

Adjunct Virtual Machine

Version 4

Lauren Maietti
z/VM Software Engineer



IBM Z Technical Disclosure Teleconference

Your participation in this TDT is based on your IBM PartnerWorld membership and the Confidential Disclosure Agreement (CDA) in place between your company and IBM. Any information disclosed at this event is IBM Confidential and covered by the terms and conditions of the CDA you have with IBM and cannot be shared with anyone outside your company.

The charts and replay will be available on the Dallas Secure Server.

All lines will be muted at the start of the call. If you need to ask a question, please un-mute and then mute when done. Please help by muting your phone if not asking a question.

Contents

Key Concepts	04	Profile EXEC Tips	38
Virtual Machine	05		
The Virtual Configuration	06	Use Cases	40
Introduction	07	References	41
What is an Adjunct	08		
Adjunct and Virtual Configurations	11	Summary	42
Adjunct Setup	13		
Adjunct Externals	14		
How to set up an Adjunct	16		
ADJUNCT Commands and the Virtual Configuration	24		

Key Concepts

Virtual Machine (VM)

- Virtualizes an instance of real hardware
 - Such as devices, disks, CPUs, instructions, etc.
- Can extrapolate, VMs are comprised of the virtual equivalents
 - E.g.. Virtual devices, virtual disks, virtual CPUs (vCPUs), “virtual” (or simulated) instructions, etc.
- Can control your VM’s resources and the way it runs
- You might say VMs are rather *configurable*

The Virtual Configuration

- Resources and parameters defined for a virtual machine
 - E.g. Defining a USER with 5 vCPUs, running z/Architecture, with 1G virtual storage (memory), and the userID LMTEST
- Configure through directory statements, configuration statements, commands
- Keep in mind an **adjunct** virtual machine is exactly that
 - A virtual machine and configuration in *addition* to a “main” or **principal** virtual machine and configuration

Introduction

What is an Adjunct Virtual Machine?

- **An additional VM that runs beside a principal VM**
- **When one is active the other is inactive**
- **Though associated with the principal configuration – has its own configuration**
 - Its own virtual storage (memory), virtual devices, a vCPU
 - Spool files /RDR is shared
 - Can IPL any instance of an operating system in the adjunct
 - Intended to run CMS

Hold on, this sounds familiar...

- **Have you heard of secondary or observer users?**
- **Maybe you've used (X)AUTOLOG?**

- **Secondary users (SECUSERS) receive console output from the primary user and can control the primary with the SEND command**
 - Only when the primary is disconnected
- **Observer users can only view the primary's console output**
 - Whether the primary is disconnected or connected
- **An (X)AUTOLOGged user runs disconnected, and the invoking user can direct commands with FOR/SEND**

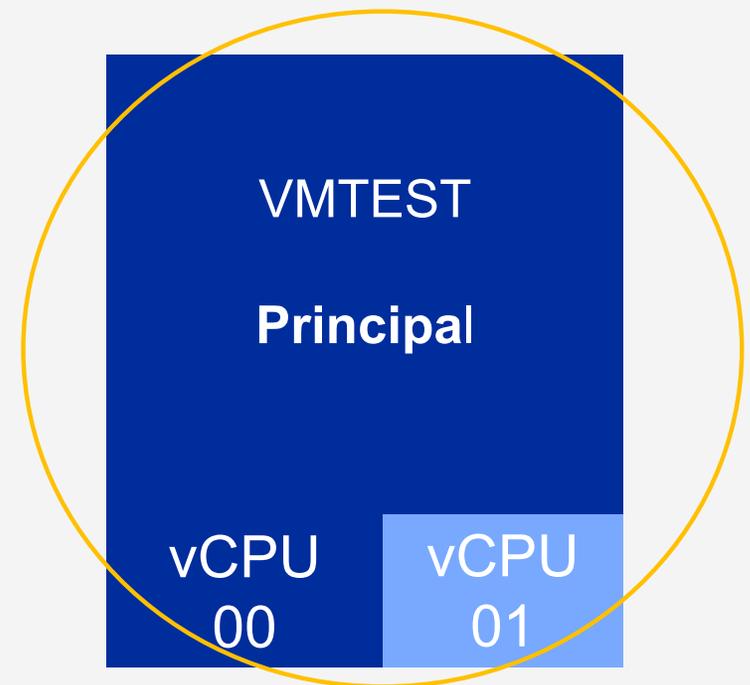
You're right, but Adjunct is different...

- **Adjuncts run “next” to the live principal VM**
- **Way to stop time and debug/control the principal**
 - **Provides a means to an operating for a VM doing some other work**
 - **Direct commands to the principal from the adjunct and vice versa**
 - **Able to access one another**
 - **Debug a live second level system**
 - **Simulate an “event” in the second level system to drive a code path**

A Closer Look at the Virtual Configuration: The Principal

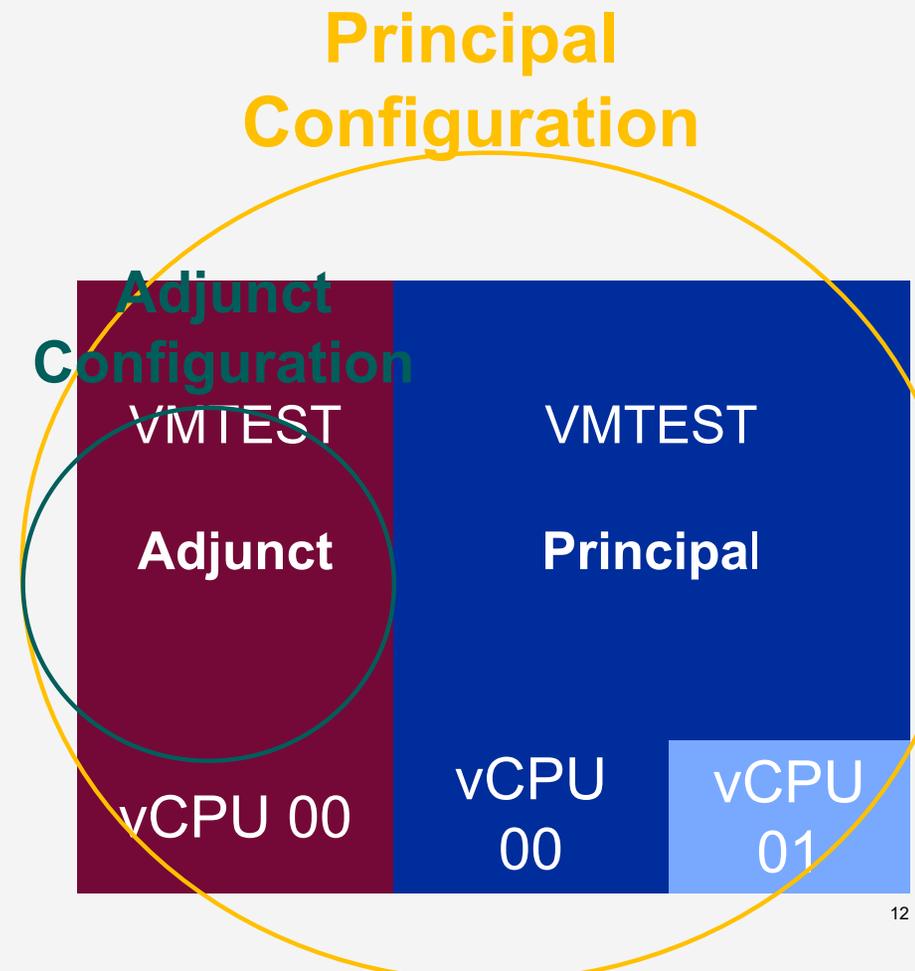
- Say a USER has the following virtual configuration
 - userID LMTEST
 - 2 vCPUs
 - 1G Virtual Storage
 - Running z/Architecture mode
 - Etc.
- Disclaimer: there are more virtual configuration attributes than what's depicted in this picture

Principal Configuration



A Closer Look at the Virtual Configuration: The Principal + Adjunct

- Similarly, the adjunct virtual configuration is defined as a USER
 - Has its own virtual storage, can run a different architecture mode, only 1 vcpu, userID, console, etc.
- The adjunct configuration is added to the principal's configuration and a few things are "filled in" for the adjunct
 - inherited userID, inherited spool, inherited reader, inherited privileges
 - Referred to as a template user
- Exists at the whim of the user



Adjunct Setup

Adjunct Externals...

- FEATURES ENABLE/DISABLE ADJUNCT
 - Configuration file statement to set the availability of the adjunct feature
- ADJUNCT directory statement
 - Include an adjunct template in a user definition
 - Allows that user to create and control the included adjunct
- SET ADJUNCTS ON/OFF
 - Dynamically enable or disable the availability of the adjunct feature

Adjunct Externals...

- ADJUNCT command
 - ADJUNCT START – logon/ipl an adjunct and suspend the principal
 - ADJUNCT STOP/SUSPEND – put the adjunct in a soft stop and reactivate the principal
 - ADJUNCT CP *command* – direct commands to the adjunct from the principal
 - ADJUNCT BEGIN/RESUME – reactive the adjunct and suspend the principal
 - ADJUNCT END – logoff/terminate the adjunct and reactive the principal
- QUERY ADJUNCT command

So, you want to set up an adjunct? Here's *Generally* How

- An optional FEATURES statement in the system configuration file
 - Adjuncts are enabled by default and there is a dynamic SET ADJUNCTS command to enable/disable
 - Remember, for permanent changes, choose the FEATURES statement
- Directory statements
 - USER statement/definition to define the adjunct template configuration
 - as many as needed to suit each principal's specific needs
 - Principal USER/IDENT statement/definition to include one ADJUNCT template in its configuration
- ADJUNCT Commands to control the adjunct
- QUERY ADJUNCT to determine the state of the adjunct.
 - Gives return codes which can be used for automation. Spoilers!

SYSTEM CONFIG Updates

- The adjunct FEATURE is enabled by default
- To disable/enable adjunct and retain the setting
 - Update the SYSTEM CONFIG file
 - To avoid FEATURE statement syntax errors, add a *new* FEATURES statement

FEATURES DISABLE ADJUNCTs

- or -

FEATURES ENABLE ADJUNCTs

- Don't forget to run CPSYNTAX!

SET ADJUNCTS Command



Privilege Class: A

- A System Administrator can dynamically change the availability of the adjunct feature
- This command does not produce a persistent setting – resets at IPL
- For any setting you want to make permanent, update the SYSTEM CONFIG with an ADJUNCT FEATURES statement.

Directory Updates

- All the following directory slides refer to your first level system's USER DIRECT
- Given a *very* basic common PROFILE to include in a USER definition:

PROFILE COMMON

IPL CMS PARM AUTO CR

CONSOLE 009

SPOOL 000C

SPOOL 000D

SPOOL 000E

LINK MAINT 190 190 RR

Directory Updates

- The common profile can be added to a template USER definition used to define the adjunct.

```
USER MRADJUNC NOLOG 32M 1G
```

```
    INCLUDE COMMON
```

```
    LINK * 191 1191 MR      (or LINK * 191 1191 RR beware of the double write access.)
```



```
    LINK * 193 191 MR
```

- Note:

1. UserID MRADJUNC is more of an identifier than it is a userID
2. NOLOG is required for an adjunct template
3. The asterisk in the LINK statement refers to the principal's configuration
4. This same adjunct template can be used in multiple user definitions
5. An adjunct may also be purpose-built for a particular principal

Directory Updates

```
➡ ADJUNCT — templateuser ⬅
```

- Now, the template user defined for the adjunct can be included in the principal configuration definition

```
USER LMTEST TESTPASS 256M 1G
```

```
INCLUDE COMMON
```

```
ADJUNCT MRADJUNC
```

```
MDISK 0191 3390 000 END LMT191 MRD RREAD WWRIT MMULT
```

```
MDISK 0193 3390 000 END ADJ191 MR ALL ALL ALL
```

- Note:
 1. There can only be one adjunct included per principal configuration
 2. Principal can be defined by a USER statement or an IDENTITY statement
 3. The MDISK 0191 statement shows the M R and D modes. The D stands for deferred; meaning not linked until the first occurrence of a LINK statement/command

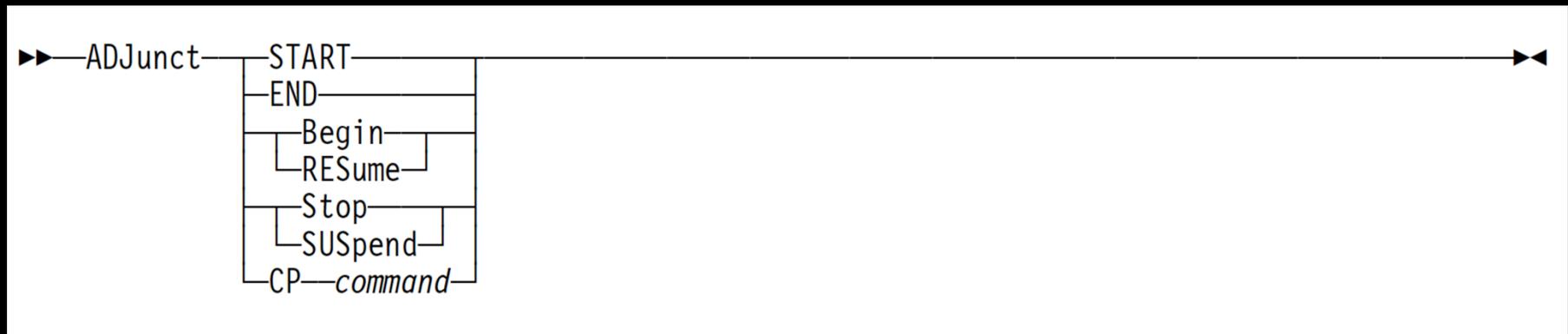
Directory Updates

- Once you've made all the updates, don't forget to run

DIRECTXA

Everything's set up, now what?

Controlling the Adjunct - ADJUNCT command

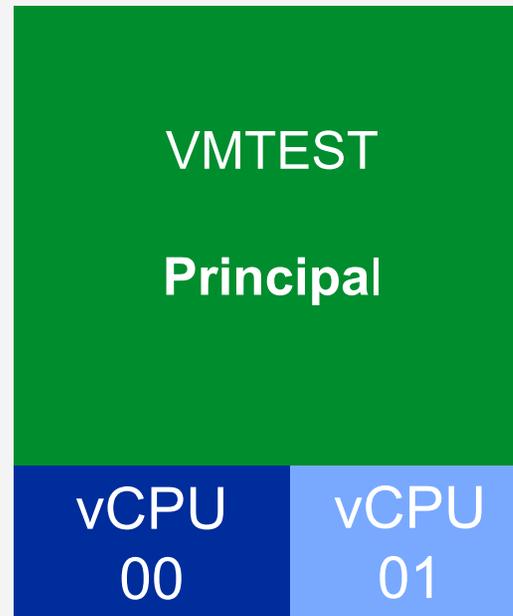


Privilege Class: G

- Used to control the adjunct from either the principal or the adjunct

Adjunct Commands and the Virtual Configuration

The Virtual Configuration: After LOGON



* Green signifies the virtual configuration is active

*Grey signifies the virtual configuration is inactive

ADJUNCT START

- Creates and IPLs an adjunct virtual machine
- While also suspending the execution of the principal configuration (“soft stop”)
- This command is only successful when the ADJUNCT FEATURE is ENABLED and an adjunct template is included in the principal issuing the command

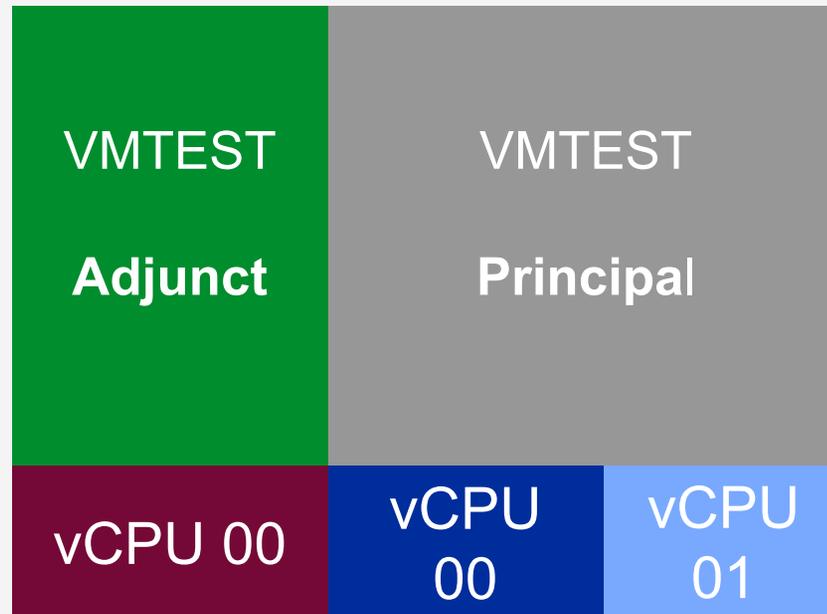
```
query adjunct
14:49:50 HCPYJT5768I LMTEST Adjunct does not exist
Ready(05768); T=0.01/0.01 14:49:50
```

```
adjunct start
15:01:17 * MSG FROM LMTEST : HELLO WORLD - USERID = MRADJUNC
```

```
query adjunct
15:02:20 HCPYJT5777I LMTEST Adjunct is currently active
Ready(05777);
```

```
adjunct start
15:03:06 HCPYJT5769E LMTEST Adjunct already exists
Ready(05769);
```

The Virtual Configuration: After ADJUNCT START



* Green signifies the virtual configuration is active

*Grey signifies the virtual configuration is inactive

ADJUNCT SUSPEND/STOP

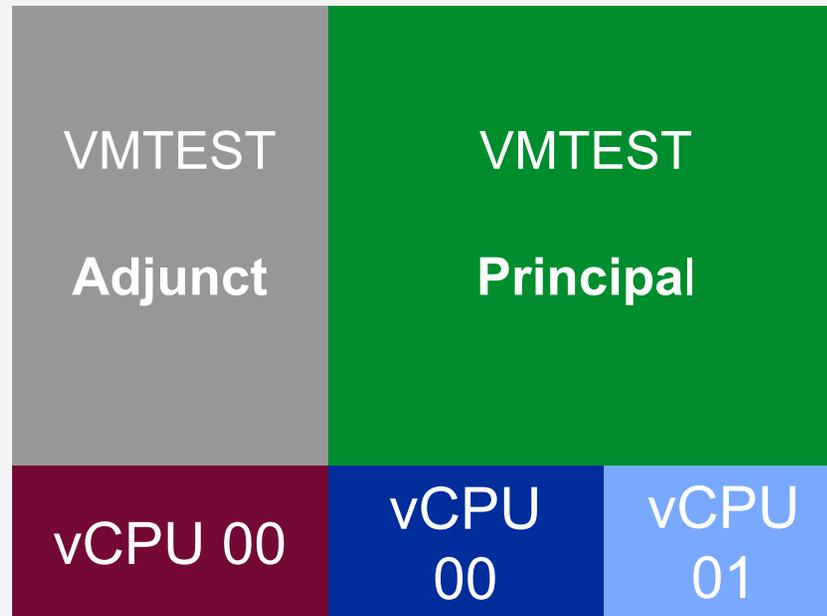
- Puts the adjunct in a soft stop - pause button
- Reactivates the principal user – play button

```
query adjunct
15:05:33 HCPYJT5777I LMTEST Adjunct is currently active
Ready(05777);
```

```
adjunct suspend
Ready;
```

```
query adjunct
15:06:40 HCPYJT5776I LMTEST Adjunct is suspended
Ready(05776);
```

The Virtual Configuration: After ADJUNCT STOP/SUSPEND



* Green signifies the virtual configuration is active

*Grey signifies the virtual configuration is inactive

ADJUNCT CP *command*

- Directs a CP command to the adjunct from the principal

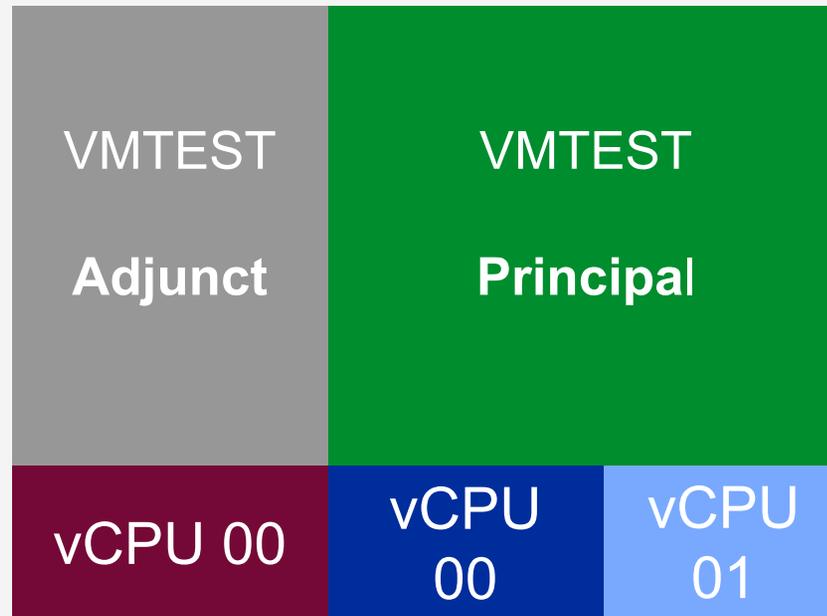
```
query adjunct
15:09:22 HCPYJT5776I LMTEST Adjunct is suspended
Ready(05776);
```

```
adjunct cp locate *
15:10:23 VMDBK      = 01C8A000 VMDBK-Real = 000000009801C000
Ready;
```

```
adjunct cp display 01C8A000.20
15:15:47 R01C8A000  00000000 00000000 00000000 00000000 04
15:15:47 R01C8A010 to 01C8A01F suppressed line(s) same as above ....
Ready;
```

```
query adjunct
15:16:15 HCPYJT5776I LMTEST Adjunct is suspended
Ready(05776);
```

The Virtual Configuration: After ADJUNCT CP *command*



* Green signifies the virtual configuration is active

*Grey signifies the virtual configuration is inactive

ADJUNCT BEGIN/RESUME

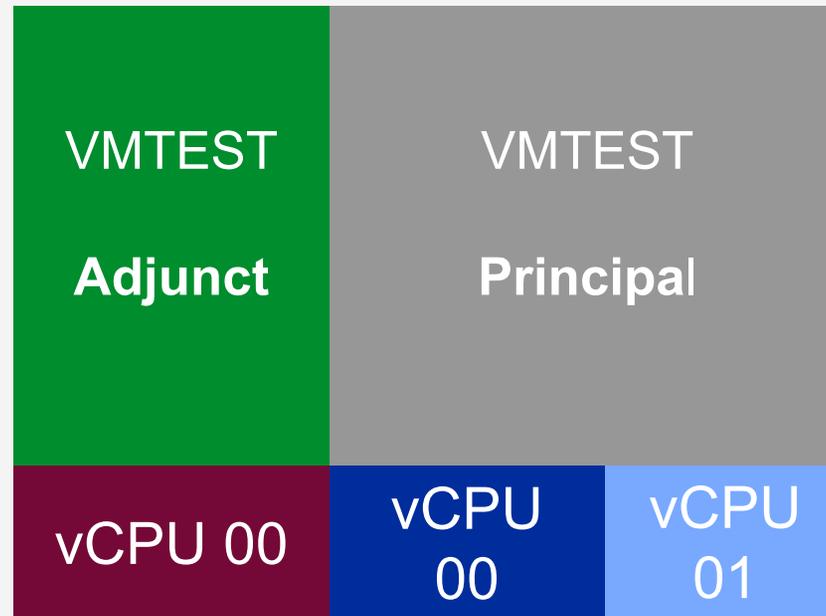
- Reactivates the adjunct - play button
- Puts the principal user in a soft stop – pause button

```
query adjunct  
15:17:23 HCPYJT5776I LMTEST Adjunct is suspended  
Ready(05776);
```

```
adjunct resume  
Ready;
```

```
query adjunct  
15:19:06 HCPYJT5777I LMTEST Adjunct is currently active  
Ready(05777);
```

The Virtual Configuration: After ADJUNCT RESUME



* Green signifies the virtual configuration is active

*Grey signifies the virtual configuration is inactive

Directing Commands to the Principal

- Use CPU * or CPU nnnn to direct commands to the principal from the adjunct

```
query adjunct
15:23:17 HCPYJT5777I LMTEST Adjunct is currently active
Ready(05777);
```

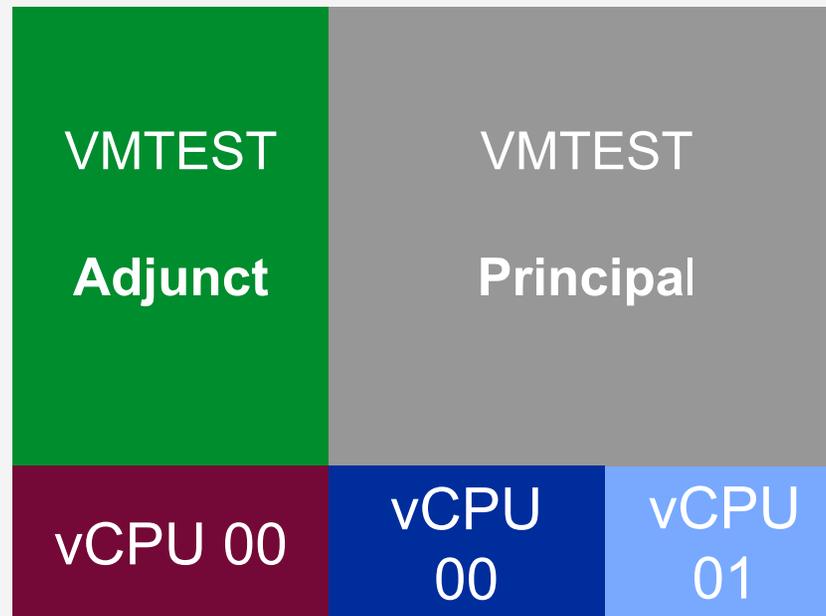
```
cpu * cp locate *
15:27:09 VMDBK      = 01DC2000 VMDBK-Real = 0000000098081000
Ready;
```

```
cpu * cp display 1dc2000.20
15:27:53 R01DC2000  17EE90DE 10045010 D00818FD 18D198E1 F6
15:27:53 R01DC2010  F00C5900 B29047D0 C0301840 5850C220
Ready;
```

```
cp display 1dc2000.20
15:28:28 R01DC2000  99854085 A79796A2 85408595 83F7F740 F6
15:28:28 R01DC2010  839693A4 9495A240 828987A4 955E827E
Ready;
```

```
query adjunct
15:31:52 HCPYJT5777I LMTEST Adjunct is currently active
Ready(05777);
```

The Virtual Configuration: After CPU nnnn CP *command*



* Green signifies the virtual configuration is active

*Grey signifies the virtual configuration is inactive

ADJUNCT END

```
query adjunct  
15:33:18 HCPYJT5777I LMTEST Adjunct is currently active  
Ready(05777);
```

```
adjunct end  
Ready;
```

```
query adjunct  
15:36:07 HCPYJT5768I LMTEST Adjunct does not exist  
Ready(05768);
```

```
adjunct resume  
15:36:53 HCPYJT5768E LMTEST Adjunct does not exist  
Ready(05768);
```

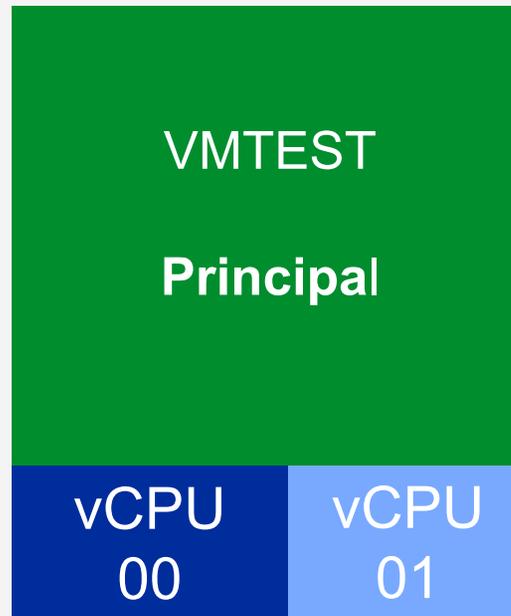
```
adjunct suspend  
15:37:27 HCPYJT5768E LMTEST Adjunct does not exist  
Ready(05768);
```

```
adjunct end  
15:38:01 HCPYJT5768E LMTEST Adjunct does not exist  
Ready(05768);
```

```
query adjunct  
15:38:34 HCPYJT5768I LMTEST Adjunct does not exist  
Ready(05768);
```

- Terminates the adjunct and reactivates the principal

The Virtual Configuration: After ADJUNCT END



* Green signifies the virtual configuration is active

*Grey signifies the virtual configuration is inactive

Profile EXEC Tips

- The principal's Profile EXEC is used for the adjunct
- Use of the QUERY ADJUNCT return codes can help to determine where the profile is executing

```

/*****
/*
/* Determine if the Principal or the Adjunct is
/* active.
/*
/*
*****/
adjflag = 0 /* Assume we're not in adjunct */
'EXECIO * CP ( SKIP STRING QUERY ADJUNCT' /* Issue QUERY Adjunct */
if right(rc,4) = '5777' then adjflag = 1 /* If HCP5777I userid Adjunct is currently active
The adjunct is running */
else if right(rc,4) = '5776' then adjflag = 2 /* If HCP5776I userid Adjunct is suspended
The principal is running */

```

- ADJFLAG can be used to SET SCREEN differently for the principal and the adjunct as well as link/access specific disks/filemodes for each VM. See examples in the next slide...

Profile EXEC Tips: Use of ADJFLAG

- Ensure the Principal's 191 disk is accessed as read only in the adjunct setup area. Otherwise, it's possible to corrupt the disk.
 - Instead, create a tdisk or attach another disk to be used as the adjunct's 191
- SFS connection uses the userID to connect to and from the SFS. Since the adjunct shares the same userID as the principal ensure the principal "disconnects" from the SFS the adjunct wants to connect to and vice versa.
- When connecting an SFS to the adjunct that the principal is connected to:
 - A system reset is required in the principal
 - A check should be added to both the principal and adjunct areas of the profile to conditionally connect/access the SFS based on the activity/connection of the other

Use Cases

- Debug what is running in the principal VM (ex. a second level guest)
 - There are a few ways to START/RESUME the adjunct:
 - PA1 to principal's CP READ
 - While stepping through a trace trap in CP read
 - » Can set TRACE with the CMD option to switch to the adjunct when the trace hits
 - CP exit (CP1STLVL) - from a second level system
 - Display the principal's storage to see what is happening in the guest
- Test
 - Simulate events for a second level guest
 - Store into the principal's memory from the adjunct to create specific test environments/circumstances

References

- Primary Sources in z/VM Library - <https://www.vm.ibm.com/library/> (7.1 and up)
 - CP Planning and Administration
 - CP Command and Utility Reference

Summary

- Adjuncts are virtual machines that have their own virtual configuration but are included in a principal's configuration
 - Runs beside its principal – only one can be active at a time
 - Main use is to debug a live second level system
 - Can run any operating system but is meant to run CMS
- *Have any fun use cases for adjunct? Email me:*

Imaiett@us.ibm.com