

API User Manual

For

FCTRCARD.DLL

Author: Andrew Burns, From Concept To Reality, L.L.C.

Date: 21 SEP 2000

URL: <http://users.nac.net/fctr>

Table of Contents

INITIALIZATION FUNCTIONS	1
FCTR_REGISTER.....	1
FCTR_ADDSTANDARDDECKTOSHOE	2
FCTR_ADDSTANDARDDOMINOESTOSHOE	3
FCTR_ADDCHINESEDOMINOESTOSHOE	4
FCTR_ADDDECK.....	5
FCTR_ADDCARD.....	6
FCTR_ADDDOMINO	7
FCTR_EMPTYSHOE	8
SHOE INFORMATIONAL FUNCTIONS	9
FCTR_CARDSDEALT.....	9
FCTR_CARDSREMAINING	9
FCTR_TOTALCARDS.....	9
FCTR_TOTALDECKS.....	9
CARD DRAWING FUNCTIONS	10
FCTR_DRAWCARDBACKING	10
FCTR_DRAWCARD	11
DOMINO DRAWING FUNCTIONS.....	12
FCTR_DRAWDOMINOBACKING	12
FCTR_DRAWDOMINO	13
FCTR_DRAWCHINESEDOMINO.....	14
SHOE MANIPULATION FUNCTIONS	15
FCTR_DEALACARD	15
FCTR_DEALADOMINO	16
FCTR_SHUFFLE	17
C++.....	18
INITIALIZATION CONSTANTS	18
RETURN CODE CONSTANTS	19
VISUAL BASIC.....	20
INITIALIZATION CONSTANTS	20
RETURN CODE CONSTANTS	21
DECLARES	22

Initialization Functions

FCTR_Register

The **FCTR_Register** function initializes the DLL with your registration code and unlocks the full functionality.

```
SHORT FCTR_Register (  
    LPCSTR lpzName,           // Registered username  
    LPCSTR lpzRegCode        // Registration code  
);
```

Parameters

lpzName

Registered username

lpzRegCode

Registration code

Return Values

If this function succeeds, the return value is **FCTR_ERROR_OK**.

If this function fails, the return value is less than zero.

Remarks

This function should be the first function called BEFORE any of the other calls. Please note that for EACH executable you will need a new registration code.

WARNING: Calls to this function without the proper registration code will shutdown your machine.

FCTR_AddStandardDeckToShoe

The **FCTR_AddStandardDeckToShoe** function adds a standard 52 card deck to the shoe.

```
SHORT FCTR_AddStandardDeckToShoe (  
    USHORT NumberOfJokersPerDeck           // Number of Jokers to add to the deck  
);
```

Parameters

NumberOfJokersPerDeck

Number of Jokers to add to the deck (0 - MAXSHORT)

Return Values

If this function succeeds, the return value is **FCTR_ERROR_OK**.

If this function fails, the return value is less than zero.

Remarks

This function is used to add a standard 52 card deck to the shoe. Please note that you are limited to 2 decks for unregistered versions.

FCTR_AddStandardDominoesToShoe

The **FCTR_AddStandardDominoesToShoe** function adds a standard set of dominoes to the shoe.

```
SHORT FCTR_AddStandardDominoesToShoe(  
    USHORT MaxNumberOfDots           // Maximum number of dots per domino  
);
```

Parameters

MaxNumberOfDots

Maximum number of dots per domino, usually 6, 9, or 12. (1 - 12)

Return Values

If this function succeeds, the return value is **FCTR_ERROR_OK**.

If this function fails, the return value is less than zero.

Remarks

This function is used to add a standard set of dominoes to the shoe. Please note that you are limited to 1 set for unregistered versions.

FCTR_AddChineseDominoesToShoe

The **FCTR_AddChineseDominoesToShoe** function adds a Chinese set of dominoes to the shoe.

SHORT FCTR_AddChineseDominoesToShoe();

Return Values

If this function succeeds, the return value is **FCTR_ERROR_OK**.

If this function fails, the return value is less than zero.

Remarks

This function is used to add a Chinese set of dominoes to the shoe. Please note that you are limited to 1 set for unregistered versions.

FCTR_AddDeck

The **FCTR_AddDeck** function creates a new, empty deck to the shoe and returns a pointer to the new deck.

```
SHORT FCTR_AddDeck (  
    LPVOID lpDeck          // Pointer to the newly added deck  
);
```

Parameters

lpDeck

Pointer to the newly added deck.

Return Values

If this function succeeds, the return value is **FCTR_ERROR_OK**.

If this function fails, the return value is less than zero.

Remarks

This function should be called BEFORE any calls to **FCTR_AddCard** or **FCTR_AddDomino**. Please note that you are limited to 1 custom deck or set for unregistered versions.

FCTR_AddCard

The **FCTR_AddCard** function adds a new card to a deck.

```
SHORT FCTR_AddCard (  
    LPVOID lpDeck,           // Pointer to the newly added deck  
    USHORT Value,           // Value of the new card  
    LPSTR lpValueName,      // Text value of the new card  
    USHORT Suit,            // Suit of the new card  
    LPSTR lpSuitName        // Text suit of the new card  
);
```

Parameters

lpDeck

Pointer to the newly added deck.

Value

Value of the new card to add.

lpValueName

Text value of the new card to add.

Suit

Suit of the new card to add

lpSuitName

Pointer to a variable to receive the text suit of dealt card.

Return Values

If this function succeeds, the return value is **FCTR_ERROR_OK**.

If this function fails, the return value is less than zero.

Remarks

This function should called AFTER **FCTR_AddDeck**.

FCTR_AddDomino

The **FCTR_AddDomino** function adds a new domino to a deck.

```
SHORT FCTR_AddDomino (  
    LPVOID lpDeck,           // Pointer to the newly added deck  
    USHORT TopValue,         // Top value of the new domino  
    USHORT BottomValue,      // Bottom value of the new domino  
);
```

Parameters

lpDeck

Pointer to the newly added deck.

TopValue

Top value of the new domino to add.

BottomValue

Bottom value of the new domino to add.

Return Values

If this function succeeds, the return value is **FCTR_ERROR_OK**.

If this function fails, the return value is less than zero.

Remarks

This function should be called AFTER **FCTR_AddDeck**.

FCTR_EmptyShoe

The **FCTR_EmptyShoe** function empties the shoe of all cards and dominoes.

```
VOID FCTR_EmptyShoe ();
```

Shoe Informational Functions

FCTR_CardsDealt

FCTR_CardsRemaining

FCTR_TotalCards

FCTR_TotalDecks

These functions return various information about the shoe.

SHORT FCTR_CardsDealt();

SHORT FCTR_CardsRemaining();

SHORT FCTR_TotalCards();

SHORT FCTR_TotalDecks();

Return Values

If the function succeeds, the return value is the information requested.

If the function fails, the return value is less than zero.

Remarks

These functions are used for cards and dominoes. For example, if there are dominoes in the shoe instead of cards, you should use **FCTR_CardsDealt** to find out how many dominoes have been dealt from the shoe.

Card Drawing Functions

FCTR_DrawCardBacking

The **FCTR_DrawCardBacking** function draws the back of a card.

```
SHORT FCTR_DrawCardBacking(  
    USHORT BackingIndex,    // Indicates which card back design to draw  
    HDC hdcDest,            // Device context to draw the card back in  
    USHORT X,               // X offset in pixels  
    USHORT Y,               // Y offset in pixels  
    USHORT Orientation      // Orientation to draw card  
);
```

Parameters

BackingIndex

Indicates which card back design to draw. (0 - 7) Please note that unregistered versions will only draw BackingIndex 0.

hdcDest

Device context to draw the card back in.

X

X offset in pixels.

Y

Y offset in pixels.

Orientation

Orientation to draw card (0 = Upright, 1 = Right 90°, 2 = Upside down, 3 = Left 90°).

Return Values

If this function succeeds, the return value is **FCTR_ERROR_OK**.

If this function fails, the return value is less than zero.

FCTR_DrawCard

The **FCTR_DrawCard** function draws the requested card.

```
SHORT FCTR_DrawCard(  
    USHORT Value,           // Value of card to draw  
    USHORT Suit,           // Suit of card to draw  
    HDC hdcDest,          // Device context to draw the card in  
    USHORT X,              // X offset in pixels  
    USHORT Y,              // Y offset in pixels  
    USHORT Orientation    // Orientation to draw card  
);
```

Parameters

Value

Value of card to draw. (1-14)

Suit

Suit of card to draw. (1-4)

hdcDest

Device context to draw the card in.

X

X offset in pixels.

Y

Y offset in pixels.

Orientation

Orientation to draw card (0 = Upright, 1 = Right 90°, 2 = Upside down, 3 = Left 90°).

Return Values

If this function succeeds, the return value is **FCTR_ERROR_OK**.

If this function fails, the return value is less than zero.

Domino Drawing Functions

FCTR_DrawDominoBacking

The **FCTR_DrawDominoBacking** function draws the back of a domino.

```
SHORT FCTR_DrawDominoBacking(  
    HDC hdcDest,           // Device context to draw the domino back in  
    USHORT X,              // X offset in pixels  
    USHORT Y,              // Y offset in pixels  
    USHORT Orientation     // Orientation to draw domino  
);
```

Parameters

hdcDest

Device context to draw the domino back in.

X

X offset in pixels.

Y

Y offset in pixels.

Orientation

Orientation to draw domino (0 = Upright, 1 = Right 90°, 2 = Upside down, 3 = Left 90°).

Return Values

If this function succeeds, the return value is **FCTR_ERROR_OK**.

If this function fails, the return value is less than zero.

FCTR_DrawDomino

The **FCTR_DrawDomino** function draws the requested domino using traditional American symbols.

```
SHORT FCTR_DrawDomino(  
    USHORT TopValue,           // Top value of domino to draw  
    USHORT BottomValue,        // Bottom value of domino to draw  
    HDC hdcDest,                // Device context to draw the domino in  
    USHORT X,                   // X offset in pixels  
    USHORT Y,                   // Y offset in pixels  
    USHORT Orientation          // Orientation to draw domino  
);
```

Parameters

TopValue

Top value of domino to draw.

BottomValue

Bottom value of domino to draw.

hdcDest

Device context to draw the domino in.

X

X offset in pixels.

Y

Y offset in pixels.

Orientation

Orientation to draw domino (0 = Upright, 1 = Right 90°, 2 = Upside down, 3 = Left 90°).

Return Values

If this function succeeds, the return value is **FCTR_ERROR_OK**.

If this function fails, the return value is less than zero.

FCTR_DrawChineseDomino

The **FCTR_DrawChineseDomino** function draws the requested domino using traditional Chinese symbols.

SHORT FCTR_DrawChineseDomino(

```
    USHORT TopValue,           // Top value of domino to draw
    USHORT BottomValue,        // Bottom value of domino to draw
    HDC hdcDest,               // Device context to draw the domino in
    USHORT X,                  // X offset in pixels
    USHORT Y,                  // Y offset in pixels
    USHORT Orientation          // Orientation to draw domino
);
```

Parameters

TopValue

Top value of domino to draw.

BottomValue

Bottom value of domino to draw.

hdcDest

Device context to draw the domino in.

X

X offset in pixels.

Y

Y offset in pixels.

Orientation

Orientation to draw domino (0 = Upright, 1 = Right 90°, 2 = Upside down, 3 = Left 90°).

Return Values

If this function succeeds, the return value is **FCTR_ERROR_OK**.

If this function fails, the return value is less than zero.

Shoe Manipulation Functions

FCTR_DealACard

The **FCTR_DealACard** function deals the next card from the shoe.

```
SHORT FCTR_DealACard(  
    USHORT* lpValue,           // Pointer to a variable to receive the value of dealt card  
    LPSTR lpValueName,         // Pointer to a variable to receive the text value of dealt card  
    USHORT* lpSuit,            // Pointer to a variable to receive the suit of dealt card  
    LPSTR lpSuitName           // Pointer to a variable to receive the text suit of dealt card  
);
```

Parameters

lpValue

Pointer to a variable to receive the value of dealt card. (1-14)

lpValueName

Pointer to a variable to receive the text value of dealt card.

lpSuit

Pointer to a variable to receive the suit of dealt card. (1-4)

lpSuitName

Pointer to a variable to receive the text suit of dealt card.

Return Values

If this function succeeds, the return value is **FCTR_ERROR_OK**.

If the function fails, the return value is less than zero.

Remarks

This function deals the next card from the shoe.

FCTR_DealADomino

The **FCTR_DealADomino** function deals the next domino from the shoe.

```
SHORT FCTR_DealADomino(  
    USHORT* lpTopValue,      // Pointer to a variable to receive the top value of dealt domino  
    USHORT* lpBottomValue   // Pointer to a variable to receive the bottom value of dealt domino  
);
```

Parameters

lpTopValue

Pointer to a variable to receive the top value of dealt domino.

lpBottomValue

Pointer to a variable to receive the bottom value of dealt domino.

Return Values

If this function succeeds, the return value is **FCTR_ERROR_OK**.

If the function fails, the return value is less than zero.

Remarks

This function deals the next domino from the shoe.

FCTR_Shuffle

The **FCTR_Shuffle** function shuffles all the cards or dominoes in the shoe.

SHORT FCTR_Shuffle();

Return Values

If this function succeeds, the return value is **FCTR_ERROR_OK**.

If the function fails, the return value is less than zero.

Remarks

This function will reset the CardsDealt before shuffling the shoe.

C++

Initialization Constants

```
#define FCTR_SUIT_SPADES      (1)
#define FCTR_SUIT_CLUBS      (2)
#define FCTR_SUIT_HEARTS     (3)
#define FCTR_SUIT_DIAMONDS   (4)

#define FCTR_VALUE_ACE        (1)
#define FCTR_VALUE_TWO        (2)
#define FCTR_VALUE_THREE     (3)
#define FCTR_VALUE_FOUR      (4)
#define FCTR_VALUE_FIVE      (5)
#define FCTR_VALUE_SIX       (6)
#define FCTR_VALUE_SEVEN     (7)
#define FCTR_VALUE_EIGHT     (8)
#define FCTR_VALUE_NINE      (9)
#define FCTR_VALUE_TEN       (10)
#define FCTR_VALUE_JACK      (11)
#define FCTR_VALUE_QUEEN     (12)
#define FCTR_VALUE_KING      (13)
#define FCTR_VALUE_JOKER     (14)

#define FCTR_CARD_PIXELWIDTH  (71)
#define FCTR_CARD_PIXELHEIGHT (96)

#define FCTR_DOMINO_PIXELWIDTH (51)
#define FCTR_DOMINO_PIXELHEIGHT (96)
```

Return Code Constants

```
#define FCTR_ERROR_OK (0)
#define FCTR_ERROR_OUT_OF_MEMORY (-14)
#define FCTR_ERROR_INVALID_PARAMETER (-20)
#define FCTR_ERROR_NULL_POINTER (-21)
#define FCTR_ERROR_EMPTY (-33)
#define FCTR_ERROR_UNREGISTERED (-34)
```

Visual Basic

Initialization Constants

```
Public Const FCTR_SUIT_SPADES = (1)
Public Const FCTR_SUIT_CLUBS = (2)
Public Const FCTR_SUIT_HEARTS = (3)
Public Const FCTR_SUIT_DIAMONDS = (4)
```

```
Public Const FCTR_VALUE_ACE = (1)
Public Const FCTR_VALUE_TWO = (2)
Public Const FCTR_VALUE_THREE = (3)
Public Const FCTR_VALUE_FOUR = (4)
Public Const FCTR_VALUE_FIVE = (5)
Public Const FCTR_VALUE_SIX = (6)
Public Const FCTR_VALUE_SEVEN = (7)
Public Const FCTR_VALUE_EIGHT = (8)
Public Const FCTR_VALUE_NINE = (9)
Public Const FCTR_VALUE_TEN = (10)
Public Const FCTR_VALUE_JACK = (11)
Public Const FCTR_VALUE_QUEEN = (12)
Public Const FCTR_VALUE_KING = (13)
Public Const FCTR_VALUE_JOKER = (14)
```

```
Public Const FCTR_CARD_PIXELWIDTH = (71)
Public Const FCTR_CARD_PIXELHEIGHT = (96)
```

```
Public Const FCTR_DOMINO_PIXELWIDTH = (51)
Public Const FCTR_DOMINO_PIXELHEIGHT = (96)
```

Return Code Constants

```
Public Const FCTR_ERROR_OK = (0)
Public Const FCTR_ERROR_OUT_OF_MEMORY = (-14)
Public Const FCTR_ERROR_INVALID_PARAMETER = (-20)
Public Const FCTR_ERROR_NULL_POINTER = (-21)
Public Const FCTR_ERROR_EMPTY = (-33)
Public Const FCTR_ERROR_UNREGISTERED = (-34)
```

Declares

```
Public Declare Function FCTR_Register Lib "FCTRCard.dll" _
    (ByVal lpszName As String, ByVal lpszRegCode As String) As Integer

Public Declare Function FCTR_AddStandardDeckToShoe Lib "FCTRCard.dll" _
    (ByVal NumberOfJokersPerDeck As Integer) As Integer

Public Declare Function FCTR_CardsDealt Lib "FCTRCard.dll" () As Integer

Public Declare Function FCTR_CardsRemaining Lib "FCTRCard.dll" () As Integer

Public Declare Function FCTR_TotalCards Lib "FCTRCard.dll" () As Integer

Public Declare Function FCTR_DealACard Lib "FCTRCard.dll" _
    (ByRef lpValue As Integer, ByVal lpValueName As String, _
    ByRef lpSuit As Integer, ByVal lpSuitName As String) As Integer

Public Declare Function FCTR_Shuffle Lib "FCTRCard.dll" () As Integer

Public Declare Function FCTR_TotalDecks Lib "FCTRCard.dll" () As Integer

Public Declare Function FCTR_DrawCardBacking Lib "FCTRCard.dll" _
    (ByVal BackingIndex As Integer, ByVal hdcDest As Long, ByVal X As Integer, ByVal Y As Integer) As Integer

Public Declare Function FCTR_DrawCard Lib "FCTRCard.dll" _
    (ByVal Value As Integer, ByVal Suit As Integer, _
    ByVal hdcDest As Long, ByVal X As Integer, ByVal Y As Integer) As Integer

Public Declare Function FCTR_AddDeck Lib "FCTRCard.dll" (ByVal lpDeck As Long) As Integer

Public Declare Function FCTR_AddCard Lib "FCTRCard.dll" _
    (ByVal lpDeck As Long, ByVal Value As Integer, ByVal lpValueName As String, _
    ByVal Suit As Integer, ByVal lpSuitName As String) As Integer

Public Declare Function FCTR_AddStandardDominoesToShoe Lib "FCTRCard.dll" _
    (ByVal MaxNumberOfDots As Integer) As Integer

Public Declare Function FCTR_AddChineseDominoesToShoe Lib "FCTRCard.dll" () As Integer

Public Declare Function FCTR_DrawDominoBacking Lib "FCTRCard.dll" _
    (ByVal hdcDest As Long, ByVal X As Integer, ByVal Y As Integer, ByVal Orientation As Integer) As Integer

Public Declare Function FCTR_DrawDomino Lib "FCTRCard.dll" _
    (ByVal TopValue As Integer, ByVal BottomValue As Integer, ByVal hdcDest As Long, _
    ByVal X As Integer, ByVal Y As Integer, ByVal Orientation As Integer) As Integer

Public Declare Function FCTR_DrawChineseDomino Lib "FCTRCard.dll" _
    (ByVal TopValue As Integer, ByVal BottomValue As Integer, ByVal hdcDest As Long, _
    ByVal X As Integer, ByVal Y As Integer, ByVal Orientation As Integer) As Integer

Public Declare Function FCTR_DealADomino Lib "FCTRCard.dll" _
    (ByRef lpTopValue As Integer, ByRef lpBottomValue As Integer) As Integer

Public Declare Function FCTR_AddDomino Lib "FCTRCard.dll" _
    (ByRef lpDeck As Long, ByVal TopValue As Integer, ByVal BottomValue As Integer) As Integer

Public Declare Sub FCTR_EmptyShoe Lib "FCTRCard.dll" ()
```