning;
char StaffPlayerReflexes;
char StaffPlayerRightFoot;
char StaffPlayerStamina;
char StaffPlayerStrength;
char StaffPlayerTackling;
char StaffPlayerTeamwork;
char StaffPlayerTechnique;
char StaffPlayerThrowIns;
char StaffPlayerVersatility;
char StaffPlayerVision;
char StaffPlayerWorkRate;

// runtime data
char StaffPlayerMorale;
};

// ----- //
// Player details //
// ----- //

struct cm3_staff_preferences
{
long StaffPreferencesID;
CM3_CLUBS *StaffFavouriteClubs1;
CM3_CLUBS *StaffFavouriteClubs2;
CM3_CLUBS *StaffFavouriteClubs3;
CM3_CLUBS *StaffDislikedClubs1;
CM3_CLUBS *StaffDislikedClubs2;
CM3_CLUBS *StaffDislikedClubs3;
CM3_STAFF *StaffFavouriteStaff1;
CM3_STAFF *StaffFavouriteStaff2;
CM3_STAFF *StaffFavouriteStaff3;
CM3_STAFF *StaffDislikedStaff1;
CM3_STAFF *StaffDislikedStaff2;
CM3_STAFF *StaffDislikedStaff3;
};

```



```

};

// ----- //
// Staff History //
// ----- //

struct cm3_staff_history
{
long StaffHistoryID;
CM3_STAFF *StaffHistoryStaff;
short StaffHistoryYear;
CM3_CLUBS *StaffHistoryClub;
char StaffHistoryOnLoan;
unsigned char StaffHistoryApps;
unsigned char StaffHistoryGoals;
};

// ----- //
// Club Competitions //
// ----- //

struct cm3_club_comps
{
long ClubCompID;
char ClubCompName[ STANDARD_TXT_LENGTH ];
char ClubCompGenderName;
char ClubCompNameShort[ SHORT_TXT_LENGTH ];
char ClubCompGenderNameShort;
char ClubCompNameThreeLetter[ 4 ];
char ClubCompScope;
char ClubCompSelected;
CM3_CONTINENTS *ClubCompContinent;
CM3_NATIONS *ClubCompNation;
CM3_COLOURS *ClubCompForegroundColour;
CM3_COLOURS *ClubCompBackgroundColour;
short ClubCompReputation; // Version 0x02 - Changed char->short
};

// ----- //
// Club Competition History //
// ----- //

struct cm3_club_comp_history
{
long ClubCompHistoryID;
CM3_CLUB_COMPS *ClubCompHistoryClubComp;
short ClubCompHistoryYear;
CM3_CLUBS *ClubCompHistoryWinners;
CM3_CLUBS *ClubCompHistoryRunnersUp;
CM3_CLUBS *ClubCompHistoryThirdPlaced;
CM3_CLUBS *ClubCompHistoryHosts;
};

// ----- //
// Staff Competitions //
// ----- //

struct cm3_staff_comps
{
long StaffCompID;
char StaffCompName[ STANDARD_TXT_LENGTH ];
char StaffCompGenderName;
char StaffCompNameShort[ SHORT_TXT_LENGTH ];
char StaffCompGenderNameShort;
CM3_CONTINENTS *StaffCompContinent;
CM3_NATIONS *StaffCompNation;
CM3_COLOURS *StaffCompForegroundColour;
CM3_COLOURS *StaffCompBackgroundColour;
short StaffCompReputation; // Version 0x02 - Changed char->short
};

// ----- //
// Staff Competition History //
// ----- //

struct cm3_staff_comp_history
{
long StaffCompHistoryID;
CM3_STAFF_COMPS *StaffCompHistoryStaffComp;
short StaffCompHistoryYear;
char *StaffCompHistoryFirstPlacedFirstName;
char *StaffCompHistoryFirstPlacedSecondName;
CM3_STAFF *StaffCompHistoryFirstPlacedIndex;
long StaffCompHistoryFirstPlacedInfo;
char *StaffCompHistorySecondPlacedFirstName;
char *StaffCompHistorySecondPlacedSecondName;
CM3_STAFF *StaffCompHistorySecondPlacedIndex;
long StaffCompHistorySecondPlacedInfo;
char *StaffCompHistoryThirdPlacedFirstName;
char *StaffCompHistoryThirdPlacedSecondName;
CM3_STAFF *StaffCompHistoryThirdPlacedIndex;
long StaffCompHistoryThirdPlacedInfo;
};

```

```

// ----- //
// Officials //
// ----- //

struct cm3_officials
{
long OfficialID;
char *OfficialFirstName;
char *OfficialSecondName;
CM_DATE OfficialDateOfBirth;
short OfficialYearOfBirth;
CM3_NATIONS *OfficialNation;
CM3_CITIES *OfficialCity;
short OfficialCurrentAbility;
short OfficialPotentialAbility;
short OfficialReputation;
char OfficialAllowingFlow;
char OfficialDiscipline;
char OfficialImportantMatches;
char OfficialPressure;
char OfficialRefereeing;
char OfficialRunningLine;
char OfficialTimekeeping;
};

// ----- //
// Colours //
// ----- //

struct cm3_colours
{
long ColourID;
char ColourName[ STANDARD_TXT_LENGTH ];
unsigned char ColourRedIntensity;
unsigned char ColourGreenIntensity;
unsigned char ColourBlueIntensity;
};

// ----- //
// Names //
// ----- //

// The cm3_names structure contains the information upon a name which is used
// by either the staff or officials within championship manager 3.
struct cm3_names
{
char Name[ STANDARD_TXT_LENGTH ];
long NameID;
long Nation;
char count;
};

// reset alignment
#ifdef PLATFORM_MAC
#pragma options align=reset
#endif

// ----- //
// Variables //
// ----- //

#define MAX_INDEX_NAME 50

// Index file struct //
typedef struct
{
char filename[ MAX_INDEX_NAME + 1 ];
long file_id;
long table_sz;
long offset;
long version;
} INDEX_TABLE;

#ifdef DATABASE_CPP
INDEX_TABLE index_info[ NUM_INDEX_TABLES ];

static BOOL game_format_data = FALSE;
static long num_of_friendlys = 0,
num_of_humans = 0;

// Table sizes //
long cm3_continents_tbl_sz = 0;
long cm3_nations_tbl_sz = 0;
long cm3_cities_tbl_sz = 0;
long cm3_stadiums_tbl_sz = 0;
long cm3_clubs_tbl_sz = 0;
long cm3_nation_clubs_tbl_sz = 0;
long cm3_staff_tbl_sz = 0;
long cm3_non_players_tbl_sz = 0;
long cm3_players_tbl_sz = 0;
long cm3_staff_preferences_tbl_sz = 0;
long cm3_staff_history_tbl_sz = 0;
long cm3_club_comps_tbl_sz = 0;
long cm3_club_comp_history_tbl_sz = 0;

```

```

long cm3_nation_club_comps_tbl_sz = 0;
long cm3_nation_club_comp_history_tbl_sz = 0;
long cm3_staff_comps_tbl_sz = 0;
long cm3_staff_comp_history_tbl_sz = 0;
long cm3_officials_tbl_sz = 0;
long cm3_colours_tbl_sz = 0;
long cm3_first_names_list_sz = 0;
long cm3_second_names_list_sz = 0;
long cm3_common_names_list_sz = 0;

char use_real_players = 1;

// Table ptrs //
CM3_CONTINENTS *cm3_continents_tbl = NULL;
CM3_NATIONS *cm3_nations_tbl = NULL;
CM3_CITIES *cm3_cities_tbl = NULL;
CM3_STADIUMS *cm3_stadiums_tbl = NULL;
CM3_CLUBS *cm3_clubs_tbl = NULL;
CM3_CLUBS *cm3_nation_clubs_tbl = NULL;
CM3_STAFF *cm3_staff_tbl = NULL;
CM3_NON_PLAYERS *cm3_non_players_tbl = NULL;
CM3_PLAYERS *cm3_players_tbl = NULL;
CM3_STAFF_PREFERENCES *cm3_staff_preferences_tbl = NULL;
CM3_STAFF_HISTORY *cm3_staff_history_tbl = NULL;
CM3_CLUB_COMPS *cm3_club_comps_tbl = NULL;
CM3_CLUB_COMP_HISTORY *cm3_club_comp_history_tbl = NULL;
CM3_CLUB_COMPS *cm3_nation_club_comps_tbl = NULL;
CM3_CLUB_COMP_HISTORY *cm3_nation_club_comp_history_tbl = NULL;
CM3_STAFF_COMPS *cm3_staff_comps_tbl = NULL;
CM3_STAFF_COMP_HISTORY *cm3_staff_comp_history_tbl = NULL;
CM3_OFFICIALS *cm3_officials_tbl = NULL;
CM3_COLOURS *cm3_colours_tbl = NULL;
CM3_NAMES *cm3_first_names_list = NULL;
CM3_NAMES *cm3_second_names_list = NULL;
CM3_NAMES *cm3_common_names_list = NULL;
#else
// Table sizes //
extern long cm3_continents_tbl_sz;
extern long cm3_nations_tbl_sz;
extern long cm3_cities_tbl_sz;
extern long cm3_stadiums_tbl_sz;
extern long cm3_clubs_tbl_sz;
extern long cm3_nation_clubs_tbl_sz;
extern long cm3_staff_tbl_sz;
extern long cm3_non_players_tbl_sz;
extern long cm3_players_tbl_sz;
extern long cm3_staff_preferences_tbl_sz;
extern long cm3_staff_history_tbl_sz;
extern long cm3_club_comps_tbl_sz;
extern long cm3_club_comp_history_tbl_sz;
extern long cm3_nation_club_comps_tbl_sz;
extern long cm3_nation_club_comp_history_tbl_sz;
extern long cm3_staff_comps_tbl_sz;
extern long cm3_staff_comp_history_tbl_sz;
extern long cm3_officials_tbl_sz;
extern long cm3_colours_tbl_sz;
extern long cm3_first_names_list_sz;
extern long cm3_second_names_list_sz;
extern long cm3_common_names_list_sz;

// Table sizes //
extern CM3_CONTINENTS *cm3_continents_tbl;
extern CM3_NATIONS *cm3_nations_tbl;
extern CM3_CITIES *cm3_cities_tbl;
extern CM3_STADIUMS *cm3_stadiums_tbl;
extern CM3_CLUBS *cm3_clubs_tbl;
extern CM3_CLUBS *cm3_nation_clubs_tbl;
extern CM3_STAFF *cm3_staff_tbl;
extern CM3_NON_PLAYERS *cm3_non_players_tbl;
extern CM3_PLAYERS *cm3_players_tbl;
extern CM3_STAFF_PREFERENCES *cm3_staff_preferences_tbl;
extern CM3_STAFF_HISTORY *cm3_staff_history_tbl;
extern CM3_CLUB_COMPS *cm3_club_comps_tbl;
extern CM3_CLUB_COMP_HISTORY *cm3_club_comp_history_tbl;
extern CM3_CLUB_COMPS *cm3_nation_club_comps_tbl;
extern CM3_CLUB_COMP_HISTORY *cm3_nation_club_comp_history_tbl;
extern CM3_STAFF_COMPS *cm3_staff_comps_tbl;
extern CM3_STAFF_COMP_HISTORY *cm3_staff_comp_history_tbl;
extern CM3_OFFICIALS *cm3_officials_tbl;
extern CM3_COLOURS *cm3_colours_tbl;
extern CM3_NAMES *cm3_first_names_list;
extern CM3_NAMES *cm3_second_names_list;
extern CM3_NAMES *cm3_common_names_list;

extern char use_real_players;
#endif

// Language header structure ..
typedef struct
{
    long club_tbl_sz;
    long continent_tbl_sz;
    long nation_comp_tbl_sz;
    long nation_tbl_sz;

```

```

long cities_tbl_sz;
long club_comp_tbl_sz;
long stadiums_tbl_sz;
long staff_comp_tbl_sz;
} LANG_HEADER;

// Structure containing club language information
typedef struct
{
char ClubName[ STANDARD_TXT_LENGTH ];
char GenderClubName;
char ClubNameShort[ SHORT_TXT_LENGTH ];
char GenderClubNameShort;
} CLUB_LANG;

// Structure containing continent language information
typedef struct
{
char ContinentName[ SHORT_TXT_LENGTH ];
char GenderContinentName;
char ContinentNameThreeLetter[ 4 ];
char ContinentNameNationality[ SHORT_TXT_LENGTH ];
char ContinentFederationName[ LONG_TXT_LENGTH ];
char GenderContinentFederationName;
char ContinentFederationNameShort[ SHORT_TXT_LENGTH ];
char GenderContinentFederationNameShort;
} CONTINENT_LANG;

// Structure containing nation competition language information
typedef struct
{
char NationCompName[ STANDARD_TXT_LENGTH ];
char GenderNationCompName;
char NationCompNameShort[ SHORT_TXT_LENGTH ];
char GenderNationCompNameShort;
char NationCompNameThreeLetter[ 4 ];
} NATION_COMP_LANG;

// Structure containing nation language information
typedef struct
{
char NationName[ STANDARD_TXT_LENGTH ];
char GenderNationName;
char NationNameShort[ SHORT_TXT_LENGTH ];
char GenderNationNameShort;
char NationNameThreeLetter[ 4 ];
char NationNameNationality[ SHORT_TXT_LENGTH ];
} NATION_LANG;

// Structure containing city language information
typedef struct
{
char CityName[ STANDARD_TXT_LENGTH ];
char GenderCityName;
} CITY_LANG;

// Structure containing stadium language information
typedef struct
{
char StadiumName[ STANDARD_TXT_LENGTH ];
char GenderStadiumName;
} STADIUM_LANG;

// Structure containing club comp language information
typedef struct
{
char ClubCompName[ STANDARD_TXT_LENGTH ];
char GenderClubCompName;
char ClubCompNameShort[ SHORT_TXT_LENGTH ];
char GenderClubCompNameShort;
char ClubCompNameThreeLetter[ 4 ];
} CLUB_COMP_LANG;

// Structure containing staff language information
typedef struct
{
char StaffCompName[ STANDARD_TXT_LENGTH ];
char GenderStaffCompName;
char StaffCompNameShort[ SHORT_TXT_LENGTH ];
char GenderStaffCompNameShort;
} STAFF_COMP_LANG;

// Class containing English language text for database
class ENGLISH_LANG
{
public:

ENGLISH_LANG( void );
~ENGLISH_LANG( void );

// access functions
char *get_nation_name( CM3_NATIONS *p_nation );
char *get_nation_name_three_letter( CM3_NATIONS *p_nation );
char *get_club_comp_name( CM3_CLUB_COMPS *p_club_comp );

```

```
char *get_continent_name( CM3_CONTINENTS *p_continent );
char *get_staff_comp_name( CM3_STAFF_COMPS *p_staff_comp );
char *get_club_name( CM3_CLUBS *p_club );
char *get_club_name_short( CM3_CLUBS *p_club );
char *get_stadium_name( CM3_STADIUMS *p_stadium );

private:
BOOL read_english_database( void );

LANG_HEADER lang_header;
CLUB_LANG *club_lang_data;
CONTINENT_LANG *continent_lang_data;
NATION_COMP_LANG *nation_comp_lang_data;
NATION_LANG *nation_lang_data;
CITY_LANG *city_lang_data;
STADIUM_LANG *stadium_lang_data;
CLUB_COMP_LANG *club_comp_lang_data;
STAFF_COMP_LANG *staff_comp_lang_data;
};

#ifdef DATABASE_CPP
LANGUAGE current_database_language = NO_LANGUAGE_SET;
long *club_remapping_indices = NULL;
#else
extern LANGUAGE current_database_language;
extern long *club_remapping_indices;
#endif
```

archibalduk

17-06-14, 07:53 PM

Quote:

Originally Posted by [milo](#)

another silly question: can't you check, examine the extraction tool made by john locke via olly or something similar, to see if the code you are looking for it's somewhere there, or even fix the tool to make it work like your ehms one?

No, I'm afraid not. Olly just shows assembler code rather than source code.

However, I thought I'd try it just for the hell of it anyhow. I downloaded Flex 2 and to my pleasant surprise JL has provided all of his source code with his tool! It's not quite in the format I need but, with some work, I should be able to re-create the DB structure using various parts of his source code. So thank you very much for pointing me in towards the answer!! I'm going to drop Graeme Kelly a PM to see if he has the structure file I need, but failing that, I should be absolutely fine using JL's source code. :ok:

Once I have reconstructed the DB structure I'll get to work on the editor (data exporting will be the first task). I'll share the DB structure in this thread because it'll be useful if anybody else needs it in the future. Although it's only of interest to a very small number of people, it is essentially the holy grail if you want to write any sort of tool for the game.

Quote:

Originally Posted by [djole2mcloud](#)

Can u update competition and staff histories via your tool?

Yes, the EHM tool can do both of those things. Also updating contract data, moving players/staff between teams, editing player/staff attributes and adding/deleting players/staff. There are also various other things it does (such as automatically setting all players' Pass Tendency attributes based on their career history).

I'm not proposing adding all of this for a CM editor as my plate is really full (working on the EHM editor, researching data for EHM and FHM and running my own site). What I'm looking to do is some basic editing - i.e. data exporting, attribute editing, contract/transfer editing, staff history importing and *maybe* adding/deleting players and staff.

Quote:

Originally Posted by [Craig Forrest](#)

Hey an EHM fan.... original free EHM or the overly-complicated and buggy 2005/2007 versions? :P

I think I'm a member of the blue line too..... not very active though.... only a handful of posts if I remember correctly

Any help you can give in terms of your knowledge would be awesome :ok:

Go Flyers :D

EHM 2007! And yes, I remember you joining and posting in our Arrivals forum.

archibalduk

17-06-14, 08:01 PM

Hugo9 - You, my friend, are a hero!!! Thank you ever so much. This is exactly what I need.

Apologies for any confusion caused by my earlier post. I started writing it about 2 hours ago but then went off to have something to eat. I clicked on "Submit" 2 hours later and so I hadn't spotted Hugo's post until afterwards.

Time to get to work on that editor...

djole2mcloud

17-06-14, 08:10 PM

another "HUGE" question...earlier versions of FM allowed data export from a save game...that how i extract all data i needed for patches ,sort them in .csv and then imported via JL tools in Cm DB...now,they have locked exporting data,except basic data(name,DOB,nationality,position),but most important attributes are locked out for export...
can u create a tool that can extract data from a FM save,or from a FM DB?
thanks

archibalduk

17-06-14, 09:29 PM

I took a brief look into this for you. It seems SI stopped publishing the FM DB structure roughly ten years ago. So it would involve a lot of work trying to figure out the structure by hand. It sounds like the .dat files are now compressed which makes this more challenging.

There's an outside chance that it *might* be possible. But I wouldn't want to commit to anything at this stage because I want to get the CM 01/02 Editor up and running first and also I'm a little concerned about whether or not dumping the data from FM might be frowned upon.

djole2mcloud

17-06-14, 11:13 PM

there is a tool FM Scout that can export those data from FM save...only,in later versions (after FM 11),most important data export is switched off as i mentioned earlier...
well,i understand your concerns of course...
<http://www.fmscout.com/a-fm-genie-scout-14.html>



Database Structure

Printable View

Show 40 post(s) from this thread on one page

Page 2 of 2 First 1 2

archibalduk

18-06-14, 06:50 PM

I expect SI probably asked them to remove that feature.

A few years back, before I wrote my EHM tool, I came across [ABBY Screenshot Reader](#). I wanted to export some attributes from EHM so that I could put them in a spreadsheet and chart how various players progressed over their careers. The screenshot reader was really useful because it would take a screenshot of the attributes list in-game (similar to [this screen](#)) and save it as an Excel spreadsheet.

I think you can get a 30 trial of the Screenshot Reader and the full version is only about £9 (I think I bought it half price at around £4 when it was on offer). I've found it very useful because it'll take a screenshot of anything, pick the text out, and save it as a spreadsheet or text document.

archibalduk

13-03-16, 03:28 PM

I thought I should mention that the Staff Comp History structure listed earlier in this thread is incorrect ([link](#)). The "char" members are in fact "long". So the structure is actually this:

Code:

```
// ----- //
// Staff Competition History //
// ----- //

struct cm3_staff_comp_history
{
    long StaffCompHistoryID;
    CM3_STAFF_COMPS *StaffCompHistoryStaffComp;
    short StaffCompHistoryYear;
    long *StaffCompHistoryFirstPlacedFirstName;
    long *StaffCompHistoryFirstPlacedSecondName;
    CM3_STAFF *StaffCompHistoryFirstPlacedIndex;
    long StaffCompHistoryFirstPlacedInfo;
    long *StaffCompHistorySecondPlacedFirstName;
    long *StaffCompHistorySecondPlacedSecondName;
    CM3_STAFF *StaffCompHistorySecondPlacedIndex;
    long StaffCompHistorySecondPlacedInfo;
    long *StaffCompHistoryThirdPlacedFirstName;
    long *StaffCompHistoryThirdPlacedSecondName;
    CM3_STAFF *StaffCompHistoryThirdPlacedIndex;
    long StaffCompHistoryThirdPlacedInfo;
};
```

The same goes for the Staff table too. I haven't checked through every table, but wherever a table has a pointer to a Common/First/Second Name which is listed as a "char" it ought to be a "long".

Code:

```
// ----- //
// Staff //
// ----- //

struct cm3_staff
{
    long StaffID;
    long *StaffFirstName;
    long *StaffSecondName;
    long *StaffCommonName;
    CM_DATE StaffDateOfBirth;
    short StaffYearOfBirth;
    CM3_NATIONS *StaffNation;
    CM3_NATIONS *StaffSecondNation;
    unsigned char StaffInternationalApps;
    unsigned char StaffInternationalGoals;
    CM3_CLUBS *StaffNationContracted;
    char StaffJobForNation;
    CM_DATE StaffDateJoinedNation;
    CM_DATE StaffContractExpiresNation;
    CM3_CLUBS *StaffClubContracted;
    char StaffJobForClub;
    CM_DATE StaffDateJoinedClub;
    CM_DATE StaffContractExpiresClub;
    long StaffEstimatedWage;
    long StaffEstimatedValue;
    char StaffAdaptability;
    char StaffAmbition;
    char StaffDetermination;
    char StaffLoyalty;
    char StaffPressure;
    char StaffProfessionalism;
```

```
char StaffSportsmanship;
char StaffTemperament;
char StaffPlayingSquad;
char StaffClassification;
char StaffClubValuation;
CM3_PLAYERS *StaffPlayerData;
CM3_STAFF_PREFERENCES *StaffPreferences; // Version 0x02 - New ptr type
CM3_NON_PLAYERS *StaffNonPlayerData;

// Runtime data //
char StaffSquadSelectedFor; // should be StaffEuroSquadFlag - Kev
};
```

vfilatov

15-07-16, 08:23 AM

Hi guys,

I want to write CMScout-like software which can show intrinsic values and search players based on these values. I found CMScout source code which provided info about database structure. Attribute values internally stored as 2-byte integer. For CA15 attributes CMScout use some magic code to convert internal value to 1-20 value depending on CA:

Code:

```
function TCMDatabase.HighConvert(Attribute: ShortInt; Ability: Word): Byte;
var
  dblTemp,
  dblResult: Double;
begin
  dblTemp:=(Attribute / 10) + (Ability / 20) + 10;
  dblResult:=(dblTemp * dblTemp / 30) + (dblTemp / 3) + 0.5;
  if (dblResult < 1) then
    dblResult:=1;
  if (dblResult > 20) then
    dblResult:=20;
  Result:=Trunc(dblResult);
end;

function TCMDatabase.LowConvert(Attribute: ShortInt; Ability: Word): Byte;
var
  dblTemp,
  dblResult: Double;
begin
  dblTemp:=(Attribute / 10) + (Ability / 200) + 10;
  dblResult:=(dblTemp * dblTemp / 30) + (dblTemp / 3) + 0.5;
  if (dblResult < 1) then
    dblResult:=1;
  if (dblResult > 20) then
    dblResult:=20;
  Result:=Trunc(dblResult);
end;
```

Can anybody know how I can convert internal value to 0-255 intrinsic value? May be GK Save Game Editor source code available or conversion algorithm described elsewhere?

Thanks,
Vladimir

Dermotron

26-10-16, 01:03 PM

Any way of converting the .dat files to read by access or sql?

vfilatov

06-03-17, 12:03 PM

Quote:

Originally Posted by [vfilatov](#)

Can anybody know how I can convert internal value to 0-255 intrinsic value?

My question is stupid. Attribute values internally stored as 1-byte integer and stored values is intrinsic directly.

vfilatov

23-01-20, 02:47 PM

Is a complete description of the database structure available somewhere?
And the same question about .tct file format.

MadScientist

23-01-20, 06:31 PM

Quote:

Originally Posted by [vfilatov](#)

Is a complete description of the database structure available somewhere?

Yes, in this post: <https://champman0102.co.uk/showthread...351#post173351>
And some corrections of it here: <https://champman0102.co.uk/showthread...927#post286927>
And an open source database reader/writer is implemented here: <https://github.com/archibalduk/Trans...aster/database>

Quote:

Originally Posted by **vfilatov** 

And the same question about .tct file format.

I don't know.

MarcoVanBast

29-04-20, 02:04 PM

Quote:

Originally Posted by **Hugo9** 

Code:

```
//-----  
// Project: Championship Manager 3 Program (and subsequent updates)  
// Sports Interactive  
// Copyright © 1997. All Rights Reserved.  
//  
// SUBSYSTEM: Fix Application  
// FILE: database.h  
// AUTHOR: Marc Vaughan  
//  
// OVERVIEW  
// ~~~~~  
// Header file for the database access functions.  
//  
//-----  
  
// Ifdef wrapping to prevent multiple inclusions  
#ifndef DATABASE_H  
#define DATABASE_H  
  
//Added by Graeme Kelly  
typedef struct  
{  
short day; // days from Jan 1st.  
short year;  
long LeapYear; // =1 if year is a leapyear, 0 otherwise.  
} CM_DATE;  
  
// ----- //  
// Custom Include Files //  
// ----- //  
#include "date.h"  
#include "utils.h"  
#include "zipdir.h"  
#include "club.h"  
#include "langlib.h"  
  
// Define for initialising club rep table  
#define INITIALISE_TABLE TRUE  
  
// Invalid date def  
#define INVALID_DATE CM_DATE( 31, 1900 )  
  
// Languages  
#define DB_ALBANIAN 1  
#define DB_ARABIC 2  
#define DB_BULGARIAN 3  
#define DB_CHINESE 4  
#define DB_CROATIAN 5  
#define DB_CZECH 6  
#define DB_DUTCH 7  
#define DB_ENGLISH 8  
#define DB_FINNISH 9  
#define DB_FRENCH 10  
#define DB_GERMAN 11  
#define DB_GREEK 12  
#define DB_HEBREW 13  
#define DB_HUNGARIAN 14  
#define DB_ITALIAN 15  
#define DB_JAPANESE 16  
#define DB_KOREAN 17  
#define DB_POLISH 18  
#define DB_PORTUGUESE 19  
#define DB_ROMANIAN 20  
#define DB_RUSSIAN 21  
#define DB_NORWEGIAN 22  
#define DB_SERBIAN 23  
#define DB_SPANISH 24  
#define DB_TURKISH 25  
#define DB_BURMESE 26
```

```

#define DB_MALAY 27
#define DB_PERSIAN 28
#define DB_SLOVAK 29
#define DB_SWEDISH 30
#define DB_ICELANDIC 31
#define DB_DANISH 32
#define DB_AFRIKAANS 33
#define DB_ALBANIAN 34
#define DB_AMHARIC 35
#define DB_ARMENIAN 36
#define DB_AZERI 37
#define DB_BANGLA 38
#define DB_ESTONIAN 39
#define DB_GEORGIAN 40
#define DB_HINDI 41
#define DB_JAVANESE 42
#define DB_KHMER 43
#define DB_KYRGYZ 44
#define DB_LAO 45
#define DB_LATVIAN 46
#define DB_LETBURGESH 47
#define DB_LITHUANIAN 48
#define DB_MACEDONIAN 49
#define DB_MALAGASY 50
#define DB_MALTESE 51
#define DB_SINHALESE 52
#define DB_SLOVENIAN 53
#define DB_SOMALI 54
#define DB_SWAHILI 55
#define DB_TAJIK 56
#define DB_TAMIL 57
#define DB_THAI 58
#define DB_UZBEK 59
#define DB_VIETNAMESE 60

#define DB_NUMBER_OF_CM3_LANGUAGES 60

// Formation definitions
#define FORMATION_NOT_SET 0
#define FORMATION_532SW 1
#define FORMATION_532 2
#define FORMATION_442 3
#define FORMATION_433 4
#define FORMATION_424 5
#define FORMATION_352 6
#define FORMATION_41212 7
#define FORMATION_451 8
#define FORMATION_343 9

#define DB_NUMBER_OF_FORMATION_SHAPES 10

// NationGroupDefines
#define EEC_COUNTRY 2
#define NON_EEC_COUNTRY 1

// Selected leagues defs
#define NOT_SELECTED 0x0000
#define SELECTED_BACKGROUND 0x0001
#define SELECTED_FOREGROUND 0x0002
#define ALL_DIVISIONS_ACTIVE 0x0004

// Compatability defs //
#define cm3_second_name_list cm3_second_names_list
#define CM3_PLAYER CM3_PLAYERS
#define CM3_NON_PLAYER CM3_NON_PLAYERS
#define CM3_CLUB_COMP CM3_CLUB_COMPS
#define cm3_club_tbl cm3_clubs_tbl
#define cm3_club_tbl_sz cm3_clubs_tbl_sz
#define cm3_player_tbl cm3_players_tbl
#define CM3_NAME CM3_NAMES
#define CENTER CENTRE
#define CM3_YOUTH_PLAYER CM3_YOUTH_PLAYERS

// -----//
// Constants //
// -----//

#define INDEX_FILENAME "index.dat"

#define DB_CONTINENTS_VERSION 0x01
#define DB_NATIONS_VERSION 0x02
#define DB_CITIES_CORRUPT_ON_MAC 0x01
#define DB_CITIES_VERSION 0x02
#define DB_STADIUMS_VERSION 0x01
#define DB_CLUBS_VERSION 0x02
#define DB_STAFF_VERSION 0x02
#define DB_NON_PLAYERS_VERSION 0x02
#define DB_PLAYERS_VERSION 0x02
#define DB_STAFF_PREFERENCES_VERSION 0x01
#define DB_STAFF_HISTORY_VERSION 0x01
#define DB_CLUB_COMPS_VERSION 0x02
#define DB_CLUB_COMP_HISTORY_VERSION 0x01
#define DB_STAFF_CMPS_VERSION 0x02
#define DB_STAFF_COMP_HISTORY_VERSION 0x01

```

```

#define DB_OFFICIALS_VERSION 0x01
#define DB_COLOURS_VERSION 0x01
#define DB_NAMES_VERSION 0x01

// Primary colour definitions
#define PRIMARY_BLACK 1
#define PRIMARY_WHITE 2
#define PRIMARY_GREY 3
#define PRIMARY_YELLOW 4
#define PRIMARY_GREEN 5
#define PRIMARY_BLUE 6
#define PRIMARY_PURPLE 7
#define PRIMARY_RED 8
#define PRIMARY_ORANGE 9
#define PRIMARY_BROWN 10

// Condition level below which a player is considered unable to play
#define CONDITION_INJURED 60

// max number of friendly competitions
#define MAX_FRIENDLY_COMPETITIONS 127

// text lengths
#define LONG_TXT_LENGTH 101
#define STANDARD_TXT_LENGTH 51
#define SHORT_TXT_LENGTH 26

// club staff table sizes
#define DIRECTOR_SIZE 3
#define SQUAD_SIZE 50
#define SCOUT_SIZE 7
#define COACH_SIZE 5
#define PHYSIO_SIZE 3

// max number of human players
#define MAX_HUMAN_PLAYERS 16

// Maximum number of tactics in training at once
#define MAX_TACTIC_TRAINING 4

// Maximum size of a squad for a match
#define TEAM_SZ 20

// Ability defs
#define LOWER_TECHNICAL_ABILITY_LIMIT -125
#define UPPER_TECHNICAL_ABILITY_LIMIT 125
#define LOWER_NON_TECHNICAL_ABILITY_LIMIT 1
#define UPPER_NON_TECHNICAL_ABILITY_LIMIT 20

// ----- //
// Enum definitions //
// ----- //

// Home and away team definitions
#define HOME_TEAM 0
#define AWAY_TEAM 1

// nation region
#define REGION_AFRICA 1
#define REGION_ASIA 2
#define REGION_BALKAN 3
#define REGION_BRAZIL 4
#define REGION_BRITAIN 5
#define REGION_CARIBBEAN 6
#define REGION_CENTRAL_AMERICA 7
#define REGION_EASTERN_EUROPE 8
#define REGION_FRANCE 9
#define REGION_GERMANY 10
#define REGION_HOLLAND 11
#define REGION_ITALY 12
#define REGION_JAPAN 13
#define REGION_MIDDLE_EAST 14
#define REGION_NORTH_AFRICA 15
#define REGION_NORTH_AMERICA 16
#define REGION_NORTHERN_EUROPE 17
#define REGION_OCEANIA 18
#define REGION_RUSSIA 19
#define REGION_SCANDINAVIA 20
#define REGION_SOUTH_AFRICA 21
#define REGION_SOUTH_AMERICA 22
#define REGION_SOUTHERN_EUROPE 23
#define REGION_SPAIN 24
#define FIRST_REGION 1
#define LAST_REGION 24

// nation actual region
#define ACTUAL_REGION_AFRICA 1
#define ACTUAL_REGION_ASIA 2
#define ACTUAL_REGION_CARIBBEAN 3
#define ACTUAL_REGION_CENTRAL_AMERICA 4
#define ACTUAL_REGION_CENTRAL_EUROPE 5
#define ACTUAL_REGION_EASTERN_EUROPE 6
#define ACTUAL_REGION_MIDDLE_EAST 7
#define ACTUAL_REGION_NORTH_AFRICA 8

```

```

#define ACTUAL_REGION_NORTH_AMERICA 9
#define ACTUAL_REGION_OCEANIA 10
#define ACTUAL_REGION_SCANDINAVIA 11
#define ACTUAL_REGION_SOUTH_AMERICA 12
#define ACTUAL_REGION_SOUTHERN_EUROPE 13
#define ACTUAL_REGION_UK_AND_IRELAND 14
#define FIRST_ACTUAL_REGION 1
#define LAST_ACTUAL_REGION 14

// club professional status
#define PROFESSIONAL 1
#define SEMI_PRO 2
#define AMATEUR 3

// states of development
#define DEVELOPED_STATE 1
#define DEVLOPING_STATE 2
#define THIRD_WORLD_STATE 3

// database days of the week
#define DB_SUN 1
#define DB_MON 2
#define DB_TUE 3
#define DB_WED 4
#define DB_THUR 5
#define DB_FRI 6
#define DB_SAT 7

// staff job for club/nation
#define JOB_INVALID_JOB 0
#define JOB_CHAIRMAN 1
#define JOB_MANAGING_DIRECTOR 2
#define JOB_GENERAL_MANAGER 3
#define JOB_DIRECTOR_OF_FOOTBALL 4
#define JOB_MANAGER 5
#define JOB_ASSISTANT_MANAGER 6
#define JOB_RESERVE_TEAM_MANAGER 7
#define JOB_COACH 8
#define JOB_SCOUT 9
#define JOB_PHYSIO 10
#define JOB_PLAYER 11
#define JOB_PLAYER_MANAGER 12
#define JOB_PLAYER_ASSISTANT_MANAGER 13
#define PLAYER_RESERVE_TEAM_MANAGER 14
#define JOB_PLAYER_COACH 15
#define JOB_PLAYER_RETIRED 16
#define JOB_MEDIA_FUNDIT 17

// staff playing squad
#define INVALID_SQUAD 0
#define CLUB_SENIOR_SQUAD 0x01
#define CLUB_RESERVE_SQUAD 0x02
#define NATION_MAIN_SQUAD 0x04
#define NATION_B_SQUAD 0x08

// staff classification
#define INVALID_CLASSIFICATION 0
#define NON_PLAYER 1
#define PLAYER 2
#define NON_PLAYER_AND_PLAYER 3
#define YOUTH_PLAYER 4
#define NEW_HUMAN_MANAGER 5
#define SPARE_NON_PLAYER 6

// name tables
#define FIRSNAME 1
#define SECONNAME 2
#define COMMONNAME 3

// Player form definitions
#define VERY_POOR 0
#define POOR 1
#define AVERAGE 2
#define GOOD 3
#define VERY_GOOD 4
#define SUPERB 5

// Index file ids //
#define NUM_INDEX_TABLES 22
#define CLUB_TABLE 0
#define NATION_CLUBS_TABLE 1
#define COLOUR_TABLE 2
#define CONTINENT_TABLE 3
#define NATION_TABLE 4
#define STADIUM_TABLE 5
#define STAFF_TABLE 6
#define OFFICIALS_TABLE 7
#define YOUTH_PLAYER_TABLE 8 // Not used
#define NON_PLAYER_TABLE 9
#define PLAYER_TABLE 10
#define STAFF_COMP_TABLE 11
#define CLUB_COMP_TABLE 12
#define FIRST_NAME_TABLE 13
#define SECOND_NAME_TABLE 14

```

```

#define COMMON_NAME_TABLE 15
#define NATION_COMP_TABLE 16
#define STAFF_HISTORY_TABLE 17
#define STAFF_COMP_HISTORY_TABLE 18
#define CLUB_COMP_HISTORY_TABLE 19
#define NATION_COMP_HISTORY_TABLE 20
#define CITY_TABLE 21
#define STAFF_PREFERENCES_TABLE 22 // MUX with youth player table

#define test_for_bits_set( info, bits ) ( ( info & bits ) == bits )

// type definitions
#ifndef PLATFORM_MAC
typedef struct cm3_continents CM3_CONTINENTS;
typedef struct cm3_nations CM3_NATIONS;
typedef struct cm3_cities CM3_CITIES;
typedef struct cm3_stadiums CM3_STADIUMS;
typedef struct cm3_clubs CM3_CLUBS;
typedef struct cm3_staff CM3_STAFF;
typedef struct cm3_non_players CM3_NON_PLAYERS;
typedef struct cm3_players CM3_PLAYERS;
typedef struct cm3_staff_preferences CM3_STAFF_PREFERENCES;
typedef struct cm3_staff_history CM3_STAFF_HISTORY;
typedef struct cm3_club_comps CM3_CLUB_COMPS;
typedef struct cm3_club_comp_history CM3_CLUB_COMP_HISTORY;
typedef struct cm3_staff_comps CM3_STAFF_COMPS;
typedef struct cm3_staff_comp_history CM3_STAFF_COMP_HISTORY;
typedef struct cm3_officials CM3_OFFICIALS;
typedef struct cm3_colours CM3_COLOURS;
typedef struct cm3_names CM3_NAMES;
#else
typedef struct mac_cm3_continents CM3_CONTINENTS;
typedef struct mac_cm3_nations CM3_NATIONS;
typedef struct mac_cm3_cities CM3_CITIES;
typedef struct mac_cm3_stadiums CM3_STADIUMS;
typedef struct mac_cm3_clubs CM3_CLUBS;
typedef struct mac_cm3_staff CM3_STAFF;
typedef struct mac_cm3_non_players CM3_NON_PLAYERS;
typedef struct mac_cm3_players CM3_PLAYERS;
typedef struct mac_cm3_staff_preferences CM3_STAFF_PREFERENCES;
typedef struct mac_cm3_staff_history CM3_STAFF_HISTORY;
typedef struct mac_cm3_club_comps CM3_CLUB_COMPS;
typedef struct mac_cm3_club_comp_history CM3_CLUB_COMP_HISTORY;
typedef struct mac_cm3_staff_comps CM3_STAFF_COMPS;
typedef struct mac_cm3_staff_comp_history CM3_STAFF_COMP_HISTORY;
typedef struct mac_cm3_officials CM3_OFFICIALS;
typedef struct mac_cm3_colours CM3_COLOURS;
typedef struct mac_cm3_names CM3_NAMES;

#include "mac_database.h"
#endif

// ----- //
// Continents //
// ----- //

// This structure contains information about a nations name list beginning and
// end within each of the various names tables.
typedef struct
{
    long first_name_start_index; // Start index for this nations first names
    long second_name_start_index; // Start index for this nations second names
    long common_name_start_index; // Start index for this nations common names
    long number_first_names; // Total number of first names
    long number_second_names; // Total number of second names
    long number_common_names; // Total number of common names
    long first_name_count; // Count of different types of first names
    long second_name_count; // Count of different types of second names
    long common_name_count; // Count of different types of common names
} NATION_NAME_INDEX_ENTRY;

#ifdef DATABASE_CPP
NATION_NAME_INDEX_ENTRY *nation_name_index_tbl = (NATION_NAME_INDEX_ENTRY*)0;
// Size is the same as the cm3_nations_tbl_sz
#else
extern NATION_NAME_INDEX_ENTRY *nation_name_index_tbl;
#endif

// the following structures need to be byte aligned
#ifdef PLATFORM_MAC
#pragma options align=packed
#endif

// ----- //
// Continents //
// ----- //

struct cm3_continents
{
    // original data
    long ContinentID;
    char ContinentName[ SHORT_TXT_LENGTH ];

```

```

char ContinentGenderName;
char ContinentNameThreeLetter[ 4 ];
char ContinentNameContinentality[ SHORT_TXT_LENGTH ];
char ContinentFederationName[ LONG_TXT_LENGTH ];
char ContinentGenderFederationName;
char ContinentFederationNameShort[ SHORT_TXT_LENGTH ];
char ContinentGenderFederationNameShort;
SI_DOUBLE ContinentRegionalStrength;
};

// ----- //
// Nations //
// ----- //

struct cm3_nations
{
// original data
long NationID;
char NationName[ STANDARD_TXT_LENGTH ];
char NationGenderName;
char NationNameShort[ SHORT_TXT_LENGTH ];
char NationGenderNameShort;
char NationNameThreeLetter[ 4 ];
char NationNameNationality[ SHORT_TXT_LENGTH ];
CM3_CONTINENTS *NationContinent;
char NationRegion;
char NationActualRegion;
char NationFirstLanguage;
char NationSecondLanguage;
char NationThirdLanguage;
CM3_CITIES *NationCapitalCity;
char NationStateOfDevelopment;
char NationGroupMembership;
CM3_STADIUMS *NationNationalStadium;
char NationGameImportance;
char NationLeagueStandard;
short NationNumberClubs;
long NationNumberStaff; // Version 0x02 - Added
short NationSeasonUpdateDay; // Version 0x02 - Added
short NationReputation; // Version 0x02 - Changed char->short
CM3_COLOURS *NationForegroundColour1;
CM3_COLOURS *NationBackgroundColour1;
CM3_COLOURS *NationForegroundColour2;
CM3_COLOURS *NationBackgroundColour2;
CM3_COLOURS *NationForegroundColour3;
CM3_COLOURS *NationBackgroundColour3;
SI_DOUBLE NationFIFACoefficient;
SI_DOUBLE NationFIFACoefficient91;
SI_DOUBLE NationFIFACoefficient92;
SI_DOUBLE NationFIFACoefficient93;
SI_DOUBLE NationFIFACoefficient94;
SI_DOUBLE NationFIFACoefficient95;
SI_DOUBLE NationFIFACoefficient96;
SI_DOUBLE NationUEFACoefficient91;
SI_DOUBLE NationUEFACoefficient92;
SI_DOUBLE NationUEFACoefficient93;
SI_DOUBLE NationUEFACoefficient94;
SI_DOUBLE NationUEFACoefficient95;
SI_DOUBLE NationUEFACoefficient96;
CM3_NATIONS *NationRivals1;
CM3_NATIONS *NationRivals2;
CM3_NATIONS *NationRivals3;

// runtime data
char NationLeagueSelected;
long NationShortListOffset; // Version 0x02 - Added
char NationGamesPlayed; // Version 0x02 - Moved to runtime
};

// WEATHER STRUCTURE DEFINITIONS //
// (Have Fun PJ) //
// This structure defines the current weather settings for an area.

#define SPECIAL_HAIL 1
#define SPECIAL_SLEET 2
#define SPECIAL_SNOW 3

typedef struct
{
char wind;
char precipitation;
char temperature;
char special;
} WEATHER_CONDITION;

// Wind defintion structure, this structure should give the precentage scale
// upon which each of the various wind type will happen within a season.
// Eg. calm - 3 // 0 - 3 % means calm weather occurs
// breezy - 20
// gusty - 75
// strong - 90
// gale - 100
typedef struct

```

```

{
char calm;
char breezy;
char gusty;
char strong;
char gale;
} WIND;

// Precipitation weather structure, this structure gives the percentage chance
// of each of the various types of precipitations happening.
typedef struct
{
char dry;
char wet;
char drizzle;
char shower;
char down_pour;
} PRECIPITATION;

// Temperature weather structure, this structure gives the percentage chance
// of each of the various types of precipitations happening.
typedef struct
{
char freezing;
char cold;
char mild;
char fine;
char warm;
char hot;
char very_hot;
} TEMPERATURE;

// This structure defines a seasons weather conditions within CM3.
typedef struct
{
short WeatherSeasonStartDay;
WIND WeatherSeasonWind;
PRECIPITATION WeatherSeasonPrecipitation;
TEMPERATURE WeatherSeasonTemperature;
} WEATHER_SEASON;

// CITY_WEATHER : This structure defines the weather for a particular city.
// This information should be editable through the data editor
// (Thanks PJ)

// Season definitions ... for indexing into CityWeatherSeason
#define SPRING 0
#define SUMMER 1
#define AUTUMN 2
#define WINTER 3

typedef struct
{
long CityWeatherID; // ID of the city (auto generated by editor)
char CityWeatherName[ SHORT_TXT_LENGTH ]; // Name of the city which is modelled (editable)
WEATHER_SEASON CityWeatherSeason[ 4 ]; // Seasonal model for this city (editable)
WEATHER_CONDITION CityWeatherCondition; // Current Weather in this city (not editable)
} CITY_WEATHER;

// ----- //
// Cities //
// ----- //

struct cm3_cities
{
// original data
long CityID;
char CityName[ SHORT_TXT_LENGTH ];
char CityGenderName;
CM3_NATIONS *CityNation;
SI_DOUBLE CityLatitude;
SI_DOUBLE CityLongitude;
char CityAttraction;
long CityWeather;
// CITY_WEATHER *CityWeatherCity; // Weather city to use for this cities weather
};

// ----- //
// Stadiums //
// ----- //

struct cm3_stadiums
{
// original data
long StadiumID;
char StadiumName[ STANDARD_TXT_LENGTH ];
char StadiumGenderName;
CM3_CITIES *StadiumCity;
long StadiumCapacity;
long StadiumSeatingCapacity;
long StadiumExpansionCapacity;
CM3_STADIUMS *StadiumNearbyStadium;
};

```

```

char StadiumCovered;
char StadiumUnderSoilHeating;
};

// ----- //
// Clubs //
// ----- //

struct cm3_clubs
{
// original data
long ClubID;
char ClubName[ STANDARD_TXT_LENGTH ];
char ClubGenderName;
char ClubNameShort[ SHORT_TXT_LENGTH ];
char ClubGenderNameShort;
CM3_NATIONS *ClubNation;
CM3_CLUB_COMPS *ClubDivision;
CM3_CLUB_COMPS *ClubLastDivision;
char ClubLastPosition;
CM3_CLUB_COMPS *ClubReserveDivision;
char ClubProfessionalStatus;
long ClubCash;
CM3_STADIUMS *ClubStadium;
char ClubOwnStadium;
CM3_STADIUMS *ClubReserveStadium;
char ClubHomeMatchDay;
long ClubAttendance;
long ClubMinAttendance;
long ClubMaxAttendance;
char ClubTraining;
short ClubReputation; // Version 0x02 - Changed char->short
char ClubPLC;
CM3_COLOURS *ClubForegroundColour1;
CM3_COLOURS *ClubBackgroundColour1;
CM3_COLOURS *ClubForegroundColour2;
CM3_COLOURS *ClubBackgroundColour2;
CM3_COLOURS *ClubForegroundColour3;
CM3_COLOURS *ClubBackgroundColour3;
CM3_STAFF *ClubFavouriteStaff1;
CM3_STAFF *ClubFavouriteStaff2;
CM3_STAFF *ClubFavouriteStaff3;
CM3_STAFF *ClubDislikedStaff1;
CM3_STAFF *ClubDislikedStaff2;
CM3_STAFF *ClubDislikedStaff3;
CM3_CLUBS *ClubRivals1;
CM3_CLUBS *ClubRivals2;
CM3_CLUBS *ClubRivals3;
CM3_STAFF *ClubChaiman;
CM3_STAFF *ClubDirectorList[ DIRECTOR_SIZE ];
CM3_STAFF *ClubManager;
CM3_STAFF *ClubAssistantManager;
CM3_STAFF *ClubSquadList[ SQUAD_SIZE ];
CM3_STAFF *ClubCoachList[ COACH_SIZE ];
CM3_STAFF *ClubScoutList[ SCOUT_SIZE ];
CM3_STAFF *ClubPhysioList[ PHYSIO_SIZE ];

// runtime data
long ClubEuroFlag;
char ClubEuroSeeding;
CM3_STAFF *ClubTeamSelected[ TEAM_SZ ];
long ClubTacticTraining[ MAX_TACTIC_TRAINING ];
long ClubTacticSelected;
char ClubHasLinkedClub;

};

// ----- //
// Staff //
// ----- //

struct cm3_staff
{
long StaffID;
char *StaffFirstName;
char *StaffSecondName;
char *StaffCommonName;
CM_DATE StaffDateOfBirth;
short StaffYearOfBirth;
CM3_NATIONS *StaffNation;
CM3_NATIONS *StaffSecondNation;
unsigned char StaffInternationalApps;
unsigned char StaffInternationalGoals;
CM3_CLUBS *StaffNationContracted;
char StaffJobForNation;
CM_DATE StaffDateJoinedNation;
CM_DATE StaffContractExpiresNation;
CM3_CLUBS *StaffClubContracted;
char StaffJobForClub;
CM_DATE StaffDateJoinedClub;
CM_DATE StaffContractExpiresClub;
long StaffEstimatedWage;
long StaffEstimatedValue;
char StaffAdaptability;
};

```



```

char StaffAmbition;
char StaffDetermination;
char StaffLoyalty;
char StaffPressure;
char StaffProfessionalism;
char StaffSportsmanship;
char StaffTemperament;
char StaffPlayingSquad;
char StaffClassification;
char StaffClubValuation;
CM3_PLAYERS *StaffPlayerData;
CM3_STAFF_PREFERENCES *StaffPreferences; // Version 0x02 - New ptr type
CM3_NON_PLAYERS *StaffNonPlayerData;

// Runtime data //
char StaffSquadSelectedFor; // should be StaffEuroSquadFlag - Kev
};

// ----- //
// Non Players //
// ----- //

struct cm3_non_players
{
long StaffNonPlayerID;
short StaffNonPlayerCurrentAbility;
short StaffNonPlayerPotentialAbility;
short StaffNonPlayerHomeReputation; // Version 0x02 - Changed char->short
short StaffNonPlayerCurrentReputation; // Version 0x02 - Changed char->short
short StaffNonPlayerWorldReputation; // Version 0x02 - Changed char->short
char StaffNonPlayerAttacking;
char StaffNonPlayerBusiness;
char StaffNonPlayerCoaching;
char StaffNonPlayerCoachingGks;
char StaffNonPlayerCoachingTechnique;
char StaffNonPlayerDirectness;
char StaffNonPlayerDiscipline;
char StaffNonPlayerFreeRoles;
char StaffNonPlayerInterference;
char StaffNonPlayerJudgement;
char StaffNonPlayerJudgingPotential;
char StaffNonPlayerManHandling;
char StaffNonPlayerMarking;
char StaffNonPlayerMotivating;
char StaffNonPlayerOffside;
char StaffNonPlayerPatience;
char StaffNonPlayerPhysiotherapy;
char StaffNonPlayerPressing;
char StaffNonPlayerResources;
char StaffNonPlayerTactics;
char StaffNonPlayerYoungsters;
CM3_STAFF *StaffNonPlayerGoalkeeper;
CM3_STAFF *StaffNonPlayerSweeper;
CM3_STAFF *StaffNonPlayerDefender;
CM3_STAFF *StaffNonPlayerDefensiveMidfielder;
CM3_STAFF *StaffNonPlayerMidfielder;
CM3_STAFF *StaffNonPlayerAttackingMidfielder;
CM3_STAFF *StaffNonPlayerAttacker;
CM3_STAFF *StaffNonPlayerWingBack;
char StaffNonPlayerFormationPreferred;
};

// ----- //
// Players //
// ----- //

struct cm3_players
{
// original data
long StaffPlayerID;
char StaffPlayerSquadNumber;
short StaffPlayerCurrentAbility;
short StaffPlayerPotentialAbility;
short StaffPlayerHomeReputation; // Version 0x02 - Changed char->short
short StaffPlayerCurrentReputation; // Version 0x02 - Changed char->short
short StaffPlayerWorldReputation; // Version 0x02 - Changed char->short
char StaffPlayerGoalkeeper;
char StaffPlayerSweeper;
char StaffPlayerDefender;
char StaffPlayerDefensiveMidfielder;
char StaffPlayerMidfielder;
char StaffPlayerAttackingMidfielder;
char StaffPlayerAttacker;
char StaffPlayerWingBack;
char StaffPlayerRightSide;
char StaffPlayerLeftSide;
char StaffPlayerCentral;
char StaffPlayerFreeRole;
char StaffPlayerAcceleration;
char StaffPlayerAggression;
char StaffPlayerAgility;
char StaffPlayerAnticipation;
char StaffPlayerBalance;
char StaffPlayerBravery; // Value

```

```

char StaffPlayerConsistency;
char StaffPlayerCorners;
char StaffPlayerCrossing;
char StaffPlayerDecisions;
char StaffPlayerDirtiness;
char StaffPlayerDribbling;
char StaffPlayerFinishing;
char StaffPlayerFlair;
char StaffPlayerFreeKicks;
char StaffPlayerHandling;
char StaffPlayerHeading;
char StaffPlayerImportantMatches;
char StaffPlayerInjuryProneness;
char StaffPlayerJumping;
char StaffPlayerLeadership;
char StaffPlayerLeftFoot;
char StaffPlayerLongShots;
char StaffPlayerMarking;
char StaffPlayerMovement;
char StaffPlayerNaturalFitness;
char StaffPlayerOneOnOnes;
char StaffPlayerPace;
char StaffPlayerPassing;
char StaffPlayerPenalties;
char StaffPlayerPositioning;
char StaffPlayerReflexes;
char StaffPlayerRightFoot;
char StaffPlayerStamina;
char StaffPlayerStrength;
char StaffPlayerTackling;
char StaffPlayerTeamwork;
char StaffPlayerTechnique;
char StaffPlayerThrowIns;
char StaffPlayerVersatility;
char StaffPlayerVision;
char StaffPlayerWorkRate;

// runtime data
char StaffPlayerMorale;
};

// ----- //
// Player details //
// ----- //

struct cm3_staff_preferences
{
long StaffPreferencesID;
CM3_CLUBS *StaffFavouriteClubs1;
CM3_CLUBS *StaffFavouriteClubs2;
CM3_CLUBS *StaffFavouriteClubs3;
CM3_CLUBS *StaffDislikedClubs1;
CM3_CLUBS *StaffDislikedClubs2;
CM3_CLUBS *StaffDislikedClubs3;
CM3_STAFF *StaffFavouriteStaff1;
CM3_STAFF *StaffFavouriteStaff2;
CM3_STAFF *StaffFavouriteStaff3;
CM3_STAFF *StaffDislikedStaff1;
CM3_STAFF *StaffDislikedStaff2;
CM3_STAFF *StaffDislikedStaff3;
};

// ----- //
// Staff History //
// ----- //

struct cm3_staff_history
{
long StaffHistoryID;
CM3_STAFF *StaffHistoryStaff;
short StaffHistoryYear;
CM3_CLUBS *StaffHistoryClub;
char StaffHistoryOnLoan;
unsigned char StaffHistoryApps;
unsigned char StaffHistoryGoals;
};

// ----- //
// Club Competitions //
// ----- //

struct cm3_club_comps
{
long ClubCompID;
char ClubCompName[ STANDARD_TXT_LENGTH ];
char ClubCompGenderName;
char ClubCompNameShort[ SHORT_TXT_LENGTH ];
char ClubCompGenderNameShort;
char ClubCompNameThreeLetter[ 4 ];
char ClubCompScope;
char ClubCompSelected;
CM3_CONTINENTS *ClubCompContinent;
CM3_NATIONS *ClubCompNation;
CM3_COLOURS *ClubCompForegroundColour;
};

```

```

CM3_COLOURS *ClubCompBackgroundColour;
short ClubCompReputation; // Version 0x02 - Changed char->short
};

// ----- //
// Club Competition History //
// ----- //

struct cm3_club_comp_history
{
long ClubCompHistoryID;
CM3_CLUB_COMPS *ClubCompHistoryClubComp;
short ClubCompHistoryYear;
CM3_CLUBS *ClubCompHistoryWinners;
CM3_CLUBS *ClubCompHistoryRunnersUp;
CM3_CLUBS *ClubCompHistoryThirdPlaced;
CM3_CLUBS *ClubCompHistoryHosts;
};

// ----- //
// Staff Competitions //
// ----- //

struct cm3_staff_comps
{
long StaffCompID;
char StaffCompName[ STANDARD_TXT_LENGTH ];
char StaffCompGenderName;
char StaffCompNameShort[ SHORT_TXT_LENGTH ];
char StaffCompGenderNameShort;
CM3_CONTINENTS *StaffCompContinent;
CM3_NATIONS *StaffCompNation;
CM3_COLOURS *StaffCompForegroundColour;
CM3_COLOURS *StaffCompBackgroundColour;
short StaffCompReputation; // Version 0x02 - Changed char->short
};

// ----- //
// Staff Competition History //
// ----- //

struct cm3_staff_comp_history
{
long StaffCompHistoryID;
CM3_STAFF_COMPS *StaffCompHistoryStaffComp;
short StaffCompHistoryYear;
char *StaffCompHistoryFirstPlacedFirstName;
char *StaffCompHistoryFirstPlacedSecondName;
CM3_STAFF *StaffCompHistoryFirstPlacedIndex;
long StaffCompHistoryFirstPlacedInfo;
char *StaffCompHistorySecondPlacedFirstName;
char *StaffCompHistorySecondPlacedSecondName;
CM3_STAFF *StaffCompHistorySecondPlacedIndex;
long StaffCompHistorySecondPlacedInfo;
char *StaffCompHistoryThirdPlacedFirstName;
char *StaffCompHistoryThirdPlacedSecondName;
CM3_STAFF *StaffCompHistoryThirdPlacedIndex;
long StaffCompHistoryThirdPlacedInfo;
};

// ----- //
// Officials //
// ----- //

struct cm3_officials
{
long OfficialID;
char *OfficialFirstName;
char *OfficialSecondName;
CM_DATE OfficialDateOfBirth;
short OfficialYearOfBirth;
CM3_NATIONS *OfficialNation;
CM3_CITIES *OfficialCity;
short OfficialCurrentAbility;
short OfficialPotentialAbility;
short OfficialReputation;
char OfficialAllowingFlow;
char OfficialDiscipline;
char OfficialImportantMatches;
char OfficialPressure;
char OfficialRefereeing;
char OfficialRunningLine;
char OfficialTimekeeping;
};

// ----- //
// Colours //
// ----- //

struct cm3_colours
{
long ColourID;
char ColourName[ STANDARD_TXT_LENGTH ];
unsigned char ColourRedIntensity;

```

```

unsigned char ColourGreenIntensity;
unsigned char ColourBlueIntensity;
};

// ----- //
// Names //
// ----- //

// The cm3 names structure contains the information upon a name which is used
// by either the staff or officials within championship manager 3.
struct cm3_names
{
char Name[ STANDARD_TXT_LENGTH ];
long NameID;
long Nation;
char count;
};

// reset alignment
#ifdef PLATFORM_MAC
#pragma options align=reset
#endif

// ----- //
// Variables //
// ----- //

#define MAX_INDEX_NAME 50

// Index file struct //
typedef struct
{
char filename[ MAX_INDEX_NAME + 1 ];
long file_id;
long table_sz;
long offset;
long version;
} INDEX_TABLE;

#ifdef DATABASE_CPP
INDEX_TABLE index_info[ NUM_INDEX_TABLES ];

static BOOL game_format_data = FALSE;
static long num_of_friendlyies = 0,
num_of_humans = 0;

// Table sizes //
long cm3_continents_tbl_sz = 0;
long cm3_nations_tbl_sz = 0;
long cm3_cities_tbl_sz = 0;
long cm3_stadiums_tbl_sz = 0;
long cm3_clubs_tbl_sz = 0;
long cm3_nation_clubs_tbl_sz = 0;
long cm3_staff_tbl_sz = 0;
long cm3_non_players_tbl_sz = 0;
long cm3_players_tbl_sz = 0;
long cm3_staff_preferences_tbl_sz = 0;
long cm3_staff_history_tbl_sz = 0;
long cm3_club_comps_tbl_sz = 0;
long cm3_club_comp_history_tbl_sz = 0;
long cm3_nation_club_comps_tbl_sz = 0;
long cm3_nation_club_comp_history_tbl_sz = 0;
long cm3_staff_comps_tbl_sz = 0;
long cm3_staff_comp_history_tbl_sz = 0;
long cm3_officials_tbl_sz = 0;
long cm3_colours_tbl_sz = 0;
long cm3_first_names_list_sz = 0;
long cm3_second_names_list_sz = 0;
long cm3_common_names_list_sz = 0;

char use_real_players = 1;

// Table ptrs //
CM3_CONTINENTS *cm3_continents_tbl = NULL;
CM3_NATIONS *cm3_nations_tbl = NULL;
CM3_CITIES *cm3_cities_tbl = NULL;
CM3_STADIUMS *cm3_stadiums_tbl = NULL;
CM3_CLUBS *cm3_clubs_tbl = NULL;
CM3_CLUBS *cm3_nation_clubs_tbl = NULL;
CM3_STAFF *cm3_staff_tbl = NULL;
CM3_NON_PLAYERS *cm3_non_players_tbl = NULL;
CM3_PLAYERS *cm3_players_tbl = NULL;
CM3_STAFF_PREFERENCES *cm3_staff_preferences_tbl = NULL;
CM3_STAFF_HISTORY *cm3_staff_history_tbl = NULL;
CM3_CLUB_COMPS *cm3_club_comps_tbl = NULL;
CM3_CLUB_COMP_HISTORY *cm3_club_comp_history_tbl = NULL;
CM3_NATION_CLUB_COMPS *cm3_nation_club_comps_tbl = NULL;
CM3_NATION_CLUB_COMP_HISTORY *cm3_nation_club_comp_history_tbl = NULL;
CM3_STAFF_COMPS *cm3_staff_comps_tbl = NULL;
CM3_STAFF_COMP_HISTORY *cm3_staff_comp_history_tbl = NULL;
CM3_OFFICIALS *cm3_officials_tbl = NULL;
CM3_COLOURS *cm3_colours_tbl = NULL;
CM3_NAMES *cm3_first_names_list = NULL;
CM3_NAMES *cm3_second_names_list = NULL;

```

```

CM3_NAMES *cm3_common_names_list = NULL;
#else
// Table sizes //
extern long cm3_continents_tbl_sz;
extern long cm3_nations_tbl_sz;
extern long cm3_cities_tbl_sz;
extern long cm3_stadiums_tbl_sz;
extern long cm3_clubs_tbl_sz;
extern long cm3_nation_clubs_tbl_sz;
extern long cm3_staff_tbl_sz;
extern long cm3_non_players_tbl_sz;
extern long cm3_players_tbl_sz;
extern long cm3_staff_preferences_tbl_sz;
extern long cm3_staff_history_tbl_sz;
extern long cm3_club_comps_tbl_sz;
extern long cm3_club_comp_history_tbl_sz;
extern long cm3_nation_club_comps_tbl_sz;
extern long cm3_nation_club_comp_history_tbl_sz;
extern long cm3_staff_comps_tbl_sz;
extern long cm3_staff_comp_history_tbl_sz;
extern long cm3_officials_tbl_sz;
extern long cm3_colours_tbl_sz;
extern long cm3_first_names_list_sz;
extern long cm3_second_names_list_sz;
extern long cm3_common_names_list_sz;

// Table sizes //
extern CM3_CONTINENTS *cm3_continents_tbl;
extern CM3_NATIONS *cm3_nations_tbl;
extern CM3_CITIES *cm3_cities_tbl;
extern CM3_STADIUMS *cm3_stadiums_tbl;
extern CM3_CLUBS *cm3_clubs_tbl;
extern CM3_CLUBS *cm3_nation_clubs_tbl;
extern CM3_STAFF *cm3_staff_tbl;
extern CM3_NON_PLAYERS *cm3_non_players_tbl;
extern CM3_PLAYERS *cm3_players_tbl;
extern CM3_STAFF_PREFERENCES *cm3_staff_preferences_tbl;
extern CM3_STAFF_HISTORY *cm3_staff_history_tbl;
extern CM3_CLUB_COMPS *cm3_club_comps_tbl;
extern CM3_CLUB_COMP_HISTORY *cm3_club_comp_history_tbl;
extern CM3_CLUB_COMPS *cm3_nation_club_comps_tbl;
extern CM3_CLUB_COMP_HISTORY *cm3_nation_club_comp_history_tbl;
extern CM3_STAFF_COMPS *cm3_staff_comps_tbl;
extern CM3_STAFF_COMP_HISTORY *cm3_staff_comp_history_tbl;
extern CM3_OFFICIALS *cm3_officials_tbl;
extern CM3_COLOURS *cm3_colours_tbl;
extern CM3_NAMES *cm3_first_names_list;
extern CM3_NAMES *cm3_second_names_list;
extern CM3_NAMES *cm3_common_names_list;

extern char use_real_players;
#endif

// Language header structure ..
typedef struct
{
long club_tbl_sz;
long continent_tbl_sz;
long nation_comp_tbl_sz;
long nation_tbl_sz;
long cities_tbl_sz;
long club_comp_tbl_sz;
long stadiums_tbl_sz;
long staff_comp_tbl_sz;
} LANG_HEADER;

// Structure containing club language information
typedef struct
{
char ClubName[ STANDARD_TXT_LENGTH ];
char GenderClubName;
char ClubNameShort[ SHORT_TXT_LENGTH ];
char GenderClubNameShort;
} CLUB_LANG;

// Structure containing continent language information
typedef struct
{
char ContinentName[ SHORT_TXT_LENGTH ];
char GenderContinentName;
char ContinentNameThreeLetter[ 4 ];
char ContinentNameNationality[ SHORT_TXT_LENGTH ];
char ContinentFederationName[ LONG_TXT_LENGTH ];
char GenderContinentFederationName;
char ContinentFederationNameShort[ SHORT_TXT_LENGTH ];
char GenderContinentFederationNameShort;
} CONTINENT_LANG;

// Structure containing nation competition language information
typedef struct
{
char NationCompName[ STANDARD_TXT_LENGTH ];
char GenderNationCompName;
char NationCompNameShort[ SHORT_TXT_LENGTH ];

```

```

char GenderNationCompNameShort;
char NationCompNameThreeLetter[ 4 ];
} NATION_COMP_LANG;

// Structure containing nation language information
typedef struct
{
char NationName[ STANDARD_TXT_LENGTH ];
char GenderNationName;
char NationNameShort[ SHORT_TXT_LENGTH ];
char GenderNationNameShort;
char NationNameThreeLetter[ 4 ];
char NationNameNationality[ SHORT_TXT_LENGTH ];
} NATION_LANG;

// Structure containing city language information
typedef struct
{
char CityName[ STANDARD_TXT_LENGTH ];
char GenderCityName;
} CITY_LANG;

// Structure containing stadium language information
typedef struct
{
char StadiumName[ STANDARD_TXT_LENGTH ];
char GenderStadiumName;
} STADIUM_LANG;

// Structure containing club comp language information
typedef struct
{
char ClubCompName[ STANDARD_TXT_LENGTH ];
char GenderClubCompName;
char ClubCompNameShort[ SHORT_TXT_LENGTH ];
char GenderClubCompNameShort;
char ClubCompNameThreeLetter[ 4 ];
} CLUB_COMP_LANG;

// Structure containing staff language information
typedef struct
{
char StaffCompName[ STANDARD_TXT_LENGTH ];
char GenderStaffCompName;
char StaffCompNameShort[ SHORT_TXT_LENGTH ];
char GenderStaffCompNameShort;
} STAFF_COMP_LANG;

// Class containing English language text for database
class ENGLISH_LANG
{
public:

ENGLISH_LANG( void );
~ENGLISH_LANG( void );

// access functions
char *get_nation_name( CM3 NATIONS *p_nation );
char *get_nation_name_three_letter( CM3 NATIONS *p_nation );
char *get_club_comp_name( CM3 CLUB_COMPS *p_club_comp );
char *get_continent_name( CM3 CONTINENTS *p_continent );
char *get_staff_comp_name( CM3 STAFF_COMPS *p_staff_comp );
char *get_club_name( CM3 CLUBS *p_club );
char *get_club_name_short( CM3 CLUBS *p_club );
char *get_stadium_name( CM3 STADIUMS *p_stadium );

private:
BOOL read_english_database( void );

LANG_HEADER lang_header;
CLUB_LANG *club_lang_data;
CONTINENT_LANG *continent_lang_data;
NATION_COMP_LANG *nation_comp_lang_data;
NATION_LANG *nation_lang_data;
CITY_LANG *city_lang_data;
STADIUM_LANG *stadium_lang_data;
CLUB_COMP_LANG *club_comp_lang_data;
STAFF_COMP_LANG *staff_comp_lang_data;
};

#ifdef DATABASE_CPP
LANGUAGE current_database_language = NO_LANGUAGE_SET;
long *club_remapping_indices = NULL;
#else
extern LANGUAGE current_database_language;
extern long *club_remapping_indices;
#endif

```

Can anyone explain what this all information is about?

Quote:

Originally Posted by **MarcoVanBast** 

Can anyone explain what this all information is about?

It some useful information on outlining the database structure. Which is highly valuable to know as when reading binary data you need to know the expected types to convert the binary array to.