

รายการอ้างอิง

1. Silver, B. Bringing paper to life : Adobe Acrobat Capture Unlocks Corporate Memory. Massachusetts: Bruce Silver Associates, 1997.
2. Arps, R. B., and Truong, T. K. Comparison of international standards for lossless still image compression. Proceedings of the IEEE 82 (June 1994): 889-899.
3. Pennebaker, W. B., Mitchell, J. L., Langdon, G. G., Jr., and Arps, R. B. An overview of the basic principle of the Q-coder adaptive binary arithmetic coder. IBM Journal of R&D 32 (November 1988): 717-726.
4. ITU-T Recommendation T.82: Coded Representation of Picture and Audio Information - Progressive Bi-level Image Compression. International Telecommunication Union, 1993.
5. Howard, P. G., and Vitter, J. S. Arithmetic coding for data compression. Proceedings of the IEEE 82 (June 1994): 857-865.
6. Mark, N. The Data Compression Book. M&T BOOKS, 1991.
7. Abdat, M., and Bellanger, M. G. Combining Gray coding and JBIG for lossless image compression. IEEE Computer Society Press 3 (1994): 851-855.
8. Murray, W. H., and Pappas, C. H. Cross-platform programming for Windows. California: Osborne McGraw-Hill, 1996.
9. Kuhn, M. JBIG kit Version 0.9 [Machine readable data file]. Germany: University of Erlangen-Nurnberg , 1996.
10. Netscape plug-in SDK [Machine readable data file]. Netscape Communications Corporation, 1996.
11. Oliphant, Z. Programming: Netscape plug-ins. 1st ed. Indiana: Sams.net Publishing, 1996.

ภาคผนวก

ภาคผนวก ก

JBIG-kit library

JBIG kit พัฒนาโดย Markus Kuhn จากมหาวิทยาลัย Erlangen-Numberg ประเทศเยอรมัน เป็นฟรีซอฟต์แวร์ภายใต้ GNU (General Public License) พัฒนาด้วยภาษา ANSI/ISO C และประกอบด้วยไฟล์ดังต่อไปนี้

- jbig.h : เป็นไฟล์ header สำหรับไลบรารีการบีบอัดข้อมูลภาพ JBIG
- jbig_tab.c : ประกอบด้วยตารางต่างๆ เช่น ตารางการประมาณค่าความน่าจะเป็น (Probability estimation) สำหรับส่วนถอดและเข้ารหัสภาพแบบอริธเมติก (ตามมาตรฐาน ITU T.82 ตารางที่ 24) ตาราง Resolution reduction table (ตามมาตรฐาน ITU-T T.82 ตารางที่ 17) ตาราง Deterministic prediction tables (ตามมาตรฐาน ITU-T T.82 ตารางที่ 19 to 22)
- jbig.c : ไลบรารีการบีบอัดข้อมูลภาพ JBIG

ฟังก์ชันในการเข้ารหัสและถอดรหัสภาพ มีดังนี้

ฟังก์ชันสำหรับการเข้ารหัสภาพ

```
void jbg_enc_init(struct jbg_enc_state *s, unsigned long x, unsigned long y, int pl, unsigned char **p, void (*data_out)(unsigned char *start, size_t len, void *file), void *file);
```

จุดประสงค์ : เช็ตค่าเริ่มต้นให้กับโครงสร้างของส่วนเข้ารหัส (jbg_enc_state)

พารามิเตอร์ :

s เป็นพอยเตอร์ชี้ไปยังโครงสร้างของส่วนเข้ารหัส

x ความกว้างของภาพ

y ความสูงของภาพ

pl จำนวนระนาบิตของภาพ

p เป็นพอยเตอร์ชี้ไปยังแถวลำดับของพอยเตอร์จำนวน pl ตัว ซึ่งแต่ละตัวชี้ไปยังไบต์แรกของระนาบิต

`data_out` เป็น call-back function ซึ่งจะถูกเรียกระหว่างการบีบอัดข้อมูลเพื่อส่ง BIE ไปให้โปรแกรม พารามิเตอร์ของฟังก์ชัน `data_out` คือตำแหน่งของบล็อกข้อมูลที่ถูกส่ง (`start`) และจำนวนที่ส่ง (`len`)

`file` เป็นพอยเตอร์ที่ผ่านไปยังฟังก์ชัน `data_out()` และในโปรแกรมประเภท multi-threaded แล้ว `data_out()` จะแยกแยะว่าส่วนใดของโปรแกรมเรียกใช้

ค่าส่งกลับ : ไม่มี

```
void jbg_enc_out(struct jbg_enc_state *s);
```

จุดประสงค์ : ส่งผลลัพธ์ BIE ไปยังฟังก์ชัน `data_out()`

พารามิเตอร์ :

`s` เป็นพอยเตอร์ชี้ไปยังโครงสร้างของส่วนเข้ารหัส

ค่าส่งกลับ : ไม่มี

```
void jbg_enc_free(struct jbg_enc_state *s);
```

จุดประสงค์ : เพื่อคืนหน่วยความจำที่จองไว้

พารามิเตอร์ :

`s` เป็นพอยเตอร์ชี้ไปยังโครงสร้างของส่วนเข้ารหัส

ค่าส่งกลับ : ไม่มี

```
void jbg_enc_layers(struct jbg_enc_state *s, int d);
```

จุดประสงค์ : เข้ารหัสแบบ progressive โดยกำหนดจำนวน differential resolution layer ที่ต้องการ

พารามิเตอร์ :

`s` เป็นพอยเตอร์ชี้ไปยังโครงสร้างของส่วนเข้ารหัส

`d` เป็นจำนวน differential resolution layers ที่เข้ารหัส ถ้า `d` มีค่าเป็นลบจะเข้ารหัสแบบ sequential

ค่าส่งกลับ : ไม่มี

```
int jbg_enc_lrlmax(struct jbg_enc_state *s, unsigned long mwidth, unsigned long mheight);
```

จุดประสงค์ : เข้ารหัสแบบ progressive โดยกำหนดขนาดของ lowest resolution layer ที่ต้องการ

พารามิเตอร์ :

s เป็นพอยเตอร์ชี้ไปยังโครงสร้างของส่วนเข้ารหัส

mwidth ความกว้างของภาพสูงสุดในชั้น lowest resolution layer

mheight ความสูงของภาพสูงสุดในชั้น lowest resolution layer

ค่าส่งกลับ : จำนวน differential layer

```
int jbg_enc_lrange(struct jbg_enc_state *s, int dl, int dlh);
```

จุดประสงค์ : กำหนด layer ที่ต้องการส่งไปยัง BIE ในการเข้ารหัสแบบ progressive

พารามิเตอร์ :

s เป็นพอยเตอร์ชี้ไปยังโครงสร้างของส่วนเข้ารหัส

dl คือ layer ที่มีความละเอียดต่ำสุดที่จะส่งไปยัง BIE

dh คือ layer ที่มีความละเอียดสูงสุดที่จะส่งไปยัง BIE

ค่าส่งกลับ : จำนวน differential layer

หมายเหตุ : จะต้องเรียกหลัง jbg_enc_layers หรือ jbg_enc_lrlmax และถ้าพารามิเตอร์ใดมีค่าเป็นลบพารามิเตอร์นั้นจะถูกละเลย

```
void jbg_enc_options(struct jbg_enc_state *s, int order, int options, long l0, int mx, int my);
```

จุดประสงค์ : กำหนดเลือกของอัลกอริทึม JBIG ในการเข้ารหัส

พารามิเตอร์ :

s เป็นพอยเตอร์ชี้ไปยังโครงสร้างของส่วนเข้ารหัส

order ประกอบด้วยบิต JBG_HITOLO, JBG_SEQ, JBG_ILEAVE และ JBG_SMID เป็นพารามิเตอร์จะใช้กำหนดลำดับ SDE ที่เก็บใน BIE

options ประกอบด้วยบิต JBG_LRLTWO, JBG_TPDON, JBG_TPBN, JBG_DPON และ JBG_DELAY_AT เป็นพารามิเตอร์ที่กำหนดว่าจะให้ส่วนใดในการเข้ารหัสทำงานบ้าง

l0 จำนวนบรรทัดต่อแถบภาพใน resolution layer 0

mx ค่าออฟเซตตามแนวนอนสูงสุดของ adaptive template

my ค่าออฟเซตตามแนวตั้งสูงสุดของ adaptive template

ค่าส่งกลับ : ไม่มี

หมายเหตุ : JBIG-KIT ให้ mx มีค่าไม่เกิน 23 ในการเข้ารหัส และไม่เกิน 32 ในการถอดรหัส และ my มีค่าเท่ากับ 0 เท่านั้น (ค่าโดยปริยายของ mx เท่ากับ 8 และ my เท่ากับ 0) JBIG-KIT ไม่สามารถถอดรหัสภาพที่เซต HITOLO, SEQ ได้

ฟังก์ชันสำหรับการถอดรหัสภาพ

```
void jbg_dec_init(struct jbg_dec_state *s);
```

จุดประสงค์ : เช็ตค่าเริ่มต้นให้กับโครงสร้างของส่วนถอดรหัส (jbg_dec_state)

พารามิเตอร์ :

s เป็นพอยเตอร์ชี้ไปยังโครงสร้างของส่วนถอดรหัส

ค่าส่งกลับ : ไม่มี

```
int jbg_dec_in(struct jbg_dec_state *s, unsigned char *data, size_t len, size_t *cnt);
```

จุดประสงค์ : ส่งข้อมูลจากไฟล์ BIE ไปยังส่วนถอดรหัส

พารามิเตอร์ :

s เป็นพอยเตอร์ชี้ไปยังโครงสร้างของส่วนถอดรหัส

data เป็นพอยเตอร์ชี้ไปยังตำแหน่งไบต์แรกของบล็อกข้อมูล

len ความยาวของบล็อกข้อมูล

cnt เป็นพอยเตอร์ชี้ไปยังตัวเลขแสดงจำนวนไบต์ที่อ่านไปจากบล็อกข้อมูล

ค่าส่งกลับ :

JBG_EOK สิ้นสุดการถอดรหัส ไม่มีความผิดพลาดเกิดขึ้น

JBG_EOK_INTR ถอดรหัสถึงขนาดภาพสูงสุดที่กำหนดไว้

JBG_EAGAIN ต้องการข้อมูลเพื่อถอดรหัสภาพอีก

JBG_ENOMEM หน่วยความจำไม่พอ

หมายเหตุ : การส่งข้อมูลให้แก่ jbg_dec_in() ไม่จำเป็นต้องส่งทั้งไฟล์ ส่งที่ละส่วนของไฟล์และเรียกใช้ jbg_dec_in() หลายๆครั้งก็ได้ นอกจากนั้นยังสามารถส่งไฟล์ BIE มากกว่า 1 ไฟล์มาให้ jbg_dec_in() โดย lowest resolution layer ในแต่ละไฟล์ BIE ที่ตามมา จะต้องมามีค่าเป็น highest resolution layer บวกหนึ่ง ของไฟล์ BIE ก่อนหน้า

กรณีที่บล็อกข้อมูลยังไม่ถึง end of the BIE ฟังก์ชัน jbg_dec_in() จะคืนค่า JBG_EAGAIN และ *cnt เท่ากับ len กรณีที่ถึง end of the BIE ฟังก์ชัน jbg_dec_in() จะคืนค่า JBG_EOK แสดงว่าสิ้นสุดการถอดรหัส

ถ้า `jbg_dec_in()` คืนค่า `JBG_EOK_INTR` หรือ `JBG_EOK` แล้ว ค่าของ `*cnt` อาจจะไม่เท่ากับ `len` ซึ่งแสดงว่ายังมีข้อมูลที่ยังไม่ได้รับการถอดรหัสอยู่ใน บั๊ตของข้อมูล การเรียก `jbg_dec_in()` ครั้งต่อไปต้องส่งข้อมูลนี้มาใหม่

```
void jbg_dec_maxsize(struct jbg_dec_state *s, unsigned long xmax, unsigned long ymax);
```

จุดประสงค์ : กำหนดขนาดใหญ่สุดของภาพที่ต้องการถอดรหัส ก่อนที่จะเรียกฟังก์ชัน `jbg_dec_in()` ในครั้งแรก

พารามิเตอร์ :

`s` เป็นพอยเตอร์ชี้ไปยังโครงสร้างของส่วนถอดรหัส

`xmax` ความกว้างสูงสุดของภาพที่ต้องการถอดรหัส

`ymax` ความสูงสูงสุดของภาพที่ต้องการถอดรหัส

ค่าส่งกลับ : ไม่มี

หมายเหตุ : ภาพที่ได้อาจใหญ่กว่าที่กำหนดไว้ เพราะไม่มี resolution layer ที่เหมาะสม ในไฟล์ BIE

```
long jbg_dec_getwidth(struct jbg_dec_state *s);
```

จุดประสงค์ : หาความกว้างของภาพที่ได้จากการถอดรหัส

พารามิเตอร์ :

`s` เป็นพอยเตอร์ชี้ไปยังโครงสร้างของส่วนถอดรหัส

ค่าส่งกลับ : ความกว้างของภาพที่ได้จากการถอดรหัส

```
long jbg_dec_getheight(struct jbg_dec_state *s);
```

จุดประสงค์ : หาความสูงของภาพที่ได้จากการถอดรหัส

พารามิเตอร์ :

`s` เป็นพอยเตอร์ชี้ไปยังโครงสร้างของส่วนถอดรหัส

ค่าส่งกลับ : ความสูงของภาพที่ได้จากการถอดรหัส

```
int jbg_dec_getplanes(struct jbg_dec_state *s);
```

จุดประสงค์ : หาจำนวนระนาบบิตของภาพที่ได้จากการถอดรหัส

พารามิเตอร์ :

`s` เป็นพอยเตอร์ชี้ไปยังโครงสร้างของส่วนถอดรหัส

ค่าส่งกลับ : จำนวนระนาบบิตของภาพที่ได้จากการถอดรหัส

```
unsigned char *jbg_dec_getimage(struct jbg_dec_state *s, int plane);
```

จุดประสงค์ : อ่านข้อมูลภาพที่ได้จากการถอดรหัส

พารามิเตอร์ :

s เป็นพอยเตอร์ชี้ไปยังโครงสร้างของส่วนถอดรหัส

plane ตำแหน่งของระนาบบิต

ค่าส่งกลับ : พอยเตอร์ชี้ไประนาบบิตที่ plane ของภาพที่ได้จากการถอดรหัส

```
long jbg_dec_getsize(struct jbg_dec_state *s);
```

จุดประสงค์ : หาว่าต้องใช้เนื้อที่เท่าใดสำหรับภาพบิตแมพ

พารามิเตอร์ :

s เป็นพอยเตอร์ชี้ไปยังโครงสร้างของส่วนถอดรหัส

ค่าส่งกลับ : จำนวนเนื้อที่สำหรับภาพบิตแมพ

```
void jbg_dec_free(struct jbg_dec_state *s);
```

จุดประสงค์ : คืนหน่วยความจำที่จองไว้เมื่ออ่านภาพที่ถอดรหัสโดย jbg_dec_getimage() เรียบร้อยแล้ว

พารามิเตอร์ :

s เป็นพอยเตอร์ชี้ไปยังโครงสร้างของส่วนถอดรหัส

ค่าส่งกลับ : ไม่มี

ภาคผนวก ข

The Plug-in Application Programming Interface (API)

ฟังก์ชัน API ของ Netscape

```
void NPN_Version( int *plugin_major, int *plugin_minor,  
                 int *netscape_major, int *netscape_minor )
```

จุดประสงค์ : อ่านเวอร์ชันหลัก และรองของ plug-in API และ plug-in ที่พัฒนา

พารามิเตอร์ :

plugin_major พอยเตอร์ชี้ไปยังตัวเลขเวอร์ชันหลักของ plug-in ที่พัฒนา

plugin_minor พอยเตอร์ชี้ไปยังตัวเลขเวอร์ชันรองของ plug-in ที่พัฒนา

netscape_major พอยเตอร์ชี้ไปยังตัวเลขเวอร์ชันหลักของ plug-in API

netscape_minor พอยเตอร์ชี้ไปยังตัวเลขเวอร์ชันรองของ plug-in API

ค่าส่งกลับ : ไม่มี

```
NPError NPN_NewStream( NPP instance, NPMIMEType type,  
                      const char* target, NPStream *stream )
```

จุดประสงค์ : สร้าง stream ใหม่ เพื่อส่งข้อมูลจาก plug-in ไปที่ navigator

พารามิเตอร์ :

instance เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPP

type ชนิดของ MIME type ที่สัมพันธ์กับ stream ใหม่

target หน้าต่างหรือเฟรมเป้าหมาย

stream เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPStream

ค่าส่งกลับ : NPERR_NO_ERROR , ไม่มีความผิดพลาดเกิดขึ้น

NPERR_GENERIC_ERROR , เกิดความผิดพลาดไม่ทราบสาเหตุ

NPERR_INVALID_INSTANCE_ERROR , ไม่มี instance

NPERR_OUT_OF_MEMORY_ERROR , หน่วยความจำไม่พอ

NPError NPN_DestroyStream(NPP instance, NPStream* stream, NPError reason)

จุดประสงค์ : สิ้นสุด data stream ที่สร้างจาก NPN_NewStream

พารามิเตอร์ :

instance เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPP
 stream เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPStream
 reason บอกถึงสาเหตุที่ stream ถูกทำลาย ซึ่งจะมีสาเหตุจาก

- ความผิดพลาดของเน็ตเวิร์ก (NPRES_NETWORK_ERR)
- ผู้ใช้หยุดการส่ง stream (NPRES_USER_BREAK)
- สิ้นสุดการส่ง stream โดยปรกติ (NPRES_DONE)

ค่าส่งกลับ : NPERR_NO_ERROR , ไม่มีความผิดพลาดเกิดขึ้น

NPERR_GENERIC_ERROR , เกิดความผิดพลาดไม่ทราบสาเหตุ

NPERR_INVALID_INSTANCE_ERROR , ไม่มี instance

NPERR_OUT_OF_MEMORY_ERROR , หน่วยความจำไม่พอ

NPError NPN_GetURL (NPP instance, const char* url, const char* window)

จุดประสงค์ : สร้าง stream ใหม่ และอ่านข้อมูลจาก URL

พารามิเตอร์ :

instance เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPP
 url เป็นพอยเตอร์ที่ชี้ไปยังสตริงที่เก็บชื่อ URL
 window เป็นหน้าต่างที่จะแสดงข้อมูล

ค่าส่งกลับ : NPERR_NO_ERROR , ไม่มีความผิดพลาดเกิดขึ้น

NPERR_GENERIC_ERROR , เกิดความผิดพลาดไม่ทราบสาเหตุ

NPERR_INVALID_INSTANCE_ERROR , ไม่มี instance

NPERR_OUT_OF_MEMORY_ERROR , หน่วยความจำไม่พอ

**NPError NPN_PostURL (NPP instance, const char* url, const char* window
 uint32 len, const char* buf, NPBool file)**

จุดประสงค์ : ส่งข้อมูลไปที่ URL

พารามิเตอร์ :

instance เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPP
 url เป็นพอยเตอร์ที่ชี้ไปยังสตริงที่เก็บชื่อ URL

window เป็นหน้าต่างที่จะแสดงข้อมูล

len ความยาวของบัฟเฟอร์

buf ตำแหน่งของบัฟเฟอร์สำหรับเก็บข้อมูลที่จะส่ง

file จะให้อ่านข้อมูลจากไฟล์หรือไม่

ค่าส่งกลับ : NPERR_NO_ERROR , ไม่มีความผิดพลาดเกิดขึ้น

NPERR_GENERIC_ERROR , เกิดความผิดพลาดไม่ทราบสาเหตุ

NPERR_INVALID_INSTANCE_ERROR , ไม่มี instance

NPERR_OUT_OF_MEMORY_ERROR , หน่วยความจำไม่พอ

```
NPError NPN_RequestRead( NPStream* stream, NPByteRange* rangeList)
```

จุดประสงค์ : ต้องการอ่านข้อมูลจาก stream

พารามิเตอร์ :

stream เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPStream

rangeList เป็นพอยเตอร์ที่ชี้ไปยังส่วนหัวของโครงสร้าง NPByteRange

ค่าส่งกลับ : NPERR_NO_ERROR , ไม่มีความผิดพลาดเกิดขึ้น

NPERR_GENERIC_ERROR , เกิดความผิดพลาดไม่ทราบสาเหตุ

```
int32 NPN_Write ( NPP instance, NPStream* stream, int32 len, void* buffer )
```

จุดประสงค์ : เขียนข้อมูลลงใน stream ซึ่งสร้างจาก NPN_NewStream

พารามิเตอร์ :

instance เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPP

stream เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPStream

len ความยาวเป็นจำนวนไบต์ของบัฟเฟอร์ที่ถูกเขียน

buffer ตำแหน่งของบัฟเฟอร์ที่ถูกเขียน

ค่าส่งกลับ : จำนวนไบต์ที่เขียนจริง

```
void* NPN_MemAlloc ( uint32 size )
```

จุดประสงค์ : จองหน่วยความจำ

พารามิเตอร์ :

size ขนาดของหน่วยความจำที่ต้องการจอง

ค่าส่งกลับ : ตำแหน่งของหน่วยความจำที่จัดสรรให้

```
void NPN_MemFree ( void* ptr )
```

จุดประสงค์ : คืนหน่วยความจำ

พารามิเตอร์ :

ptr ตำแหน่งของหน่วยความจำที่จัดสรรโดย NPN_MemAlloc

ค่าส่งกลับ : ไม่มี

```
void NPN_Status ( NPP instance, const char* message )
```

จุดประสงค์ : แสดงสถานะผ่าน status bar ของ Netscape

พารามิเตอร์ :

instance เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPP

message เป็นพอยเตอร์ที่ชี้ไปยังข้อความที่จะแสดง

ค่าส่งกลับ : ไม่มี

```
const char* NPN_UserAgent ( NPP instance )
```

จุดประสงค์ : อ่าน user-agent จาก Navigator

พารามิเตอร์ :

instance เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPP

ค่าส่งกลับ : เป็นพอยเตอร์ที่ชี้ไปยังบัฟเฟอร์ที่เก็บ user-agent ของ Navigator

ฟังก์ชัน API ของ plug-in

```
NPEError NPP_Initialize(void)
```

จุดประสงค์ : กำหนดค่าเริ่มต้นให้แก่ไลบรารีส่วนกลาง(global) ของ plug-in

ค่าส่งกลับ : NPERR_NO_ERROR , ไม่มีความผิดพลาดเกิดขึ้น

NPERR_GENERIC_ERROR , เกิดความผิดพลาดไม่ทราบสาเหตุ

NPERR_OUT_OF_MEMORY_ERROR , หน่วยความจำไม่พอ

หมายเหตุ ฟังก์ชัน NPP_Initialize() จะถูกเรียกใช้เพียงครั้งเดียว ตอนโหลด plug-in

โค้ด

void NPP_Shutdown(void)

จุดประสงค์ : คืนหน่วยความจำกลาง , ปิดไลบรารีส่วนกลางของ plug-in

ค่าส่งกลับ : ไม่มี

NPError NP_LOADDS

NPP_New(NPMIMEType pluginType, NPP instance, uint16 mode,

int16 argc, char* argn[], char* argv[], NPSaveData* saved)

จุดประสงค์ : สร้าง instance data ใหม่ให้แก่ plug-in instance

พารามิเตอร์ :

pluginType ชนิดของ MIME type (plug-in อันหนึ่งสามารถกำหนดให้อ่านข้อมูลได้หลายชนิด, multiple MIME type)

instance เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPP ซึ่งประกอบด้วยพอยเตอร์ที่ชี้ไปยังตำแหน่งของ plug-in's instance และ Netscape's instance *หมายเหตุ* แรกเริ่ม ตำแหน่งของ plug-in's instance จะถูกกำหนดค่าเป็น NULL เราต้องจัดสรรหน่วยความจำให้มัน และให้ตำแหน่งของ plug-in's instance ชี้ไปยังหน่วยความจำที่จัดสรรให้

mode บอกให้รู้ว่าจะให้ plug-in ทำงานในโหมดไหน (embedded , full-page , hidden)

argc มีค่าเท่ากับจำนวนอาร์กิวเมนต์ของเอกสาร HTML

argn เป็นแถวลำดับของพอยเตอร์ที่อ้างถึงค่าแรกของคู่ name/value

argv เป็นแถวลำดับของพอยเตอร์ที่อ้างถึงค่าที่สองของคู่ name/value

saved เป็นพอยเตอร์ที่ชี้ไปยังโครงสร้างที่เรียกว่า NPSaveData

ค่าส่งกลับ : NPERR_NO_ERROR , ไม่มีความผิดพลาดเกิดขึ้น

NPERR_GENERIC_ERROR , เกิดความผิดพลาดไม่ทราบสาเหตุ

NPERR_INVALID_INSTANCE_ERROR , ไม่มี instance

NPERR_OUT_OF_MEMORY_ERROR , หน่วยความจำไม่พอ

NPErrror NP_LOADDS

NPP_Destroy(NPP instance, NPSaveData save)**

จุดประสงค์ : ลบ plug-in instance และคืนหน่วยความจำทั้งหมด

พารามิเตอร์ :

instance เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPP

save เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPSaveData ซึ่งประกอบด้วย ความยาวของข้อมูล และ plug-in instance data

ค่าส่งกลับ : NPERR_NO_ERROR , ไม่มีความผิดพลาดเกิดขึ้น

NPERR_GENERIC_ERROR , เกิดความผิดพลาดไม่ทราบสาเหตุ

NPERR_OUT_OF_MEMORY_ERROR , หน่วยความจำไม่พอ

NPErrror NP_LOADDS

NPP_SetWindow(NPP instance, NPWindow* window)

จุดประสงค์ : บอกให้ plug-in รู้ว่า มีหน้าต่างใหม่ถูกสร้างขึ้นมา หน้าต่างมีการเคลื่อนย้ายหรือเปลี่ยนขนาด หน้าต่างถูกปิด

พารามิเตอร์ :

instance เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPP

window เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPWindow ซึ่ง

ประกอบด้วย แฮนเดิล(handle)ของหน้าต่าง , มุมบนซ้าย , ความกว้างและความสูงของหน้าต่าง clipping rectangle

ค่าส่งกลับ : NPERR_NO_ERROR , ไม่มีความผิดพลาดเกิดขึ้น

NPERR_GENERIC_ERROR , เกิดความผิดพลาดไม่ทราบสาเหตุ

NPERR_INVALID_INSTANCE_ERROR , ไม่มี instance

NPERR_OUT_OF_MEMORY_ERROR , หน่วยความจำไม่พอ

NPErrror NP_LOADDS

```
NPP_NewStream(      NPP instance,
                   NPMIMEType type,
                   NPStream *stream,
                   NPBool seekable,
                   uint16 *stype)
```

จุดประสงค์ : บอกให้ plug-in รู้ว่ามี stream ใหม่

พารามิเตอร์ :

instance เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPP

type ชนิดของ MIME type

stream เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPStream

seekable บอกให้รู้ว่า stream นั้นเป็นชนิด seekable หรือไม่

stype โมดของ stream (NP_NORMAL, NP_SEEK, NP_ASFILE, NP_ASFILEONLY)

ค่าส่งกลับ : NPERR_NO_ERROR , ไม่มีความผิดพลาดเกิดขึ้น

NPERR_GENERIC_ERROR , เกิดความผิดพลาดไม่ทราบสาเหตุ

NPERR_INVALID_INSTANCE_ERROR , ไม่มี instance

NPERR_OUT_OF_MEMORY_ERROR , หน่วยความจำไม่พอ

int32 NP_LOADDS

```
NPP_WriteReady( NPP instance, NPStream *stream )
```

จุดประสงค์ : พิจารณาว่า plug-in พร้อมที่จะรับข้อมูลหรือยัง และจำนวนข้อมูลที่ต้องการรับมีค่าเท่าใด

พารามิเตอร์ :

instance เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPP

stream เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPStream

ค่าส่งกลับ : ความยาวของข้อมูล(เป็นจำนวนไบต์)ที่พร้อมจะรับ

int32 NP_LOADDS

NPP_Write(NPP instance, NPStream *stream, int32 offset, int32 len, void *buffer)

จุดประสงค์ : ส่งข้อมูล(stream) มาให้ โดยมีความยาวของ stream เท่ากับ len ไบต์

พารามิเตอร์ :

instance เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPP

stream เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPStream

offset ค่าออฟเซต(เป็นจำนวนไบต์) ภายใน data stream

len ความยาว(เป็นจำนวนไบต์) ของบัฟเฟอร์

buffer เป็นพอยเตอร์ที่ชี้ไปยังบัฟเฟอร์ที่เก็บ stream data

ค่าส่งกลับ : จำนวนไบต์ที่อ่านจาก stream หรือค่าลบถ้าต้องการหยุดส่ง stream

NPEError NP_LOADDS

NPP_DestroyStream(NPP instance, NPStream *stream, NPEError reason)

จุดประสงค์ : บอกให้ plug-in รู้ว่า stream ถูกทำลาย โดยปรกติแล้วจะถูกเรียกหลังจาก stream ถูกส่งมาครบแล้ว

พารามิเตอร์ :

instance เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPP

stream เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPStream

reason บอกถึงสาเหตุที่ stream ถูกทำลาย ซึ่งจะมีสาเหตุจาก

- ความผิดพลาดของเน็ตเวิร์ก (NPRES_NETWORK_ERR)

- ผู้ใช้หยุดการส่ง stream (NPRES_USER_BREAK)

- สิ้นสุดการส่ง stream โดยปรกติ (NPRES_DONE)

ค่าส่งกลับ : NPERR_NO_ERROR , ไม่มีความผิดพลาดเกิดขึ้น

NPERR_GENERIC_ERROR , เกิดความผิดพลาดไม่ทราบสาเหตุ

NPERR_INVALID_INSTANCE_ERROR , ไม่มี instance

NPERR_OUT_OF_MEMORY_ERROR , หน่วยความจำไม่พอ


```
void NP_LOADDS
```

```
NPP_StreamAsFile( NPP instance, NPStream *stream, const char* fname )
```

จุดประสงค์ : ส่งข้อมูล ชื่อไฟล์ มาให้ สำหรับ plug-in ชนิด filebased

พารามิเตอร์ :

instance เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPP

stream เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPStream

fname เป็นพอยเตอร์ที่ชี้ไปยังชื่อไฟล์ (รวมไดเรกทอรี)

ค่าส่งกลับ : ไม่มี

```
void NP_LOADDS
```

```
NPP_Print( NPP instance, NPPrint* printInfo )
```

จุดประสงค์ : จะถูกเรียกเมื่อผู้ใช้งานต้องการพิมพ์เอกสารขณะที่ plug-in กำลังโหลดอยู่

หมายเหตุ plug-in สามารถเป็นได้ทั้งชนิด NP_EMBED ซึ่ง navigator จะทำการพิมพ์พร้อมกับ plug-in หรือ NP_FULL ซึ่ง plug-in จะจัดการเกี่ยวกับการพิมพ์ทั้งหมด

พารามิเตอร์ :

instance เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPP

printInfo เป็นพอยเตอร์ที่ชี้ไปยังตำแหน่งโครงสร้างของ NPPrint ซึ่งประกอบด้วยโมดของ plug-in , โครงสร้างของ NPFullPrint หรือ NPEmbedPrint (ขึ้นอยู่กับ โมดของ plug-in)

ค่าส่งกลับ : ไม่มี

ประวัติผู้เขียน



นายภาณุมาศ หาดทรายทอง เกิดวันที่ 14 กันยายน พ.ศ. 2516 จังหวัดนครสวรรค์ จบ
การศึกษาระดับปริญญาตรี สาขาวิศวกรรมศาสตรบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์
มหาวิทยาลัย เมื่อปี พ.ศ. 2538 ระหว่างการศึกษาได้เคยฝึกงานที่แผนกประสิทธิภาพ โรงไฟฟ้า
บางปะกง การไฟฟ้าฝ่ายผลิตแห่งประเทศไทย

พ.ศ. 2538 ได้เข้าศึกษาต่อระดับปริญญาโท สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชา
วิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย