

1. 3 Input FLIST

COMMIT_CUSTOMER is a wrapper opcodes that calls multiple opcode in between, no need to create account

NAI_OP_COMMIT_CUSTOMER input flist

```
0 PIN_FLD_POID      POID [0] 0.0.0.1 /plan PLAN_NUM 0
0 PIN_FLD_SERVICES  ARRAY [0] allocated 20, used 12
1  PIN_FLD_PASSWD_CLEAR STR [0] "MY_PW"
1  PIN_FLD_LOGIN      STR [0] "MY_LOGIN"
1  PIN_FLD_SUBSCRIPTION_OBJ POID [0] NULL poid pointer
1  PIN_FLD_SERVICE_ID  STR [0] ""
1  PIN_FLD_SUBSCRIPTION_INDEX INT [0] 0
1  PIN_FLD_BAL_INFO_INDEX INT [0] 0
1  PIN_FLD_SERVICE_CODES ARRAY [0] allocated 20, used 1
2  PIN_FLD_STATE_ID    INT [0] 0
1  PIN_FLD_INHERITED_INFO SUBSTRUCT [0] allocated 20, used 1
1  PIN_FLD_DEALS       ARRAY [0] allocated 20, used 16
2  PIN_FLD_CODE        STR [0] "DEAL_CODE"
2  PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account 1 0
2  PIN_FLD_DEAL_OBJ    POID [0] 0.0.0.1 /deal DEAL_NUM 0
2  PIN_FLD_SUBSCRIPTION_TERMS_OBJ POID [0] 0.0.0.0 0 0
2  PIN_FLD_PERMITTED   STR [0] "/service/nextai"
2  PIN_FLD_WRITE_ACCESS STR [0] "S"
2  PIN_FLD_READ_ACCESS STR [0] "B"
2  PIN_FLD_NAME        STR [0] "DEAL_NAME"
2  PIN_FLD_FLAGS       INT [0] 0
2  PIN_FLD_PRODUCTS    ARRAY [0] allocated 26, used 26
3  PIN_FLD_PURCHASE_END_T TSTAMP [0] (0) <null>
3  PIN_FLD_PURCHASE_START_T TSTAMP [0] (0) <null>
3  PIN_FLD_USAGE_END_DETAILS INT [0] 2
3  PIN_FLD_CYCLE_END_DETAILS INT [0] 2
3  PIN_FLD_RENEWAL_MODE INT [0] 0
3  PIN_FLD_QUANTITY    DECIMAL [0] 1
3  PIN_FLD_PURCHASE_END_DETAILS INT [0] 2
3  PIN_FLD_USAGE_START_DETAILS INT [0] 1
3  PIN_FLD_CYCLE_START_DETAILS INT [0] 1
3  PIN_FLD_PURCHASE_START_DETAILS INT [0] 1
3  PIN_FLD_OWN_MAX     DECIMAL [0] NULL pin_decimal_t ptr
3  PIN_FLD_MODE        ENUM [0] 0
3  PIN_FLD_OWN_MIN     DECIMAL [0] NULL pin_decimal_t ptr
```

```

3     PIN_FLD_NAME          STR [0] "PRODUCT_NAME"
3     PIN_FLD_PRODUCT_OBJ   POID [0] 0.0.0.1 /product PRODUCT_NUM 5
3     PIN_FLD_DESCR         STR [0] "PRODUCT_DESCR"
3     PIN_FLD_USAGE_DISCOUNT DECIMAL [0] 0
3     PIN_FLD_CYCLE_DISCOUNT DECIMAL [0] 0
3     PIN_FLD_PURCHASE_DISCOUNT DECIMAL [0] 0
3     PIN_FLD_STATUS        ENUM [0] 1
3     PIN_FLD_GRACE_PERIOD   INT [0] 0
3     PIN_FLD_STATUS_FLAGS   INT [0] 0
3     PIN_FLD_USAGE_END_T    TSTAMP [0] (0) <null>
3     PIN_FLD_USAGE_START_T  TSTAMP [0] (0) <null>
3     PIN_FLD_CYCLE_END_T    TSTAMP [0] (0) <null>
3     PIN_FLD_CYCLE_START_T  TSTAMP [0] (0) <null>
2     PIN_FLD_MOD_T          TSTAMP [0] (1750278497) Wed Jun 18 16:28:17 2025
2     PIN_FLD_CREATED_T      TSTAMP [0] (1750278228) Wed Jun 18 16:23:48 2025
2     PIN_FLD_DESCR         STR [0] "DEAL_DESCR"
2     PIN_FLD_ATTRIBUTE_OBJ   POID [0] 0.0.0.0 0 0
2     PIN_FLD_END_T          TSTAMP [0] (0) <null>
2     PIN_FLD_START_T        TSTAMP [0] (0) <null>
1     PIN_FLD_BAL_INFO       ARRAY [0] allocated 0, used 0
1     PIN_FLD_DEAL_INFO      SUBSTRUCT [0] allocated 20, used 7
2     PIN_FLD_PRODUCTS       ARRAY [0] allocated 36, used 36
3     PIN_FLD_CYCLE_START_T  TSTAMP [0] (0) <null>
3     PIN_FLD_PURCHASE_END_T TSTAMP [0] (0) <null>
3     PIN_FLD_PURCHASE_START_T TSTAMP [0] (0) <null>
3     PIN_FLD_DESCR         STR [0] ""
3     PIN_FLD_STATUS_FLAGS   INT [0] 0
3     PIN_FLD_USAGE_END_OFFSET INT [0] 0
3     PIN_FLD_USAGE_END_UNIT  INT [0] 0
3     PIN_FLD_CYCLE_END_OFFSET INT [0] 0
3     PIN_FLD_CYCLE_END_UNIT  INT [0] 0
3     PIN_FLD_USAGE_DISCOUNT DECIMAL [0] 0
3     PIN_FLD_PURCHASE_END_OFFSET INT [0] 0
3     PIN_FLD_PURCHASE_END_UNIT  INT [0] 0
3     PIN_FLD_USAGE_START_OFFSET INT [0] 0
3     PIN_FLD_CYCLE_DISCOUNT DECIMAL [0] 0
3     PIN_FLD_USAGE_START_UNIT  INT [0] 0
3     PIN_FLD_CYCLE_START_OFFSET INT [0] 0
3     PIN_FLD_CYCLE_START_UNIT  INT [0] 0
3     PIN_FLD_GRACE_PERIOD   INT [0] 0
3     PIN_FLD_PURCHASE_START_OFFSET INT [0] 0
3     PIN_FLD_PURCHASE_DISCOUNT DECIMAL [0] 0
3     PIN_FLD_PURCHASE_START_UNIT  INT [0] 0

```

```

3     PIN_FLD_USAGE_END_DETAILS  INT [0] 2
3     PIN_FLD_CYCLE_END_DETAILS  INT [0] 2
3     PIN_FLD_PURCHASE_END_DETAILS  INT [0] 2
3     PIN_FLD_USAGE_START_DETAILS  INT [0] 1
3     PIN_FLD_CYCLE_START_DETAILS  INT [0] 1
3     PIN_FLD_PURCHASE_START_DETAILS  INT [0] 1
3     PIN_FLD_PLAN_OBJ    POID [0] 0.0.0.1 /plan PLAN_NUM 0
3     PIN_FLD_PRODUCT_OBJ  POID [0] 0.0.0.1 /product PRODUCT_NUM 5
3     PIN_FLD_QUANTITY    DECIMAL [0] 1
3     PIN_FLD_RENEWAL_MODE  INT [0] 0
3     PIN_FLD_MODE         ENUM [0] 0
3     PIN_FLD_STATUS       ENUM [0] 1
3     PIN_FLD_USAGE_END_T  TSTAMP [0] (0) <null>
3     PIN_FLD_USAGE_START_T TSTAMP [0] (0) <null>
3     PIN_FLD_CYCLE_END_T  TSTAMP [0] (0) <null>
2     PIN_FLD_NAME        STR [0] "DEAL_NAME"
2     PIN_FLD_POID        POID [0] 0.0.0.1 /deal DEAL_NUM 0
2     PIN_FLD_END_T       TSTAMP [0] (0) <null>
2     PIN_FLD_FLAGS       INT [0] 0
2     PIN_FLD_START_T     TSTAMP [0] (0) <null>
2     PIN_FLD_DESCR       STR [0] ""
1     PIN_FLD_SERVICE_OBJ  POID [0] 0.0.0.1 /service/nextai -1 0
0 PIN_FLD_SUBSCRIPTION_TERMS_OBJ  POID [0] 0.0.0.0 / 0 0
0 PIN_FLD_MOD_T          TSTAMP [0] (1750278497) Wed Jun 18 16:28:17 2025
0 PIN_FLD_CREATED_T      TSTAMP [0] (1750278497) Wed Jun 18 16:28:17 2025
0 PIN_FLD_CODE           STR [0] "PLAN_CODE"
0 PIN_FLD_WRITE_ACCESS   STR [0] "S"
0 PIN_FLD_NAMEINFO       ARRAY [1] allocated 20, used 9
1     PIN_FLD_LAST_NAME   STR [0] "MY_LAST"
1     PIN_FLD_FIRST_NAME  STR [0] "MY_FIRST"
1     PIN_FLD_CONTACT_TYPE STR [0] "Account holder"
1     PIN_FLD_EMAIL_ADDR  STR [0] "MY_EMAIL@gmail.com"
1     PIN_FLD_COUNTRY     STR [0] "MY_COUNTRY"
1     PIN_FLD_ZIP         STR [0] "MY_ZIP"
1     PIN_FLD_STATE       STR [0] "MY_STATE"
1     PIN_FLD_CITY        STR [0] "MY_CITY"
1     PIN_FLD_ADDRESS     STR [0] "MY_ADDR"
0 PIN_FLD_READ_ACCESS    STR [0] "B"
0 PIN_FLD_NAME           STR [0] "PLAN_NAME"
0 PIN_FLD_ATTRIBUTE_OBJ  POID [0] 0.0.0.0 0 0
0 PIN_FLD_DESCR          STR [0] "PLAN_DESCR"
0 PIN_FLD_FLAGS          INT [0] 0
0 PIN_FLD_LOCALES        ARRAY [1] allocated 20, used 1

```

```

1  PIN_FLD_LOCALE      STR [0] "en_US"
0 PIN_FLD_BILLINFO     ARRAY [0] allocated 20, used 5
1  PIN_FLD_POID        POID [0] 0.0.0.1 /billinfo -1 0
1  PIN_FLD_BILLINFO_ID STR [0] "Bill Unit(1)"
1  PIN_FLD_PAYINFO     ARRAY [0]  NULL array ptr
1  PIN_FLD_BAL_INFO    ARRAY [0]  NULL array ptr
1  PIN_FLD_PAY_TYPE    ENUM [0] 10001
0 PIN_FLD_BAL_INFO     ARRAY [0] allocated 20, used 3
1  PIN_FLD_NAME        STR [0] "Balance Group<Account>"
1  PIN_FLD_POID        POID [0] 0.0.0.1 /balance_group -1 0
1  PIN_FLD_BILLINFO     ARRAY [0]  NULL array ptr
0 PIN_FLD_PAYINFO      ARRAY [0] allocated 20, used 6
1  PIN_FLD_INV_TYPE    ENUM [0] 257
1  PIN_FLD_NAME        STR [0] "Invoice1"
1  PIN_FLD_POID        POID [0] 0.0.0.1 /payinfo/invoice -1 0
1  PIN_FLD_INHERITED_INFO SUBSTRUCT [0] allocated 20, used 1
2    PIN_FLD_INV_INFO  ARRAY [0] allocated 20, used 10
3      PIN_FLD_DELIVERY_PREFER ENUM [0] 0
3      PIN_FLD_NAME     STR [0] "MY_FIRST MY_LAST"
3      PIN_FLD_INV_TERMS  ENUM [0] 0
3      PIN_FLD_DELIVERY_DESCR STR [0] "MY_EMAIL@gmail.com"
3      PIN_FLD_EMAIL_ADDR STR [0] ""
3      PIN_FLD_COUNTRY   STR [0] "MY_COUNTRY"
3      PIN_FLD_ZIP       STR [0] "MY_ZIP"
3      PIN_FLD_STATE     STR [0] "MY_STATE"
3      PIN_FLD_CITY      STR [0] "MY_CITY"
3      PIN_FLD_ADDRESS   STR [0] "MY_ADDR"
1  PIN_FLD_FLAGS       INT [0] 1
1  PIN_FLD_PAY_TYPE    ENUM [0] 10001
0 PIN_FLD_ACCTINFO     ARRAY [0] allocated 20, used 4
1  PIN_FLD_POID        POID [0] 0.0.0.1 /account -1 0
1  PIN_FLD_BAL_INFO    ARRAY [0]  NULL array ptr
1  PIN_FLD_CURRENCY     INT [0] 840
1  PIN_FLD_BUSINESS_TYPE ENUM [0] 1

```

2. Create a header file and define your new opcode

- Go to the BRM HOME directory
opt

- Go to the BRM opcode directory. BRM opcodes and their numbers are defined in the /ops*.h directory.
cd include/ops
- Create a new header file to house your new custom opcode.
vi nai_cust_ops.h
- Add your new opcode

```
#define NAI_OP_COMMIT_CUSTOMER 280204
```

3. Create memory-mapping extension file

```
cd include/ops
/opt/app/brm/congero-XX/BRM/bin/parse_custom_ops_fields.pl -L pcmc -I nai_cust_ops.h -O
nai_cust_ops.dat
```

4. Add extension file to CM and sys/test pin.conf

- ```
vi pin.conf
```
- Add entry to bottom of pin.conf  
- - ops\_fields\_extension\_file /opt/app/brm/congero-XX/BRM/include/ops/nai\_cust\_ops.dat

### 5. Create Facility Module and create FM config and Function

```
cd /source/sys
mkdir fm_nai_pol
cd fm_nai_pol
```

### 6. Create an Opcode-to-Function Mapping File

- ```
cp ../fm_order_pol/fm_order_pol_config.c fm_nai_pol_config.c
```
- Edit fm_nai_pol_config.c and define your mapping

```

-/*
* Copyright (c) 2002, 2014, Oracle and/or its affiliates. All rights reserved.
*
* This material is the confidential property of Oracle Corporation
* or its licensors and may be used, reproduced, stored or transmitted
* only in accordance with a valid Oracle license or sublicense agreement.
*
*/

```

```

#ifndef lint
static const char Sccs_id[] = "@(#) $Id: fm_nai_pol_config.c /cgbubrm_7.5.0.portalbase/1
2015/11/27 04:09:40 nishahan Exp $";
#endif

#include <stdio.h>
#include "ops/nai_cust_ops.h"
#include "pcm.h"
#include "cm_fm.h"

#define FILE_LOGNAME "fm_nai_pol_config.c"

PIN_EXPORT void * fm_nai_pol_config_func();

struct cm_fm_config fm_nai_pol_config[] = {
{ NAI_OP_COMMIT_CUSTOMER, "op_nai_pol_commit_customer" },
{ 0, (char *)0 }
};

void *
fm_nai_pol_config_func()
{
    return ((void *) (fm_nai_pol_config));
}

```

7. Write Function to Implement New Opcode

Vi fm_nai_pol_commit_customer.c

- Add the following code

```

/*
 * @(#) % %
 *
 * Copyright (c) 2002 - 2006 Oracle. All rights reserved.
 *
 * This material is the confidential property of Oracle Corporation
 * or its licensors and may be used, reproduced, stored or transmitted
 * only in accordance with a valid Oracle license or sublicense agreement.
 *
 */

```

```

#ifndef lint
static const char Sccs_id[] = "@(#) % %";
#endif

/*****m
 * Contains the NAI_OP_COMMIT_CUSTOMER operation.
 *****/

#include <stdio.h>
#include <string.h>

#include <pcm.h>
#include <pinlog.h>

#define FILE_LOGNAME "fm_nai_pol_commit_customer.c(1)"

#include "ops/nai_cust_ops.h"
#include "cm_fm.h"
#include "pin_errs.h"

/*****
 * Routines contained within.
 *****/
EXPORT_OP void
op_nai_pol_commit_customer(
    cm_nap_connection_t  *connp,
    int32                opcode,
    int32                flags,
    pin_flist_t          *i_flistp,
    pin_flist_t          **r_flistpp,
    pin_errbuf_t         *ebufp);

static void
fm_nai_pol_commit_customer(
    pcm_context_t        *ctxp,
    int32                flags,
    pin_flist_t          *i_flistp,
    pin_flist_t          **r_flistpp,
    pin_errbuf_t         *ebufp);

/*****
 * Routines needed from elsewhere.

```

```

*****/

/*****
* Main routine for the PCM_OP_ORDER_POL_CREATE operation.
*****/

void
op_nai_pol_commit_customer(
    cm_nap_connection_t    *connp,
    int32                  opcode,
    int32                  flags,
    pin_flist_t            *i_flistp,
    pin_flist_t            **r_flistpp,
    pin_errbuf_t            *ebufp)
{
    pcm_context_t          *ctxp = connp->dm_ctx;
    poid_t                  *acct_poidp = NULL;

    if (PIN_ERR_IS_ERR(ebufp))
        return;
    PIN_ERR_CLEAR_ERR(ebufp);

    /*****
    * Null out results
    *****/
    *r_flistpp = NULL;

    /*****
    * Insanity check.
    *****/
    if (opcode != NAI_OP_COMMIT_CUSTOMER) {
        pin_set_err(ebufp, PIN_ERRLOC_FM,
            PIN_ERRCLASS_SYSTEM_DETERMINATE,
            PIN_ERR_BAD_OPCODE, 0, 0, opcode);
        PIN_ERR_LOG_EBUF(PIN_ERR_LEVEL_ERROR,
            "op_nai_pol_commit_customer opcode error", ebufp);

        return;
    }

    /*****
    * Debut what we got.
    *****/

```



```

*****/
PIN_ERR_LOG_FLIST(PIN_ERR_LEVEL_DEBUG,
    "op_nai_pol_commit_customer input flist", i_flistp);

/*****
 * Main routine for this opcode
 *****/
fm_nai_pol_commit_customer(ctxp, flags, i_flistp, r_flistpp, ebufp);

/*****
 * Error?
 *****/
if (PIN_ERR_IS_ERR(ebufp)) {
    PIN_FLIST_DESTROY_EX(r_flistpp, ebufp);
    PIN_ERR_LOG_EBUF(PIN_ERR_LEVEL_ERROR,
        "op_nai_pol_commit_customer error", ebufp);
} else {
    PIN_ERR_LOG_FLIST(PIN_ERR_LEVEL_DEBUG,
        "op_nai_pol_commit_customer output flist", *r_flistpp);
}

return;
}

/*****
 * fm_nai_pol_read:
 *
 *****/
static void
fm_nai_pol_commit_customer(
    pcm_context_t      *ctxp,
    int32               flags,
    pin_flist_t         *i_flistp,
    pin_flist_t         **r_flistpp,
    pin_errbuf_t        *ebufp)
{
    poid_t              *order_poidp = NULL;
    char                 *poid_type = NULL;

    if (PIN_ERR_IS_ERR(ebufp)) {
        return;
    }

```

```

    }
    PIN_ERR_CLEAR_ERR(ebufp);

    PIN_ERR_LOG_FLIST(PIN_ERR_LEVEL_DEBUG,
        "op_nai_pol_commit_customer input flist", i_flistp);

    PCM_OP(ctxp, PCM_OP_CUST_COMMIT_CUSTOMER, flags, i_flistp, r_flistpp, ebufp);

    PIN_ERR_LOG_FLIST(PIN_ERR_LEVEL_DEBUG,
        "op_nai_pol_commit_customer output flist", *r_flistpp);

    PIN_ERR_LOG_MSG(PIN_ERR_LEVEL_ERROR, "MADE IT INSIDE CUSTOM
    OPCODE!");

    if (PIN_ERR_IS_ERR(ebufp)) {
        PIN_ERR_LOG_EBUF(PIN_ERR_LEVEL_ERROR,
            "op_nai_pol_read error", ebufp);
    }
    return;
}

```

8. Create a Makefile

vi Makefile

- Add the following

```

#
#
# Copyright (c) 2002, 2017, Oracle and/or its affiliates. All rights reserved.
#
# This material is the confidential property of Oracle Corporation
# or its licensors and may be used, reproduced, stored or transmitted
# only in accordance with a valid Oracle license or sublicense agreement.
#

```

OS=linux

PIN_VERSION=7.5

#####

PINDIR=../..

INCDIR=\$(PINDIR)/include

#####

CC_solaris = cc

CC_hpux = cc

CC_hpux_ia64 = cc

CC_aix = xlc_r

CC_linux = gcc

CC = \$(CC_\$(OS))

CFLAGS_solaris= -g

CFLAGS_hpux= -g -Ae +Z +DAportable

CFLAGS_hpux_ia64= -g -Ae +Z

CFLAGS_aix= -g -Drs6000 -D__aix -Dunix -D__unix

CFLAGS_linux= -m32 -fPIC

CFLAGS= \$(CFLAGS_\$(OS)) -DFLIST_HEAP

CPPFLAGS = -I\$(INCDIR)

LDFLAGS_solaris= -G

LDFLAGS_hpux= -b

LDFLAGS_hpux_ia64= -b

LDFLAGS_aix= -G -bnoentry -bexpall -brtl -lc

LDFLAGS_linux= -m elf_i386 -shared -L/usr/lib

LDFLAGS = \$(LDFLAGS_\$(OS))

```
SL_EXT_solaris= so
SL_EXT_hpux= sl
SL_EXT_hpux_ia64= so
SL_EXT_aix= a
SL_EXT_linux= so
SL_EXT= $(SL_EXT_$(OS))
```

```
OS_LIBS_solaris= -lm
OS_LIBS_hpux= -lsec -lm
OS_LIBS_hpux_ia64= -lm
OS_LIBS_aix= -lm
OS_LIBS_linux= -lm
OS_LIBS= $(OS_LIBS_$(OS))
```

```
####
```

```
INCFILES= $(INCDIR)/pcm.h \
    $(INCDIR)/pin_errs.h $(INCDIR)/pinlog.h \
    $(INCDIR)/pin_flds.h
```

```
FILES= fm_nai_pol_config.c \
    fm_nai_pol_commit_customer.c
```

```
OBJECTS= fm_nai_pol_config.o \
    fm_nai_pol_commit_customer.o
```

```
#
```

```
LIBORDER=fm_nai_pol_custom.$(SL_EXT)
```

```
DEP_linux = $(FILES)
```

```
DEP = $(DEP_$(OS))
```

```
#####
```

```
all: $(LIBORDER)
```

```
clean:
```

```
    rm -f $(OBJECTS) core
```

```
clobber: clean
```

```
    rm -f $(LIBORDER)
```

```
lint:
```

```
    lint $(CPPFLAGS) $(FILES)
```

```
tags: FRC
```

```
#####
```

```
$(LIBORDER): $(OBJECTS) $(INCFILES) Makefile $(DEP)
```

```
    $(LD) -o $(LIBORDER) $(LDFLAGS) $(OBJECTS) $(OS_LIBS)
```

```
$(OBJECTS): $(INCFILES) Makefile
```

```
    $(CC) -c $(CFLAGS) $(CPPFLAGS) $(FILES)
```

```
#
```

```
FRC:
```

9. Run make, create new Shared Object, copy .so to lib

make clean

make

```
mv fm_nai_pol_custom.so $PIN_HOME/lib
```

10. Update CM pin.conf, Add reference to new shared object (second line is just one line)

```
# define new the FM module(s) for this CM  
- cm fm_module /opt/app/brm/congero-XX/BRM/lib/fm_nai_pol_custom.so fm_nai_pol_config - pin
```

11. Prepare input FLIST

- Important: ALWAYS cp the backup template when create a customer, never modify the template
- Run: testnap filename