SHARE Boston – August 2010

# BCPii for Dummies: Start to finish installation, setup and usage

Steve Warren
swarren@us.ibm.com
z/OS BCPii Design and Development

This presentation is for educational purposes only.
It is not a deliverable to customers.

© 2010 IBM Corporation

# Trademarks

**The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.**

DB2*
DB2 Connect
DB2 Universal Database
e-business logo
GDPS*
Geographically Dispersed Parallel Sysplex
HyperSwap
IBM*
IBM eServer
IBM logo*
Parallel Sysplex*
System REXX
System z
Tivoli*
VM/ESA*
WebSphere*
zEnterprise*
z/OS*
z/VM*
zSeries*

* Registered trademarks of IBM Corporation

**The following are trademarks or registered trademarks of other companies.**

Intel is a registered trademark of the Intel Corporation in the United States, other countries or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

* All other products may be trademarks or registered trademarks of their respective companies.

**Notes**:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

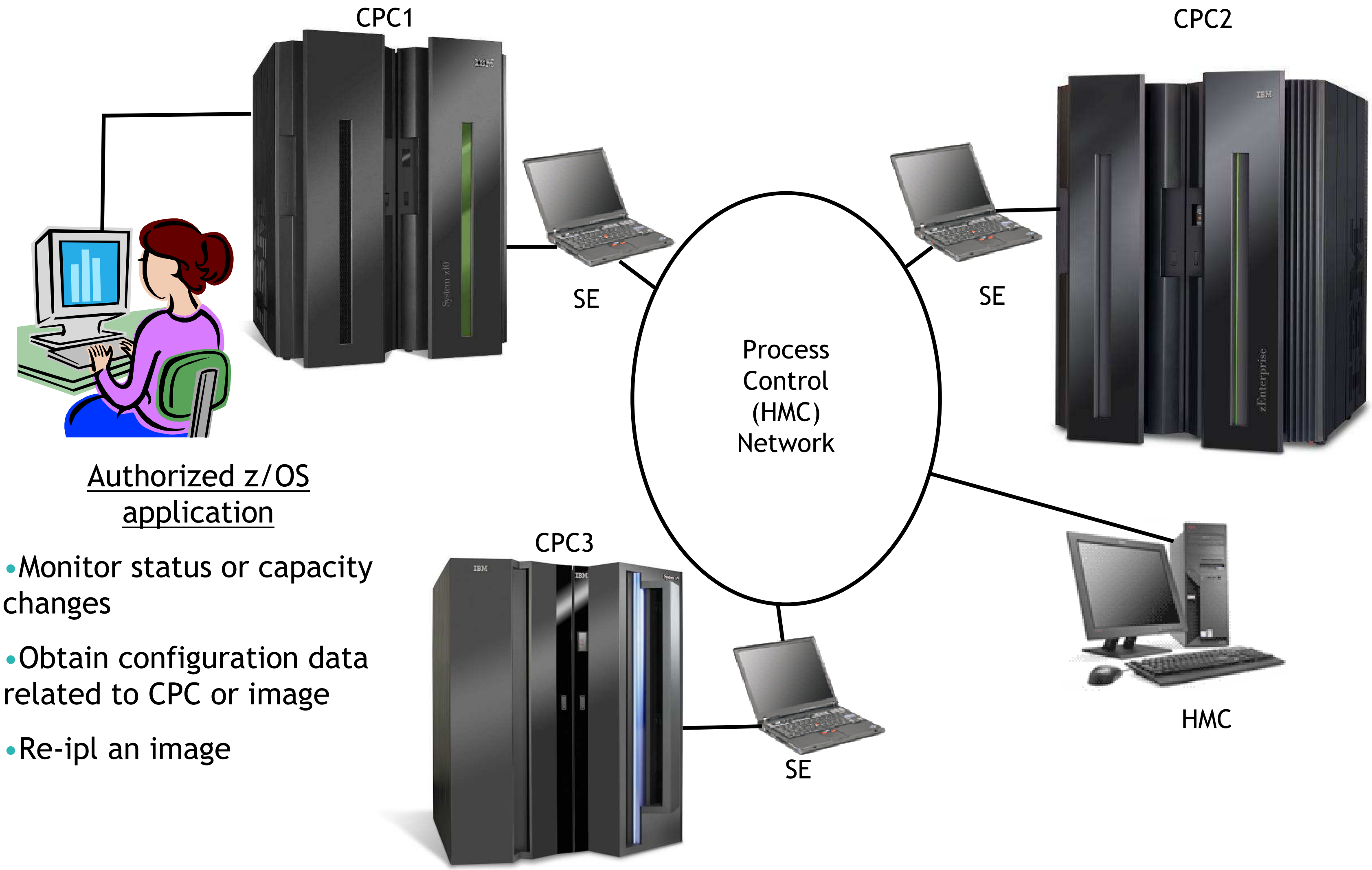
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

# Agenda

- BCPii Intro
  - *What is BCPii?*
  - *Who would use it?*
  - *How do I get it and what machines can be targeted?*
  - *How do I program using it (the BCPii APIs)?*
  - *What do the internals of BCPii look like?*
- BCPii installation and configuration
  - *Configuring the support element*
  - *Configuring the security product*
  - *Configuring the address space*
  - *Configuring optional z/OS UNIX event support*
- BCPii usage
  - *How to start and stop BCPii*
  - *BCPii diagnostics*

# What is BCPii?



- Monitor status or capacity changes
- Obtain configuration data related to CPC or image
- Re-ipl an image

# What is BCPii?

- Base Control Program internal interface
  - *Allows authorized z/OS applications to have HMC-like control over systems in the process control (HMC) network*
    - A set of robust APIs is provided
  - *Complete communication isolation of existing networks (intranet/internet) from the process control (HMC) network*
    - Communication to the support element is completely within base z/OS

# What is BCPii?

- A set of authorized APIs
- A new z/OS address space
- A communications transport between a z/OS application and the local support element
  - *Also between other support elements connected to other CPCs routed by the HMC.*

# What is z/OS BCPii vs. BCPii mentioned in TSA?

- Tivoli System Automation (ProcOps) allows its automation product to use one of 2 transport protocols:
  - *SNMP over an IP network*
  - *BCPii protocol (internal transport)*
- TSA's BCPii implementation is similar but not z/OS BCPii and requires TSA, Netview and Comm Server.
- BCPii transport in TSA is for TSA usage only
- z/OS BCPii can run in ANY address space and has no other product requirements.

# Who would use BCPii?

- z/OS operating system components
- Vendor applications
  - *Control center, systems management applications*
- In-house applications
  - *Installations that have desire to query and possibly manipulate System z resources*
    - In a more automated, programmatic approach
    - From a z/OS environment rather than a hands-on HMC environment
    - Without needing to worry about an intranet/internet/internal network being connected to the process control (HMC) network

# Who would use BCPii?

- Current BCPii exploiters
  - *Capacity Provisioning Manager (CPM)*
    - Used to have the capability of adding or removing temporary capacity based on goals defined in the tool
    - Available in z/OS V1R10 via APAR OA24945 & higher
    - See SHARE session "Introduction to Capacity Provisioning", Wednesday, Aug 4 at 4:30pm
    - See z/OS Hot Topics August 2009 Issue, pages 73-75
  - *Sysplex Failure Manager (SFM)*
    - SFM can more quickly detect that an image has failed rather than relying on their current timeout interval, minimizing "sympathy sickness" by more aggressively removing a system from the plex
    - Minimizes the *"Down error message"* on the console when a missing XCF heartbeat occurs
    - Available in z/OS V1R11

# How do I get BCPii?

- z/OS V1R10
  - *Separate deliverable*
  - *Enabled on z/OS V1R10 via APAR OA25426 (PTF UA47493)*
- z/OS V1R11 and higher
  - *Shipped in base*
  - *Additional functions available*
- After installation of PTF (V1R10) or after V1R11 installation:
  - *BCPii will attempt to auto-start during IPL of z/OS*
    - If the proper configuration has not been made for BCPii, it will graciously terminate.
    - BCPii can be re-started via the START command later once the BCPii has been configured properly

© 2010 IBM Corporation

# What is shipped in each release?

- z/OS V1R10 + PTF
  - Base functions (no HWISET)
- z/OS V1R11
  - HWISET
  - Support for IPL Token / Query PSWs
  - Activation profiles support
  - Minor internal serviceability enhancements
- z/OS V1R12
  - CTRACE enhancements
  - Improved storage utilization and serviceability of BCPii transport code
  - Additional CPC/Image attributes and commands

# What machines can be targeted?

- ***zEnterprise***
- ***z10 plus recommended microcode levels***
  - *Close to full functionality*
    - Driver 79
      - > MCL034 in the N24415 (HMC-SYSTEM) EC stream - Bundle #22
      - > MCL088 in the N24409 (SE-SYSTEM) EC stream - Bundle #22
- ***z9 plus recommended microcode levels***
  - *Some reduced functionality (no IPLTOKEN, reduced attributes, no temporary capacity options)*
- ***Lower than z9***
  - **Significantly reduced functionality (no HWICMD, reduced attributes)**

# How to program using the BCPii APIs?

- Functions performed using BCPii APIs:
  - *Obtain the System z topology of the current interconnected CPCs, Images (LPARs) and Capacity Records*
  - *Query various CPC, image (LPAR) and capacity record information*
  - *Issue commands against both the CPC and image to perform hardware and software-related functions*
  - *Listen for various hardware and software events which may take place on various CPC and images*

# How to program using the BCPii APIs?

- Environment:
  - *Services available in any address space*
    - Program-authorized, and
    - SAF-authorized
  - *C and Assembler programming languages*
  - *z/OS Unix callers can receive event notifications thru z/OS Unix-only services utilizing Common Event Adapter (CEA)*

# How to program using the BCPii APIs?

- Services available
  - *HWILIST (BCPii List)*
  - *HWICONN (BCPii Connect)*
  - *HWIDISC (BCPii Disconnect)*
  - *HWIQUERY (BCPii Query)*
  - *HWISET (BCPii Set) - introduced in V1R11*
  - *HWICMD (BCPii Command)*
  - *HWIEVENT (BCPii Event (for non-z/OS Unix callers))*
  - *HwiBeginEventDelivery, HwiEndEventDelivery, HwiManageEvents, HwiGetEvent (for z/OS Unix callers)*

# How to program using the BCPii APIs?

- **HWILIST** - Retrieve HMC and BCPii configuration-related information
  - *List CPCS*
    - List the CPCs interconnected with the local CPC
  - *List Images*
    - List the images (LPARs) contained on an individual CPC
  - *List Capacity Records*
    - List the capacity records contained on an individual CPC
  - *List Events*
    - List the events already registered on a particular BCPii connection
  - *List Local CPC, List Local Image (available in V1R11)*
    - Obtain the name of the CPC name or image (LPAR) name that the BCPii application is currently running on.
  - *List Reset Activation Profiles, List Image A.P. and List Load A.P. (available in V1R11 via APAR OA29638)*
    - List the currently defined activation profiles contained on a individual CPC

Information retrieved on the service is used on subsequent HWICONN calls

# How to program using the BCPii APIs?

- **HWICONN** - Establish a logical connection between the application and a:
  - *Central processor complex (CPC),*
  - *CPC image (LPAR) on a particular CPC,*
  - *Capacity record on particular CPC*
  - *Activation Profiles (available in V1R11 via APAR OA29638)*
- Input:
  - *Connection type (above 3 types)*
  - *Connection name (CPC example: net1.cpc01)*
  - *Previous ConnectToken (if type is image or caprec)*
- Output:
  - *ConnectToken used on subsequent BCPii calls.*

# How to program using the BCPii APIs?

- **HWIDISC** – Release a logical connection no longer needed
- Input:
  - *ConnectToken*

# How to program using the BCPii APIs?

- **HWIQUERY** - Retrieve information about objects managed by the hardware management console (HMC)/support element related to:
  - *Central processor complexes (CPCs),*
  - *CPC images (LPARs) on a particular CPC,*
  - *Capacity records on particular CPC*
  - *Activation Profiles (Reset, Image, or Load) (available in V1R11 via APAR OA29638)*

  *Input:*
  - *ConnectToken (associated with one of the above)*
  - *List of attributes requested, data areas to store the return values)*
- Output
  - *Data returned*

# How to program using the BCPii APIs?

- **Examples of information you can query**
  - *CPC information*
    - General information
      - **Name, serial, machine type, id, networking info**
    - Status information
      - **Operating status and other status values**
    - Capacity information
      - **Various CBU info, Capacity on Demand info, Processor configuration, including IFA, IFL, ICF, IIP**
    - Power savings information *(available on zEnterprise hardware only (mid-November) with APAR OA30558 on V1R10, V1R11 and V1R12)*
      - **Is power savings available?, current power savings mode, supported power saving modes available**
  - *Image information*
    - General information
      - **Name, OS info**
    - Capacity information
      - **Defined capacity, Processor weights**

© 2010 IBM Corporation

# How to program using the BCPii APIs?

- Examples of information you can query (continued):
  - *Capacity record information*
    - General information
      - **Name, Activation and expiration dates, activation days**
    - Status information
      - **Record status**
    - Capacity information
      - **The entire Capacity record**

## How to program using the BCPii APIs?

- **HWISET** *(available in V1R11)* – Change or set data for objects managed by the hardware management console (HMC)/support element related to:
  - *Central processor complexes (CPCs),*
  - *CPC images (LPARs) on a particular CPC,*
  - *Activation Profiles* *(available in V1R11 via APAR OA29638)*
- Input:
  - *ConnectToken (associated with one of the above)*
  - *Attribute (object) to modify, the modified value*
- Output
  - *Return code*

© 2010 IBM Corporation

# How to program using the BCPii APIs?

- **Examples of information you can set**
  - *CPC information*
    - Acceptable status values
    - Next Reset activation profile name
    - Processor Running Time
  - *Image information*
    - Various processor weights
  - *Activation Profile Information (available in V1R11 via APAR OA29638)*
    - Any activation profile value

# How to program using the BCPii APIs?

- **HWICMD** – Direct hardware/software commands to CPCs and images
- Input:
  - *ConnectToken (associated with a CPC or image)*
  - *Command parameter structure (based on the type of command issued)*
- Output
  - *Synchronous return code*
  - *Asynchronous command completion event delivered to previously-registered event user when command finishes.*

# How to program using the BCPii APIs?

- Examples of commands that can be issued:
  - *CPC commands*
    - Activate, Deactivate an entire CPC
    - CBU request
      - **Activate or Undo**
    - On/Off Capacity on Demand request
      - **Activate or Undo**
    - Switch Power Savings Mode *(available on zEnterprise hardware only (mid-November) with APAR OA30558 on V1R10, V1R11 and  V1R12)*

  Image commands
    - SysReset, SysReset with IPL Token *(V1R11)*
    - Load
    - Start, Stop all CPs
    - Add or remove temporary capacity
    - Issue operating system command

## How to program using the BCPii APIs?

- **HWIEVENT (non-z/OS Unix callers)-** Register/Un-register an application and its connection to be notified for hardware and software events occurring on the connected CPC or image.
- Input:
  - *ConnectToken (associated with a CPC or image)*
  - *Event action (Add or Delete)*
  - *Events for which an application wants to be notified*
  - *ENF exit to receive control when event arrives*
- BCPii registers the user with ENF for this event(s) such that the ENF exit is driven only when the CPC and/or image name of the connector matches.

# How to program using the BCPii APIs?

- Examples of events that can be listened to:
  - *Command completions*
  - *Status changes*
  - *Capacity changes*
  - *Disabled waits*
  - *Power mode changes* (available on zEnterprise hardware only (mid-November) with APAR OA30558 on V1R10, V1R11 and V1R12)
  - *BCPii status changes*

# How to program using the BCPii APIs?

- **HwiBeginEventDelivery (z/OS Unix callers)** –begin delivery of event notifications.
- Input:
  - *ConnectToken (associated with a CPC or image)*
- Output:
  - *DeliveryToken*
    - To be used on HwiManageEvents service

# How to program using the BCPii APIs?

- **HwiEndEventDelivery (z/OS Unix callers)** –End delivery of event notifications.
- Input:
  – *DeliveryToken*

# How to program using the BCPii APIs?

- **HwiManageEvents (z/OS Unix callers)** –Registers / un-registers for a list of hardware/software events.
- Input:
  - *ConnectToken*
  - *DeliveryToken*
  - *Event action (Add or Delete)*
  - *Events to be registered/unregistered*

# How to program using the BCPii APIs?

- **HwiGetEvent (z/OS Unix callers)** –Retrieve outstanding event notifications.
- Input:
  - *DeliveryToken*
  - *Buffer*
    - Where the ENF68 event data is to be returned
  - *Timeout*
    - How much time to wait for an event to occur
- Output:
  - *ENF68 Event Data in supplied buffer*

# How to program using the BCPii APIs?

- Simple Example:
  - *My company monitor application wants to keep track of the list of CPCs and how many CBU test activations are remaining on each CPC, the list of images (LPARs) on those CPCs, what operating systems are running on all those images, as well as the OS levels and if z/OS, what sysplex are they a part of.  Finally, the "Current Defined Capacity" for each image is desired.*

# How to program using the BCPii APIs?

- Application contains calls like this:
  - ***HWILIST*** *(ListCPCs)*
  - *For each CPC name returned above:*
    - **HWICONN** (CPC name (input), CPCConnectToken(output))
    - **HWIQUERY** (CPCConnectToken (input), QueryParms (HWI_CBUTESTAR))
    - **HWILIST** (CPCConnectToken(input), ListImages)
    - **For each image returned above:**
      - **HWICONN** (CPCConnectToken(input), Image name (input), ImageConnectToken(output))
      - **HWIQUERY** (ImageConnectToken(input), QueryParms(HWI_OSNAME,HWI_OSTYPE,HWI_OSLEVEL,HWI_SYSPLEX, HWI_DEFCAP)
      - **HWIDISC**(ImageConnectToken)
    - **HWIDISC** (CPCConnectToken)

# How to program using the BCPii APIs?

```
rc = -1;
numofCPCs = -1;
listtype = HWI_LIST_CPCS;
memset(List_of_CPCs, 0x00, sizeof(List_of_CPCs));
answerarealen = sizeof(List_of_CPCs);
answerarea_ptr = &answerarea[0];

hwilist(&rc,CPCoutconnecttoken,listtype,&numofCPCs,
        &answerarea_ptr,answerarealen,&diagarea);

if ( rc == 0 )
{
 printf("HWILIST for CPC: RC = %x\n",rc);
 printf("NumOfDataItem:   %d\n",numofCPCs);

 CPC_AnswerArea_into_Array(answerarea, List_of_CPCs, numofCPCs);
```

# How to program using the BCPii APIs?

```
while( i < numofCPCs )
{
 printf("CPC %d: %s\n",i+1,&List_of_CPCs[i].element);
 /* CPC **************************************************/
 /* HWICONN the CPC */
 rc = -1;
 memset(CPCinconnecttoken, 0x00, sizeof(CPCinconnecttoken));
 strcpy(CPC_target,List_of_CPCs[i].element);
 CPCconnecttypevalue = &CPC_target[0];
 CPCconnecttype = 1;

 hwiconn(&rc,CPCinconnecttoken,&CPCoutconnecttoken,CPCconnecttype,
         &CPCconnecttypevalue,&diagarea);

 if ( rc == 0 )
 {
  printf("HWICONN on %s: RC = %x\n",&List_of_CPCs[i],rc);
```

## How to program using the BCPii APIs?

```
/* HWIQUERY */
/* Calling HWIQUERY, using the returned output connecttoken from
HWICONN, query for HWI_MMODEL */
printf("HWIQUERY for HWI_MMODEL\n");
rc = -1;
numofattributes = 1;
Queryparm[0].AttributeIdentifier = HWI_MMODEL;
Queryparm[0].AttributeValue_Ptr  = &HWI_MMODEL_value[0];
Queryparm[0].AttributeValueLen   = sizeof(HWI_MMODEL_value);
Queryparm[0].AttributeValueLenReturned = -1;
query_ptr = (char *)&queryparm[0];

hwiquery(&rc,CPCoutconnecttoken,(void **)&query_ptr
         ,numofattributes,&diagarea);


if ( rc == 0 )

{printf("HWIQUERY on %s: RC = %x\n",&List_of_CPCs[i],rc);
```

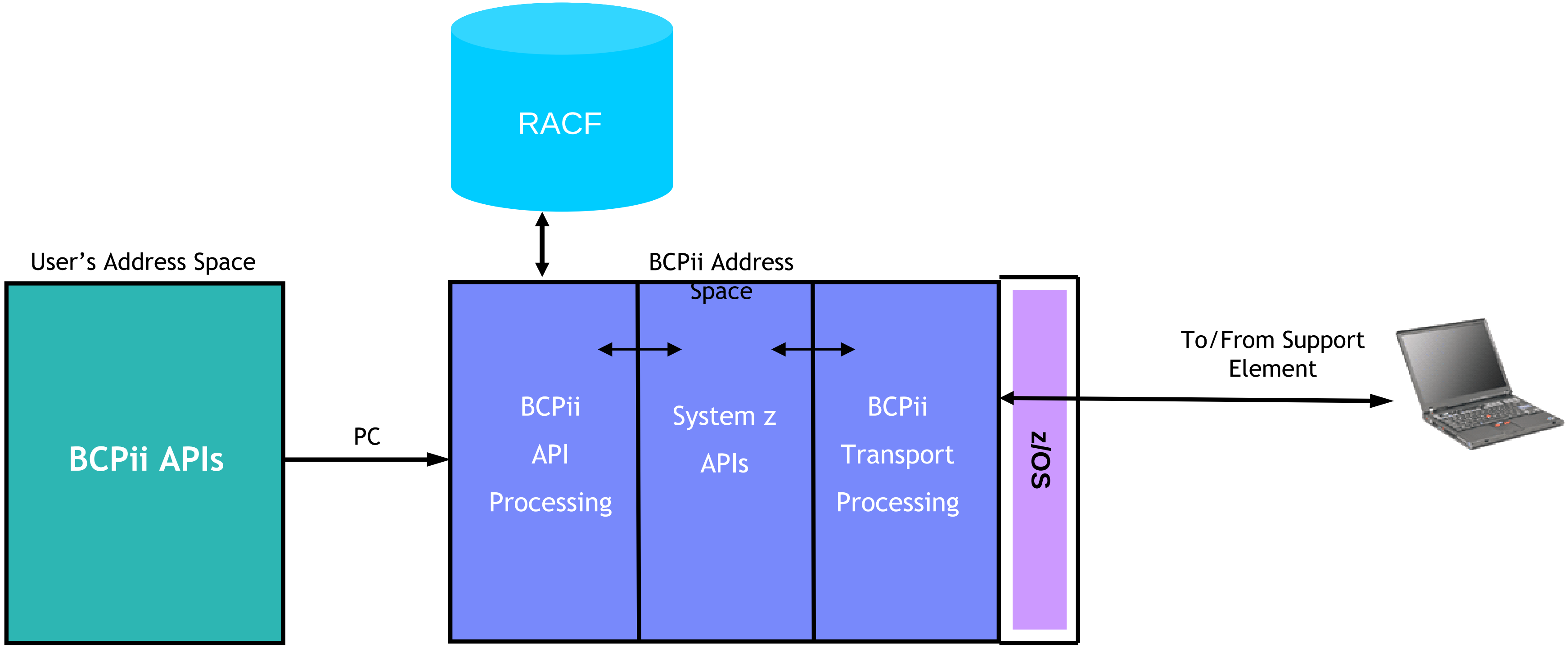
# What do the internals of BCPii look like?

- **The new BCPii address space**
  - *The bridge between a z/OS application and the support element*
    - Manages all application connections
    - Builds and receives all internal communication requests to the SE
    - Provides an infrastructure for storage required by callers and for the transport communicating with the SE
    - Provides diagnostic capabilities to help with BCPii problem determination
    - Provides security authentication of requests

# What do the internals of BCPii look like?

- The BCPii address space:
  - *Starts automatically during IPL*
  - *Requires some initial setup before becoming active*
  - *Can be stopped if necessary*
  - *Can be restarted via the HWISTART proc*

# What do the internals of BCPii look like?



# IPL Messages Prior to BCPii Setup in z/OS 1.11 or z/OS 1.12

```
IRR813I NO PROFILE WAS FOUND IN THE STARTED CLASS FOR 497
        HWIBCPII WITH JOBNAME HWIBCPII. RACF WILL USE ICHRIN03.
IEF196I         2 //HWIBCPII EXEC IEESYSAS,PROG=HWIAMIN2
IEF196I  STMT NO. MESSAGE
IEF196I           XX* THE IEESYSAS PROCEDURE IS SPECIFIED IN THE
IEF196I                   00150000
IEF196I           XX* PARAMETER LIST TO IEEMB881 BY MVS COMPONENTS
IEF196I                   00200000
IEF196I         2 IEFC001I PROCEDURE IEESYSAS WAS EXPANDED USING SYSTEM
IEE252I MEMBER CTIHWI00 FOUND IN SYS1.PARMLIB
HWI007I BCPII IS ATTEMPTING COMMUNICATION WITH THE LOCAL CENTRAL 965
PROCESSOR COMPLEX (CPC).
HWI014I THE SNMP COMMUNITY NAME FOR BCPII IN THE SECURITY 049
PRODUCT FOR THE LOCAL CPC IS EITHER NOT DEFINED OR IS INCORRECT.
IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=HWIBCPII, ASID=001A.
HWIBCPII HWIBCPII IEFPROC   00:00:00      2,117     00
IEF404I IEESYSAS - ENDED - TIME=13.31.03
IEF196I IEF142I IEESYSAS HWIBCPII - STEP WAS EXECUTED - COND CODE 0000
IEF196I IEF376I  JOB/HWIBCPII/STOP  2009239.1331 CPU    0MIN 00.02SEC
IEF196I SRB    0MIN 00.00SEC
IEF352I ADDRESS SPACE UNAVAILABLE
IEF196I IEF352I ADDRESS SPACE UNAVAILABLE
IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=*UNAVAIL, ASID=001A.
HWI006I BCPII ADDRESS SPACE HAS ENDED.
IXC104I SYSTEM STATUS DETECTION PARTITIONING PROTOCOL ELIGIBILITY: 399
  SYSTEM CANNOT TARGET OTHER SYSTEMS.
    REASON: BCPII SERVICES NOT AVAILABLE
  SYSTEM IS NOT ELIGIBLE TO BE TARGETED BY OTHER SYSTEMS.
    REASON: BCPII SERVICES NOT AVAILABLE
IXC107E SYSTEM STATUS DETECTION PARTITIONING PROTOCOL CONFIGURATION IS NOT COMPLETE
```

© 2010 IBM Corporation

# BCPii Configuration

- **BCPii Installation Steps**
  - *Configure the local SE to support BCPii*
    - HMC/SE Administrator
  - *Authorize an application to use BCPii*
    - Security Administrator
  - *Configure the BCPii address space*
    - z/OS System Administrator
  - *Set up the event notification mechanism for z/OS UNIX callers (if required)*
    - z/OS System Administrator and Security Administrator

# BCPii Configuration – Configuring the local SE

- Make sure microcode is at the correct level
- Enable BCPii communications on the SE
  - *Turn on "Cross Partition Authority"*
- Define the BCPii community name on the SE
  - *Turn on "Enable SNMP APIs"*
  - *Add a BCPii community name for the BCPii port*
- Repeat these same steps above for any additional CPCs communication is desired.

# BCPii Configuration – Configuring the local SE

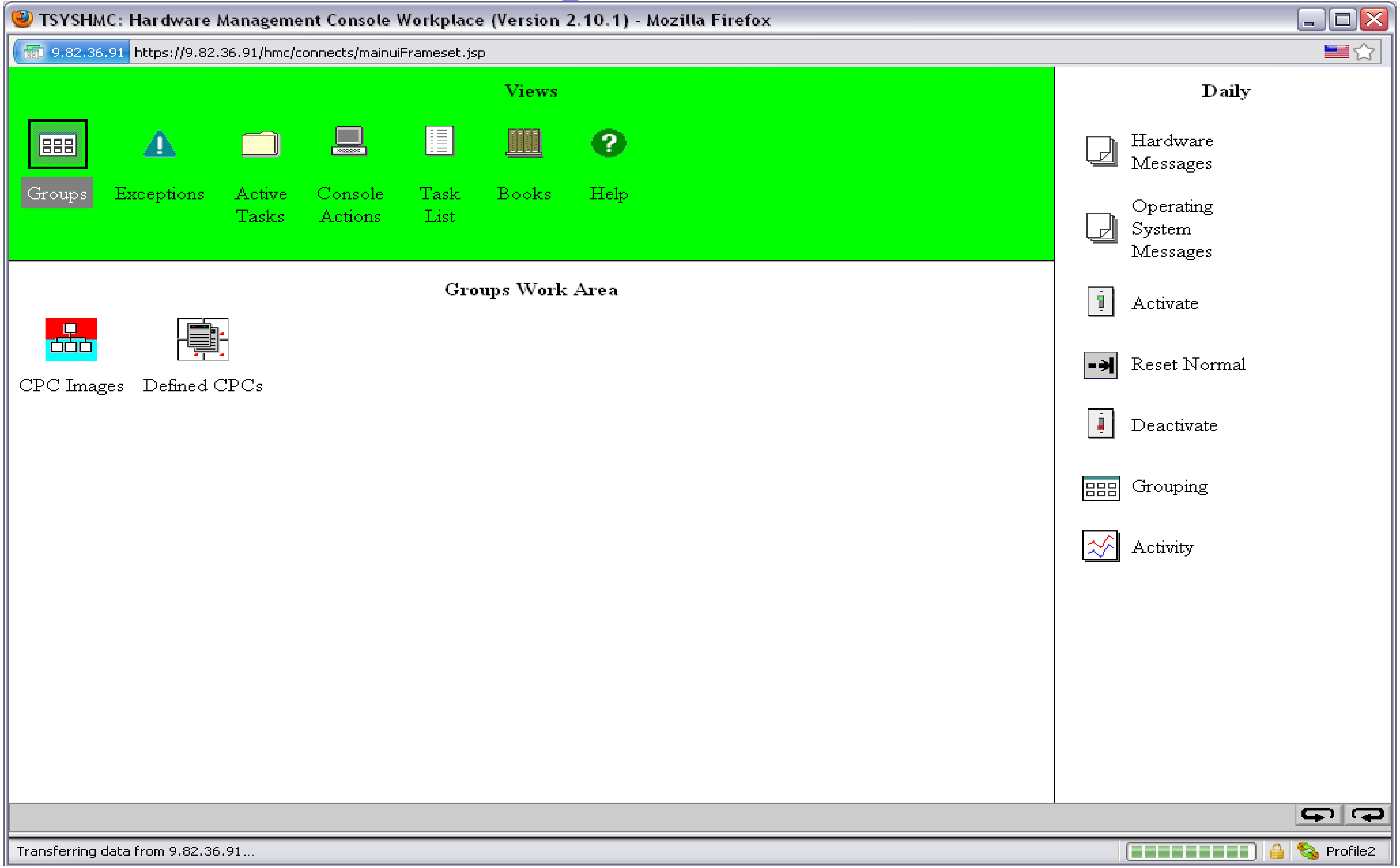
- Make sure microcode is at a minimum level
  - *z9*
    - MCL240 in the G40965 (SE-SYSTEM) EC stream in Driver 67
    - MCL046 in the G40969 (HMC-SYSTEM) EC stream in Driver 67
  - *z10*
    - MCL088 in the N24409 (SE-SYSTEM) EC stream in Driver 79
    - MCL034 in the N24415 (HMC-SYSTEM) EC stream in Driver 79
  - *zEnterprise*
    - MCL044 in the N29802 (SE-SYSTEM) EC stream in Driver 86, z196 GA level driver.

© 2010 IBM Corporation

# BCPii Configuration – Configuring the local SE

- Enable BCPii communications on the SE
  - *Turn on "Cross Partition Authority"*
    - Open a reset or image activation profile
    - Select the security page
    - Click the "Cross Partition Authority" check box.

# Access Hardware Management Console

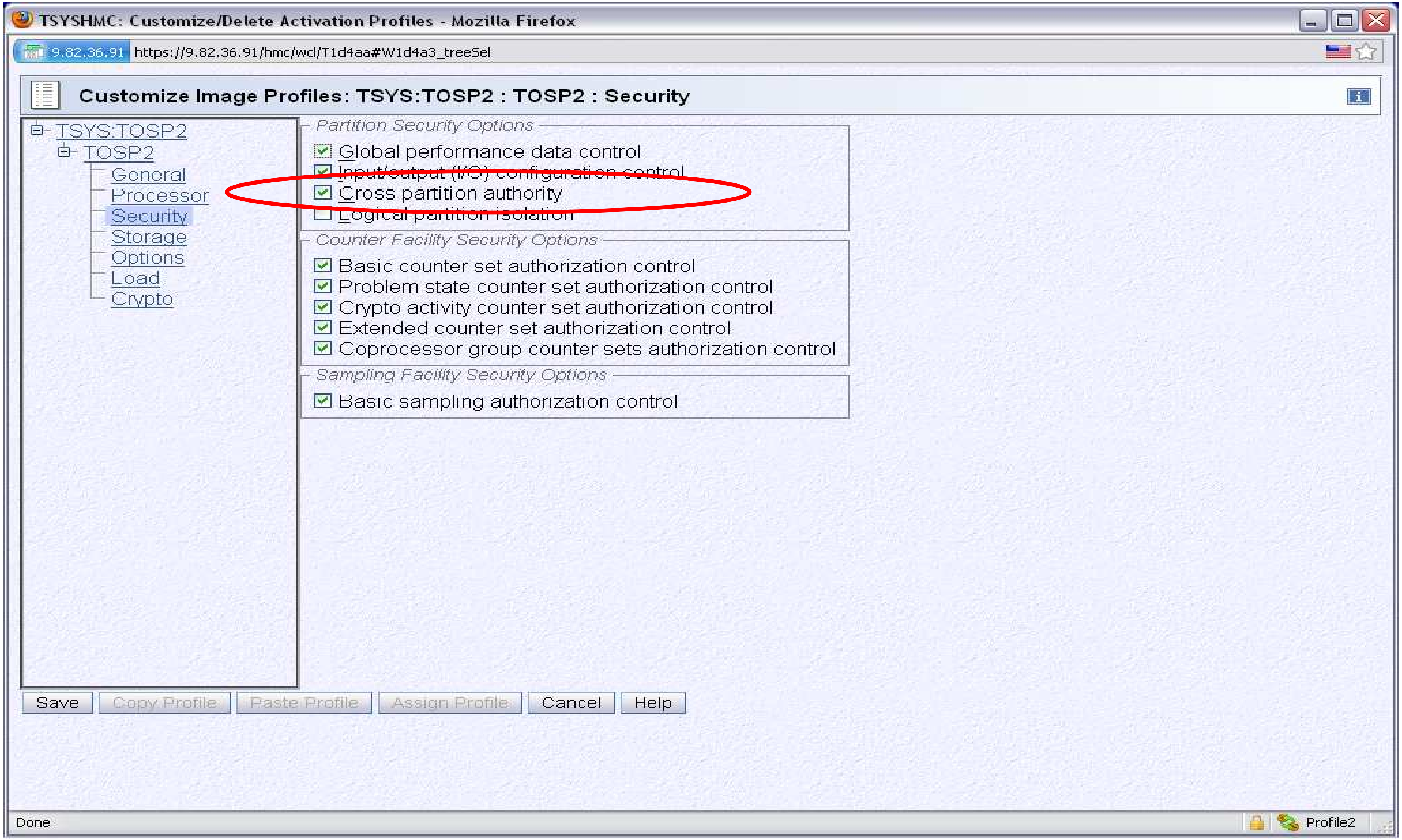


TSYSHMC: Hardware Management Console Workplace (Version 2.10.1) - Mozilla Firefox

9.82.36.91 https://9.82.36.91/hmc/connects/mainuiFrameset.jsp

Transferring data from 9.82.36.91...

# Cross Partition Authority



TSYSHMC: Customize/Delete Activation Profiles - Mozilla Firefox

9.82.36.91 https://9.82.36.91/hmc/wcl/T1d4aa#W1d4a3_treeSel

Customize Image Profiles: TSYS:TOSP2 : TOSP2 : Security

*Partition Security Options*

☑ Global performance data control
☑ Input/output (I/O) configuration control
☑ Cross partition authority
□ Logical partition isolation

*Counter Facility Security Options*

☑ Basic counter set authorization control
☑ Problem state counter set authorization control
☑ Crypto activity counter set authorization control
☑ Extended counter set authorization control
☑ Coprocessor group counter sets authorization control

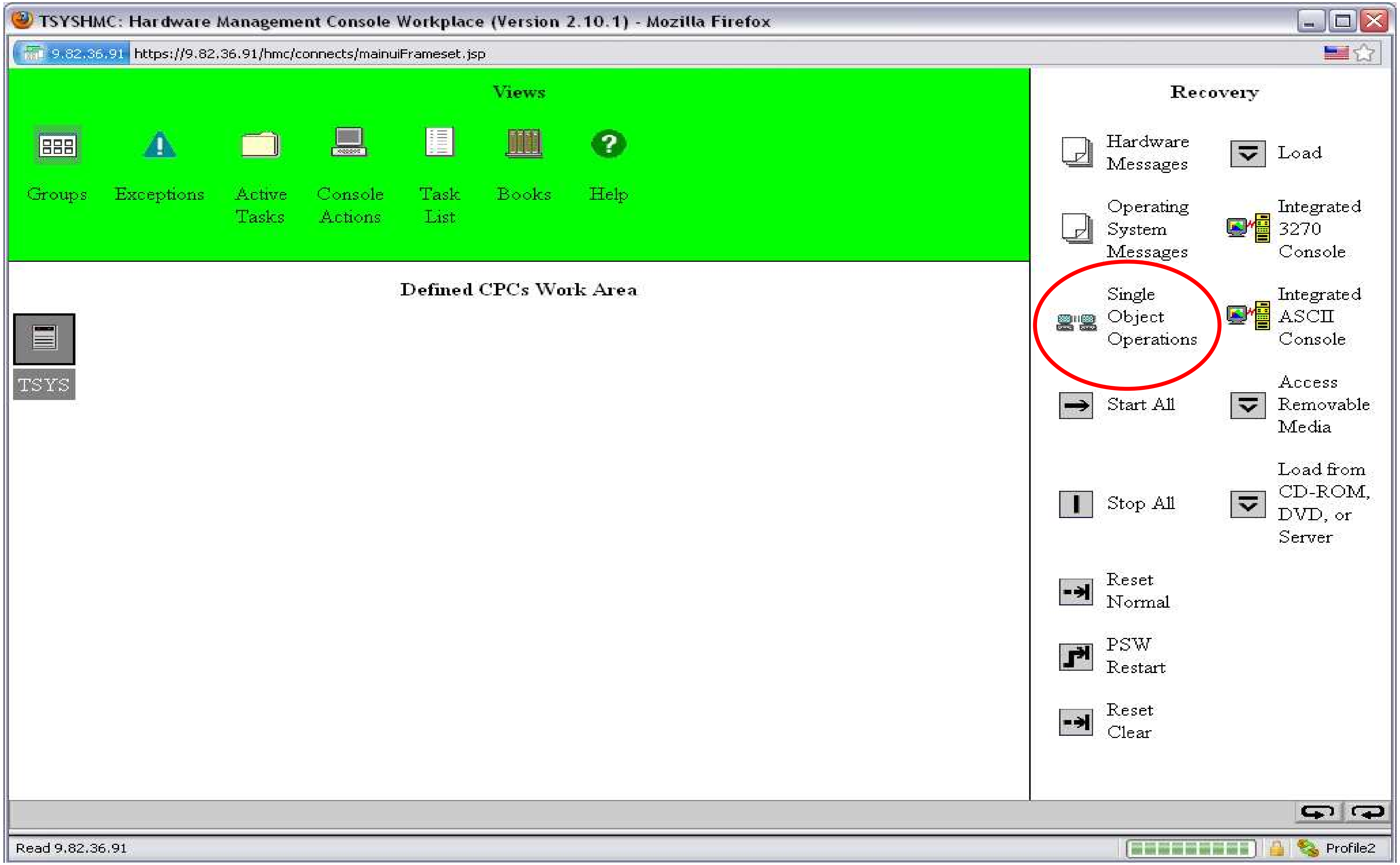
*Sampling Facility Security Options*

☑ Basic sampling authorization control

# BCPii Configuration – Configuring the local SE

- Define the BCPii community name on the SE
  - *Click on "Enable SNMP APIs" checkbox*
    - Optionally click on *"Allow Capacity change API requests"*
  - *Add a BCPii community name for the BCPii port*
    - Specify an UPPERCASE-only name
    - Specify address as 127.0.0.1
    - Specify network mask as 255.255.255.255
    - Specify *Write* access

# Access z Support Element



TSYSHMC: Hardware Management Console Workplace (Version 2.10.1) - Mozilla Firefox

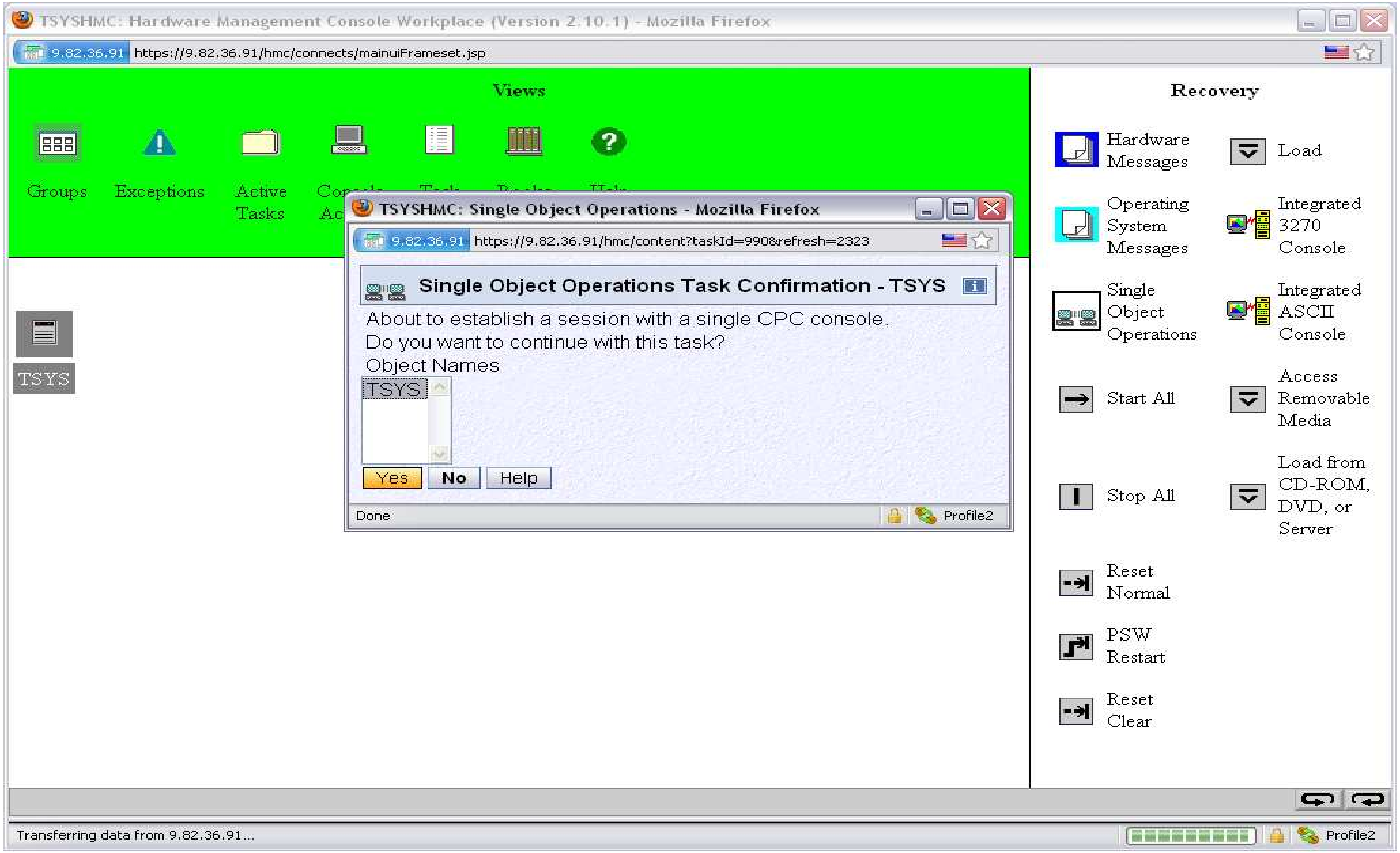
9.82.36.91 https://9.82.36.91/hmc/connects/mainuiFrameset.jsp

Defined CPCs Work Area

Recovery

© 2010 IBM Corporation

# HMC Single Object Operation



TSYSHMC: Hardware Management Console Workplace (Version 2.10.1) - Mozilla Firefox

9.82.36.91 https://9.82.36.91/hmc/connects/mainuiFrameset.jsp

TSYSHMC: Single Object Operations - Mozilla Firefox

9.82.36.91 https://9.82.36.91/hmc/content?taskId=990&refresh=2323

Single Object Operations Task Confirmation - TSYS

About to establish a session with a single CPC console.
Do you want to continue with this task?

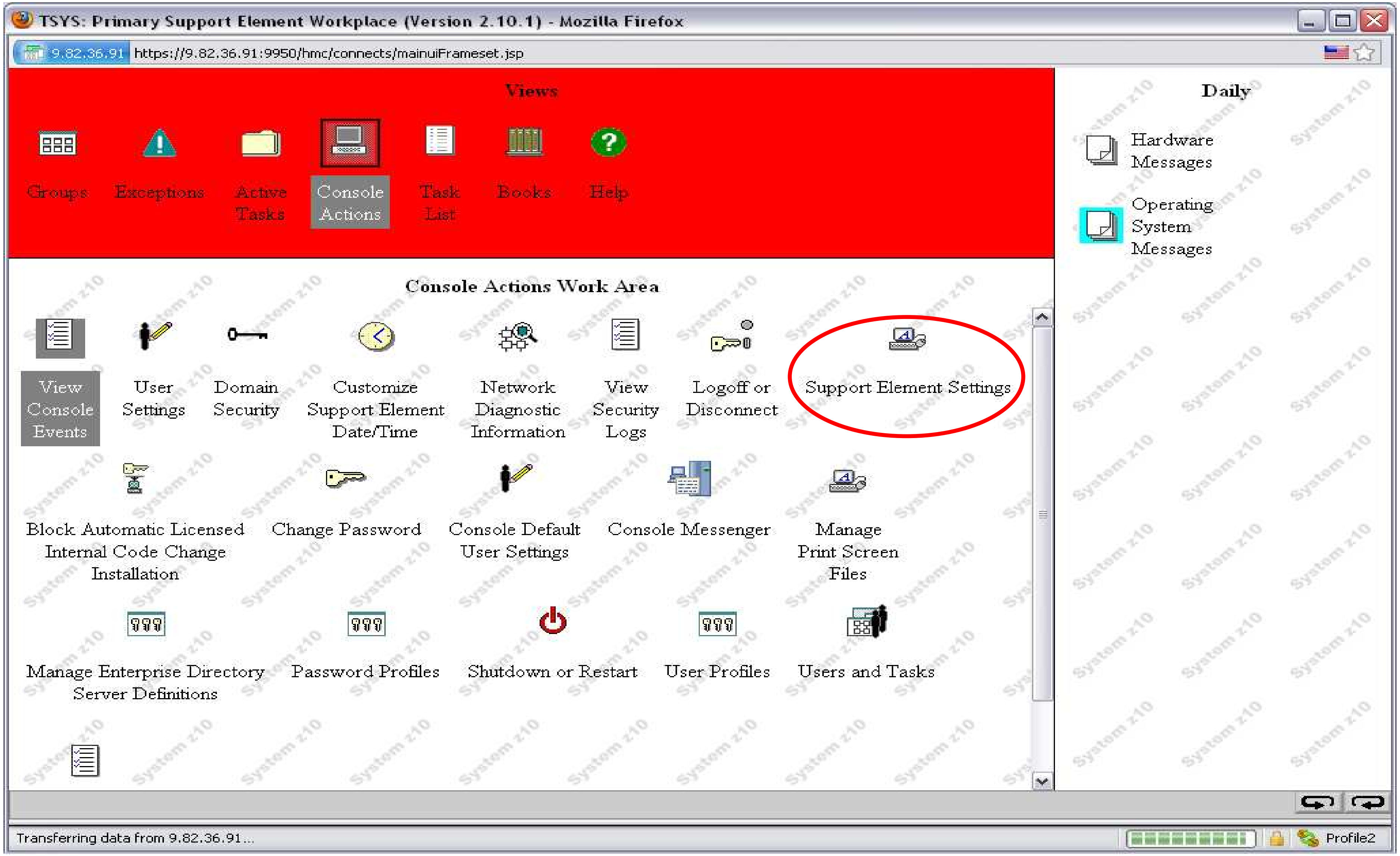
Object Names

TSYS

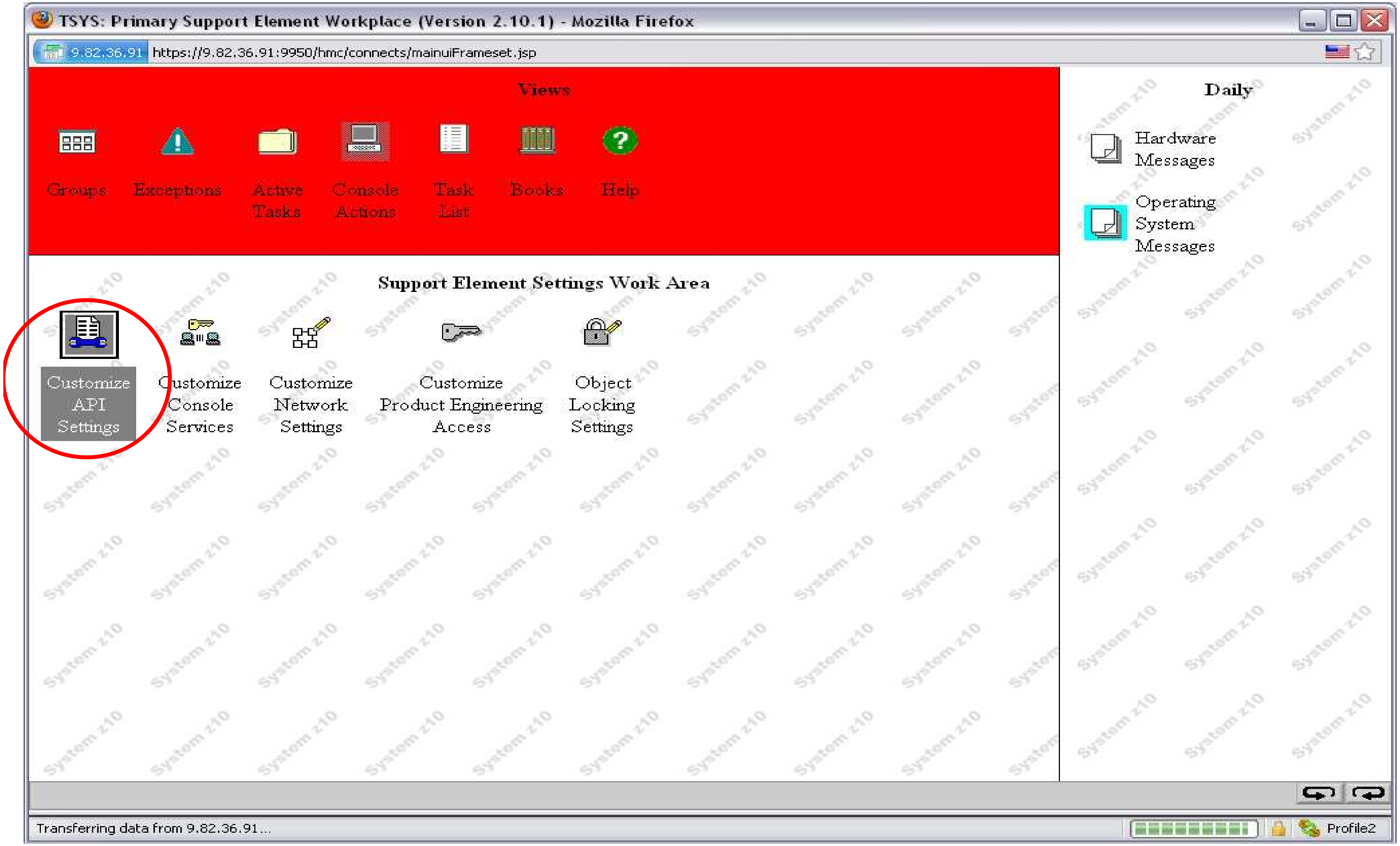
Done

Transferring data from 9.82.36.91...

# Support Element Settings


TSYS: Primary Support Element Workplace (Version 2.10.1) - Mozilla Firefox
9.82.36.91
https://9.82.36.91:9950/hmc/connects/mainuiFrameset.jsp
Views
Groups
Exceptions
Active Tasks
Console Actions
Task List
Books
Help
Daily
Hardware Messages
Operating System Messages
Console Actions Work Area
View Console Events
User Settings
Domain Security
Customize Support Element Date/Time
Network Diagnostic Information
View Security Logs
Logoff or Disconnect
Support Element Settings
Block Automatic Licensed Internal Code Change Installation
Change Password
Console Default User Settings
Console Messenger
Manage Print Screen Files
Manage Enterprise Directory Server Definitions
Password Profiles
Shutdown or Restart
User Profiles
Users and Tasks
Transferring data from 9.82.36.91...
Profile2

# Customize API Settings



TSYS: Primary Support Element Workplace (Version 2.10.1) - Mozilla Firefox

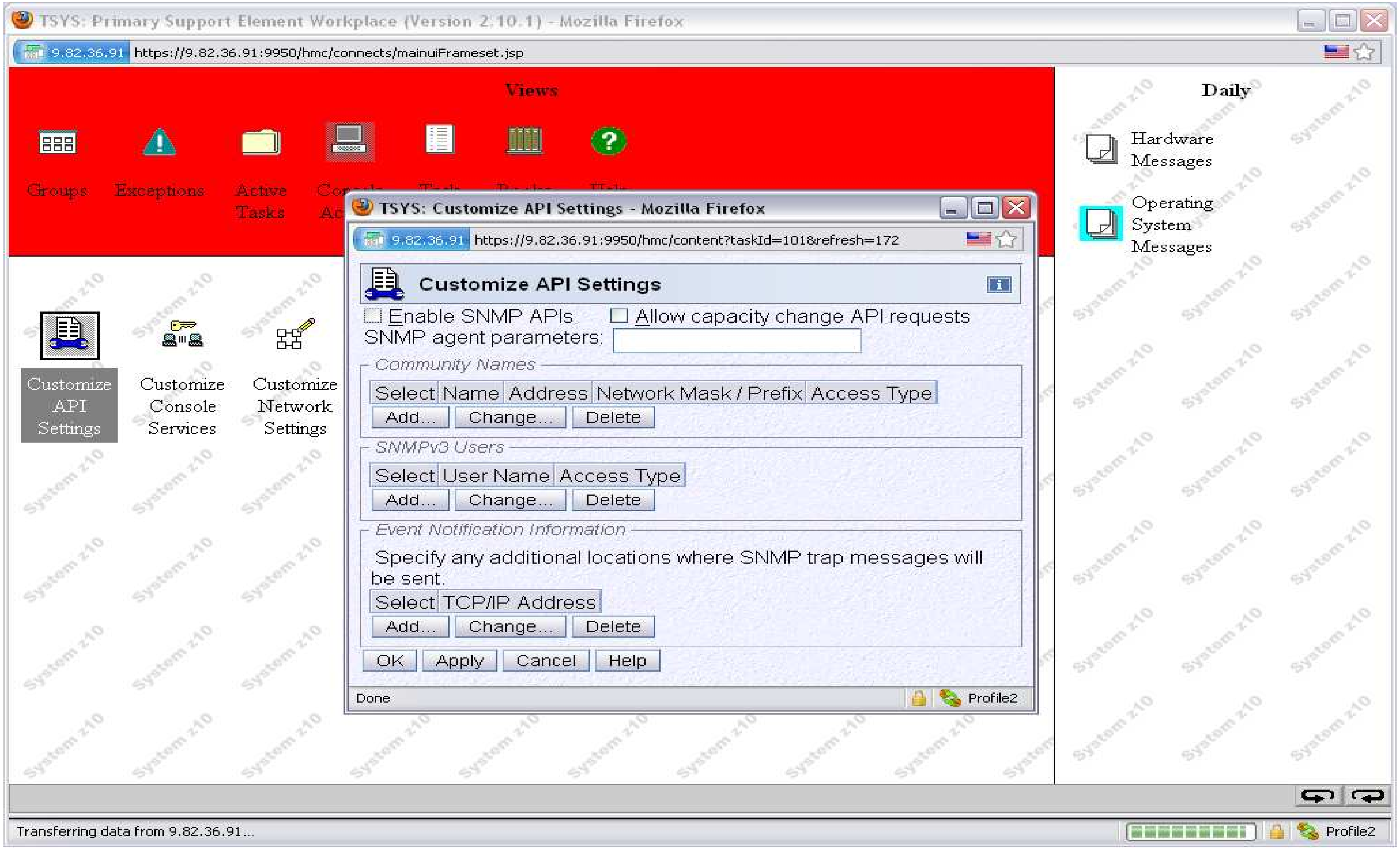
9.82.36.91 https://9.82.36.91:9950/hmc/connects/mainuiFrameset.jsp

Support Element Settings Work Area

Customize API Settings
Customize Console Services
Customize Network Settings
Customize Product Engineering Access
Object Locking Settings

Transferring data from 9.82.36.91...

# Community Name



TSYS: Primary Support Element Workplace (Version 2.10.1) - Mozilla Firefox
9.82.36.91 https://9.82.36.91:9950/hmc/connects/mainuiFrameset.jsp
Views
Groups
Exceptions
Active Tasks
Console Actions
Tasks
Books
Help
Customize API Settings
Customize Console Services
Customize Network Settings
Daily
Hardware Messages
Operating System Messages
Transferring data from 9.82.36.91...
Profile2

TSYS: Customize API Settings - Mozilla Firefox

9.82.36.91 https://9.82.36.91:9950/hmc/content?taskId=101&refresh=172

### Customize API Settings

☐ Enable SNMP APIs ☐ Allow capacity change API requests

SNMP agent parameters:

Community Names

| Select | Name | Address | Network Mask / Prefix | Access Type |
|---|---|---|---|---|

Add... Change... Delete

SNMPv3 Users

| Select | User Name | Access Type |
|---|---|---|

Add... Change... Delete

Event Notification Information

Specify any additional locations where SNMP trap messages will be sent.

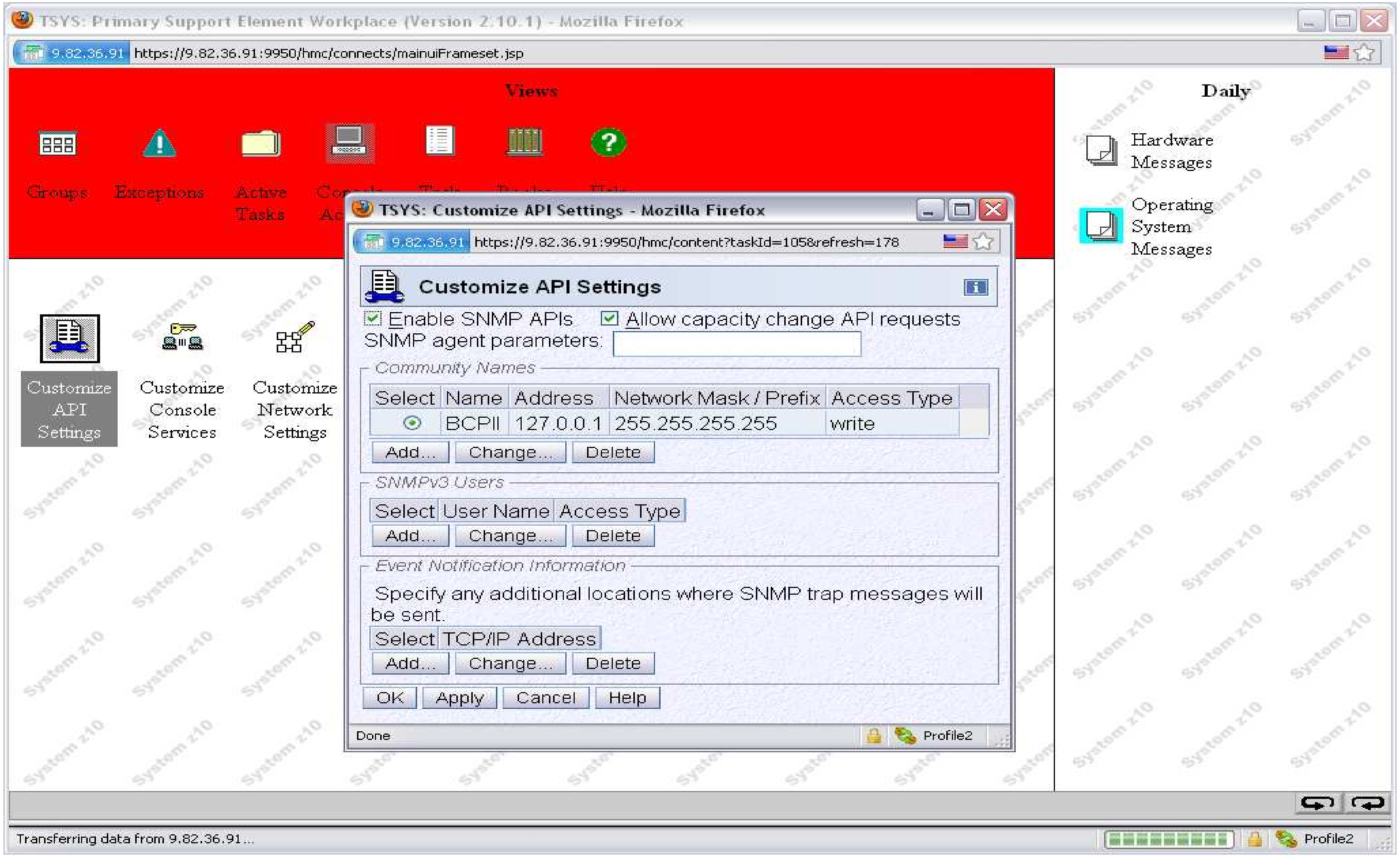
| Select | TCP/IP Address |
|---|---|

Add... Change... Delete

OK Apply Cancel Help

Done Profile2

# Community Name Information



Views
Groups
Exceptions
Active Tasks
Customize API Settings
Customize Console Services
Customize Network Settings
Daily
Hardware Messages
Operating System Messages
Transferring data from 9.82.36.91...
Profile2
Done

TSYS: Primary Support Element Workplace (Version 2.10.1) - Mozilla Firefox

9.82.36.91 https://9.82.36.91:9950/hmc/connects/mainuiFrameset.jsp

TSYS: Customize API Settings - Mozilla Firefox

9.82.36.91 https://9.82.36.91:9950/hmc/content?taskId=105&refresh=178

## Customize API Settings

☑ Enable SNMP APIs ☑ Allow capacity change API requests

SNMP agent parameters:

*Community Names*

| Select | Name | Address | Network Mask / Prefix | Access Type |
|---|---|---|---|---|
| ● | BCPII | 127.0.0.1 | 255.255.255.255 | write |

Add... Change... Delete

*SNMPv3 Users*

| Select | User Name | Access Type |
|---|---|---|

Add... Change... Delete

*Event Notification Information*

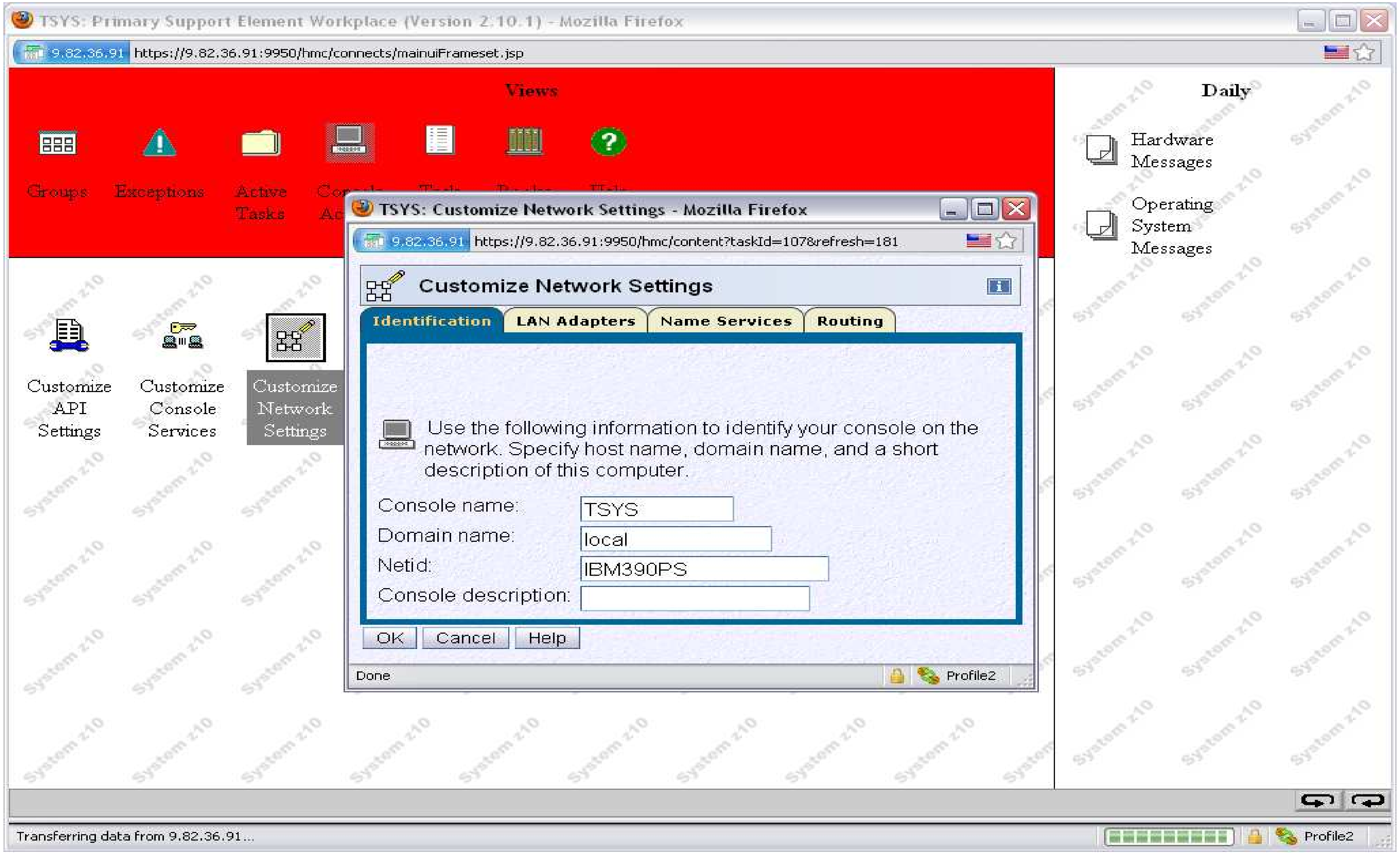
Specify any additional locations where SNMP trap messages will be sent.

| Select | TCP/IP Address |
|---|---|

Add... Change... Delete

OK Apply Cancel Help

# How to find the CPC SNA Name



TSYS: Primary Support Element Workplace (Version 2.10.1) - Mozilla Firefox
9.82.36.91 https://9.82.36.91:9950/hmc/connects/mainuiFrameset.jsp
Views
Groups
Exceptions
Active Tasks
Daily
Hardware Messages
Operating System Messages
Customize API Settings
Customize Console Services
Customize Network Settings
TSYS: Customize Network Settings - Mozilla Firefox
9.82.36.91 https://9.82.36.91:9950/hmc/content?taskId=107&refresh=181
Identification
LAN Adapters
Name Services
Routing
OK
Cancel
Help
Done
Profile2
Transferring data from 9.82.36.91...

Customize Network Settings

Use the following information to identify your console on the network. Specify host name, domain name, and a short description of this computer.

Console name: TSYS

Domain name: local

Netid: IBM390PS

Console description:

# BCPii Configuration – Setting up the support element

- After all these steps, REBOOT the SE.
- Repeat these steps on all the other SEs associated with the CPCs that BCPii access is desired

# BCPii Configuration – Authorizing an application

- Authorize the application to use BCPii
  - *Give program authority to application issuing BCPii APIs*
  - *Give general SAF authority*
  - *Give SAF authority to a specific resource*

# BCPii Configuration – Program Authorization

- Give program authority to application issuing BCPii APIs
  - *Put the module into an APF-authorized library as specified in your PROGxx parmlib member*
    - 
      ```
      APF ADD
          DSNAME(MYBCPII.PROG.LIB)
          VOLUME(ABCDEF)
      ```
  - *Issue SET PROG=xx console command*
  - *Program is NOT required to run in supervisor state or in PKM 0-7*

# BCPii Configuration – Giving general authority to app

- Give general SAF authority
  - Give application at least READ access to HWI.APPLNAME.HWISERV profile in the FACILITY resource class
  - Make sure the FACILITY class is in RACLIST
  - RACF Example:
    - `RDEF FACILITY HWI.APPLNAME.HWISERV UACC(NONE)`
    - `PE HWI.APPLNAME.HWISERV CL(FACILITY) ID(STEVE) ACCESS(READ)`
    - `SETR RACLIST(FACILITY) REFR`

# BCPii Configuration – Security definitions

- Form the correct SAF profile name
  - *Specifying the CPC name*
    - The netid.nau is the SNA CPC gathered from the CPC or from the HWILIST List CPCs operation.
  - *Facility class profile name depends on type of connection*
    - CPC and activation profile connections
      - **CPC Profile: HWI.TARGET.netid.nau**
    - Image connections
      - **Image Profile: HWI.TARGET.netid.nau.imagename**
    - Capacity record connections
      - **Caprec Profile: HWI.CAPREC.netid.nau.caprec**

# BCPii Configuration - Security definitions

- Choose SAF authority to a specific resource
  - *Give at least READ access to the SAF profile name below in the FACILITY resource class (LIST, CONN, DISC, EVENT, QUERY)*
  - *Give at least UPDATE access for SET*
  - *Give at least CONTROL access of CMD*

© 2010 IBM Corporation

# BCPii Configuration – Security definitions

- Specify the community name on the CPC facility class definition
  - *Define community name in SAF for each CPC*
    - Associate the community name already defined on the support element for BCPii with the SAF CPC profile already defined
      - **APPLDATA field used to save this information on the CPC definition in SAF**
    - Can be done at the time of the CPC definition
    - BCPii propagates this community name to the support element when a connection is established for validation by the SE.

*The local CPC's community name MUST be defined to SAF or the BCPii address space will fail to come up.*

# BCPii Configuration – Security definitions

- CPC RACF Example:
  - *RDEF FACILITY HWI.TARGET.**IBM390PS.CPC1** UACC(NONE) APPLDATA(‘BCPII’)*
  - *PE HWI.TARGET.**IBM390PS.CPC1** CL(FACILITY) ID(STEVE) ACCESS(READ)*
  - *SETR RACLIST(FACILITY) REFR*
- Image RACF Example:
  - *RDEF FACILITY HWI.TARGET.IBM390PS.CPC1**.LP1** UACC(NONE)*
  - *PE HWI.TARGET.IBM390PS.CPC1**.LP1** CL(FACILITY) ID(STEVE) ACCESS(UPDATE)*
  - *SETR RACLIST(FACILITY) REFR*

# BCPii Configuration

- Configure the BCPii address space
  - *Make sure the high-level-qualifier.SCEERUN2 and high-level-qualifier.SCEERUN data sets are in the link list concatenation.*
  - *Optionally, configure the BCPii CTRACE parmlib member (CTIHWI00) to set up default CTRACE options for the address space.*
    - `TRACEOPTS`

```
    ON OPTIONS('ALL')
```

or `OPTIONS(‘MIN’)`

# BCPii Configuration – z/OS UNIX Event Authorization

- Optionally, set up the event notification mechanism for z/OS Unix callers
  - *Start the Common Event Adapter address space in full-function mode.*
  - *Grant a z/OS Unix BCPii application authority to listen to ENF68 events using profile definitions in the SAF SERVAUTH resource class.*

© 2010 IBM Corporation

# BCPii Configuration – z/OS UNIX Event Authorization…

- Start the Common Event Adapter address space in full-function mode.
  - *Add these security definitions (RACF example):*
    - *ADDUSER CEA DFLTGRP(SYS1) OMVS(UID(any) FILEPROCMAX(1024))*
    - *RDEFINE STARTED CEA.** STDATA(USER(CEA) GROUP(SYS1) TRACE(YES))*
    - *SETROPTS RACLIST(STARTED) REFRESH*
    - *PERMIT ’SYS1.PARMLIB’ GENERIC ACCESS(READ) ID(CEA)*
    - *SETROPTS GENERIC(DATASET) REFRESH*
  - *See z/OS Planning for Installation for more details*

# BCPii Configuration – z/OS UNIX Event Authorization…

- Build a resource name for the event to be protected
  - ***Hardware events are in the form:***
    - **03xx00yy**
      - Where xx is the event source
        - > ’01’x – CPC
        - > ’02’x – Image
      - Where yy denotes the particular event
        - > See the ENFREQ documention in *z/OS MVS Programming: Authorized Assember Services Reference EDT-IXG for particulars*
        - > *Example: ’01’x = CmdResp event*

# BCPii Configuration – z/OS UNIX Event Authorization…

- Grant a z/OS Unix BCPii application authority to listen to ENF68 events using profile definitions in the SAF SERVAUTH resource class.
  - RDEF SERVAUTH CEA.CONNECT UACC(NONE)
  - RDEF SERVAUTH CEA.SUBSCRIBE.ENF_006803010001 UACC(NONE)
  - PE CEA.CONNECT CL(SERVAUTH) ID(JOE) ACCESS(READ)
  - PE CEA.SUBSCRIBE.ENF_006803010001 CL(SERVAUTH) ID(JOE) ACCESS(READ)
  - SETR RACLIST(SERVAUTH) REFR

# BCPii Operations

- BCPii can be stopped, cancelled and forced
  - *P HWIBCPII (Stop command)*
  - *C HWIBCPII (Cancel command)*
  - *FORCE HWIBCPII (Force command)*
- BCPii can be restarted after it has been terminated
  - *S HWISTART (NOT "S HWIBCPII")*
- Almost all maintenance in BCPii does not require a re-IPL of z/OS.
  - *Stop, update the code, refresh LLA, restart BCPii*

# BCPii Diagnostics

- API Return Codes and Diag Area
- CTRACE
  - *BCPii cuts CTRACE records using SYSBCPII CTRACE comp*
  - *Default CTRACE CTIHWI00 parmlib member shipped*
  - *Two CTRACE options:*
    - Min
    - All
  - *Dump is taken whenever CTRACE is turned off*
- Symptom Records
  - *Limited first failure data capture for select problems*
- Support Element Tracing

# BCPii Documentation

- z/OS MVS Programming: Callable Services for High-Level Languages
  - *Primary BCPii documentation including:*
    - Installation instructions
    - BCPii API documentation
- z/OS MVS Programming: Authorized Assembler Services Reference, Volume 2 (EDT-IXG)
  - *BCPii's ENF68 documentation*
- z/OS MVS System Commands
  - *START HWISTART and STOP HWIBCPII commands*
- z/OS MVS Diagnosis: Tools and Service Aids
  - *BCPii's CTRACE documentation*
- z/OS MVS Initialization and Tuning Reference
  - *Miscellaneous documentation*
- z/OS MVS System Codes
  - *BCPii abend '042'x documentation*

# Questions?

- Any questions?

© 2010 IBM Corporation