IBM Education Assistance for z/OS V2R2

Item: RRSF Dynamic MAIN Switching
Element/Component: RACF/RRSF

Material current as of May 2015

Agenda

- Trademarks
- Presentation Objectives
- Overview
- Usage & Invocation
- Migration & Coexistence Considerations
- Installation
- Presentation Summary
- Appendix

Trademarks

- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.

Presentation Objectives

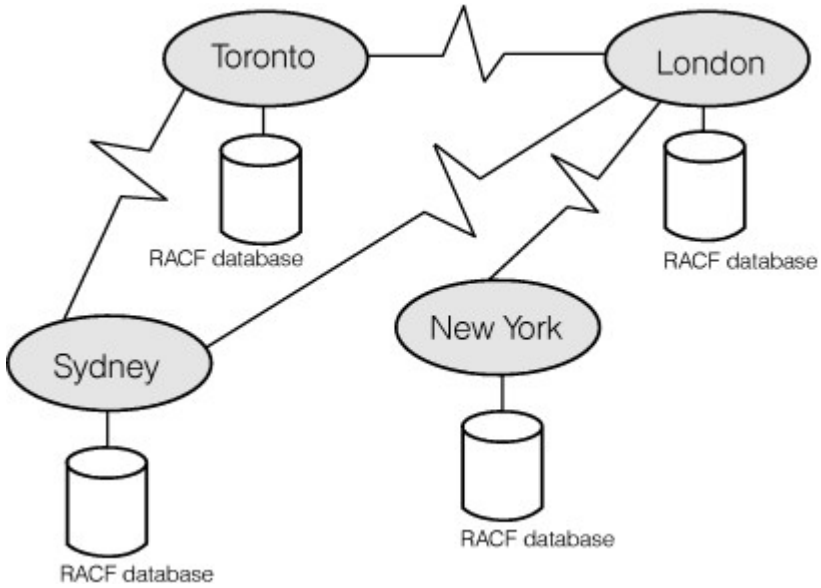
- Conduct a brief review of what RRSF is
- Discuss switching the MAIN system in an RRSF multisystem node: past, future, and in-between
- Discuss new programming interfaces with which to obtain RRSF configuration information

Overview – What is RRSF?

- The RACF remote sharing facility allows RACF to communicate with other z/OS systems that use RACF, allowing you to maintain remote RACF databases.
- Benefits of RRSF support for the security administrator include:
  - Administration from anywhere in the RRSF network.
  - User ID associations, supporting directed commands and password synchronization.
  - Automatic synchronization of databases by RACF class.
- RRSF is designed in roughly three layers:
  - Application layer: Administrative commands and profiles
  - Presentation layer: command execution and return of command output and error and informational messages
  - Transport layer: Communication protocols used to transmit requests

Overview – The RRSF network

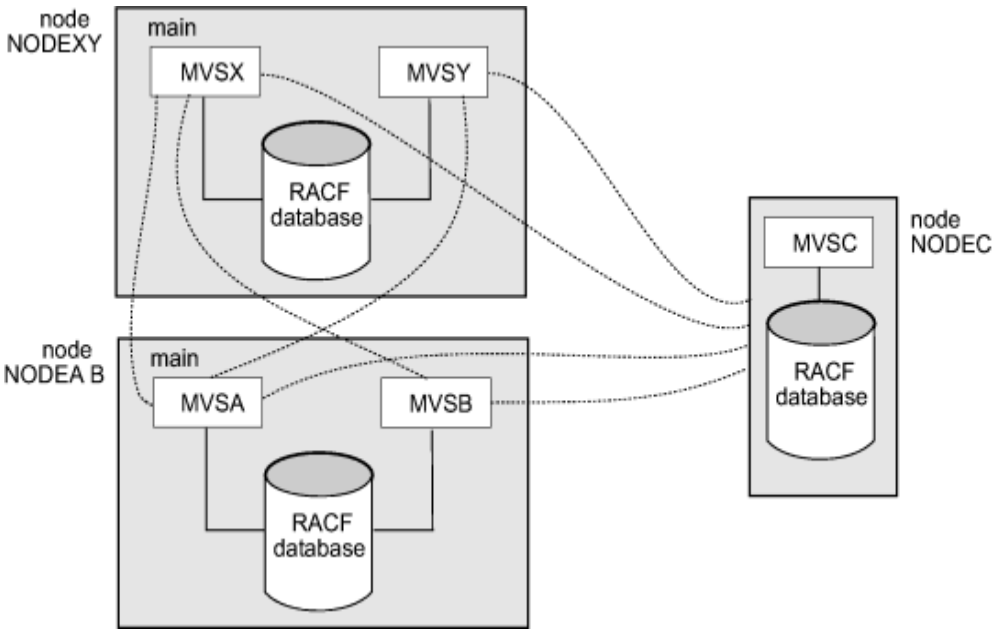
- Consists of **nodes**
  - Local node: The one I'm logged on to at the moment
  - Local node can run in “local mode”, where there are no remote nodes
  - Remote nodes (all the others)
- The TARGET operator command is used to define, modify, and delete, and list nodes, as well as to de/activate them
- TARGET commands are contained within the RACF parameter library, and are executed automatically when the RACF subsystem starts
- The RACF parameter library member is specified in your started procedure JCL
- RACF parameter library members can be “chained together” using the SET INCLUDE(xx) command



Sample RRSF network containing 4 nodes

Overview – Multi-System Node (MSN)

- A set of systems sharing a RACF database (can be in a SYSPLEX, or simply on shared DASD)
- Managed with the TARGET command by specifying both NODE and SYSNAME
- All Single System Nodes (SSNs) send requests only to the MAIN system of a MSN
- All peer systems of an MSN send requests only to SSNs, and to the MAIN systems of remote MSNs
- Peer systems do not speak with each other, and do not speak with non-MAIN systems of remote MSNs



Sample RRSF network containing two Multi-System Nodes and a Single System Node

Overview – Workspace data sets (i.e. checkpoint files)

- VSAM data sets that RACF uses to temporarily hold data that RACF is sending from one node to another.
- RACF deletes data from the workspace data sets when it receives confirmation that the data has been successfully processed at the receiving node.
- RACF uses two workspace data sets, the INMSG data set and the OUTMSG data set, for the local node and for each of its remote nodes.
  - The INMSG data set is used to temporarily hold requests that are being sent to the local node from itself or from a remote node (e.g. commands directed to the local node, or output from RACF commands, application updates, and password changes that were directed to a remote node)
  - The OUTMSG data set is used to temporarily hold requests that are being sent to a remote node (e.g. commands, application updates, and password changes directed from the local node, or output to be returned to a remote node)
- Requests are queued to the files while a connection is DORMANT. Queued work is sent when the connection becomes OPERATIVE ACTIVE.
- Requests are “casually encrypted” while checkpointed

Overview

- Problem Statement / Need Addressed
  - Switching the MAIN system in a multisystem node is a brutal “11”-step manual process that is not feasible to implement for short-term changes. For example, to accommodate an IPL-window on the MAIN system without suffering an “outage”, as perceived by users.
- Solution
  - This process is essentially replaced by the issuance of a single command (of course, with caveats)
- Benefit / Value
  - Allows you to avoid even minor outage windows
  - Or major ones where remote checkpoint files fill up with queued work
  - Allows you to move RRSF workload off of a busy system
  - New programming interfaces introduce possibility of automating the switch entirely

Overview – the dreaded 11-step process prior to V2R2

1) Drop TSO/E and JES on the original local main system.

2) On the original local main system, issue the RACF STOP command to stop the RACF subsystem.

3) Make connections dormant:
   1) On the local system that is to be the new main, issue a TARGET DORMANT command for its local connection. Also **issue TARGET DORMANT commands to make all connections with remote nodes dormant.**
   2) **On each remote node, issue TARGET DORMANT commands** for the original and new main systems. Do not perform step 7 until the INMSG files for the original and new main systems on each remote node have drained.

Issue TARGET LIST commands to verify that the INMSG data sets on the local node have been drained **before you go on to the next step.**

4) If the workspace data sets for the original main system and the new main system are not on shared DASD with a shared catalog, copy the workspace data sets for the original main system to DASD accessible to the new main system, using the same workspace data set names.

5) On the new main system, issue a TARGET MAIN command to make it the main system. **If you have not specified the prefixes for the workspace data sets and the LU names for the member systems consistently** in the TARGET commands that defined the local multisystem node, **this step will fail.**

6) Issue the same TARGET MAIN command that you issued in step 5 **on each nonmain system** on the local multisystem node. Issue this command on the original main system only if it is to remain in the multisystem node.

7) Issue TARGET LIST commands to **verify that the INMSG data sets on the remote nodes have been drained** before you perform this step. **On each remote system** (that is, all remote systems of all remote nodes), issue the same TARGET MAIN command that you issued in step 5.

8) On the new main system, issue TARGET OPERATIVE commands to make the connection with itself and all connections with remote nodes operative.

9) On each remote system (that is, all remote systems of all remote nodes), issue TARGET OPERATIVE commands for the original main (if it is to remain in the multisystem node) and new main systems.

10)Update the TARGET commands in the RACF parameter libraries for all systems on all nodes in the RRSF network to reflect the new main system. If you fail to update the RACF parameter library for a system, the next time that system has its RACF subsystem restarted or is IPLed, the original TARGET commands will be issued, and requests and returned output will accumulate in the wrong OUTMSG workspace data set. However, RACF will issue appropriate error messages and prevent communications.

11)If the original main system is still part of the multisystem node, (and assuming that you have updated its RACF parameter library as discussed in step 10) restart the RACF subsystem, TSO/E and JES on the original main system.

Usage & Invocation...performing a switch on V2R2 in a sysplex

- When the MSN is in a sysplex:
  1)TARGET NODE(msn-name) SYSNAME(new-main) PLEXNEWMAIN
    - From **any** system in the MSN

```
IRRM110I SYSTEM new-main HAS REPLACED SYSTEM old-main AS THE MAIN SYSTEM IN
         LOCAL NODE msn-name.
```

  2)Optionally, update the RACF parameter library

-

- Wasn't that nicer than the 11-step process?
  - There are no actions required on any remote system
- Caveat: This is in a perfect world where the entire network (or at least all the systems in the switching MSN) is at V2R2 or higher, and all non-MAINs have been enabled for switches (enablement described soon)
  - More on mixed-release MSNs and networks later...

Usage & Invocation...performing a switch on V2R2 in a non-sysplex

- When the MSN is **not** in a sysplex (again, in a perfect world):

1) From the old (current) MAIN system issue:
   TARGET NODE(msn-name) SYSNAME(new-main) NEWMAIN

```
IRRM098I DRAINING SYSTEM OF INBOUND WORK. DO NOT INITIATE THE MAIN SWITCH ON
THE NEW MAIN SYSTEM UNTIL MESSAGE IRRM099I IS ISSUED.
IRRM099I ALL INBOUND WORK HAS COMPLETED. IT IS NOW SAFE TO INITIATE THE MAIN
SWITCH ON THE NEW MAIN SYSTEM.
```

2)
3) From the new MAIN system, issue:
   TARGET NODE(msn-name) SYSNAME(new-main) NEWMAIN

```
IRRM102I SYSTEM new-main IS NOW THE MAIN SYSTEM IN LOCAL NODE msn-name.
```

4)
5) From the remaining peer systems, issue:
   TARGET NODE(msn-name) SYSNAME(new-main) NEWMAIN
6)
7) Harden the change in parmlib, if you expect reIPLs before switching back, or if the change is intended to be "permanent".
   - By moving the MAIN keyword in the TARGET definitions

- Still much better than the 11-step process! Still no actions required on any remote system

Usage & Invocation...Setup for dynamic MAIN switches

- First, you must enable non-MAIN systems to become MAIN
  - Meaning, it must have now have checkpoint files for, and communication channels to, remote non-MAIN systems
  - And those remote non-MAINs must be enabled as well
- Accomplished with a new SET FULLRRSFCOMM command
  - Add SET FULLRRSFCOMM to the top of the RACF parameter library member used by an MSN and wait for nature to take its course at the next IPL
  -
  - Or, to do it **right now**: on each non-MAIN system, issue
    - SET FULLRRSFCOMM
    - RESTART CONNECTION NODE(remote-MSN1) SYSNAME(*)
    - RESTART CONNECTION NODE(remote-MSN*n*) SYSNAME(*)
    - (or simply RESTART CONNECTION)
    - And put SET FULLRRSFCOMM in parmlib, as above

What are the principles at work here?

1)MAIN changes are communicated remotely solely by means of handshaking, which occurs only at connection establishment
   - Old and new MAINs will be automatically breaking and then reestablishing connections during the switch. Old MAIN will be draining work in this window to preserve order of requests.

2)If either partner is downlevel, the existing handshaking rules apply
   - A mismatch in MAIN definitions prevents the connection, and requests are queued until the situation is resolved

3)If both partners are uplevel, the “local” partner will accept the MAIN status of the remote partner as asserted by that partner, regardless of its local TARGET definitions, with only the following exception.

4)If the remote partner asserts that it is MAIN, but an OPERATIVE-ACTIVE connection to a MAIN system in that MSN already exists for the local system, the connection is rejected with a handshaking error/message.

5)A downlevel non-MAIN system, and a disabled uplevel non-MAIN system, will never attempt to connect outbound to a non-MAIN, and will never accept such a connection

How the remote uplevel system reacts to a MAIN switch

- Accepts old MAINs reconnection as non-MAIN
  - TARGET list would display it as “EX-MAIN”
  - Will return any pending output to old MAIN
  - Will accept new work and returned output from old MAIN
  - Continues to checkpoint outbound work to the MSN in the workspace file (OUTMSG) of the recent MAIN, but does not send the work
- Accepts new MAIN's reconnection as MAIN
  - Sends work checkpointed in ex-MAIN's OUTMSG file
    - Painstakingly insures that all work is sent in the proper order
  - Checkpoints all new work in new MAIN's OUTMSG file
  - Stops showing old MAIN as “EX-MAIN”. Simply omits any MAINish reference, and it now appears as any other non-MAIN system does
- On a re-IPL, will accept remote's assertion of non/MAINliness, thus continuing to honor the switch

How will a remote downlevel system react? MAIN or SSN

- Mismatch on either side's conception of either side's MAIN status results in a handshaking error:
  - For APPC:
    - IRRI014I ERROR: LOCAL NODE node-name [SYSNAME system-name] AND PARTNER NODE node-name [SYSNAME system-name] HAVE CONFLICTING TARGET STATEMENTS WITH {LOCAL | REMOTE} LUNAME luname. REASON CODE reason-code.
  - For TCP/IP
    - IRRI016I ERROR: LOCAL NODE node-name [SYSNAME system-name] AND PARTNER NODE node-name [SYSNAME system-name] HAVE CONFLICTING TARGET STATEMENTS WITH THE {LOCAL | REMOTE} SYSTEM. REASON CODE reason-code.
  - Where reason-code = 4: "There is no agreement if the system on the multisystem node is the MAIN system."
- Connection remains in OPERATIVE-PENDING-VERIFICATION state on both sides. Remote system checkpoints outbound work.
-
- i.e. exact same behavior as today. But uplevel systems can benefit from the switch.

How will a remote downlevel system react? non-MAIN

- A downlevel non-MAIN system, and a disabled uplevel non-MAIN system, will never attempt to connect outbound to a non-MAIN
  - The connection is in the DEFINED state from its point of view
-
- And if an enabled non-MAIN attempts to connect to it, the connection will be rejected.
  - `IRRC059I (>) RACF REMOTE SHARING CONNECTION TO NODE node-name SYSNAME system-name DID NOT COMPLETE SUCCESSFULLY. REMOTE NODE REPORTS ITS STATE AS DEFINED.`

  (This rejection actually occurs prior to handshaking.)
- The connection goes to DORMANT-REMOTE state on the initiating side.
- This is simply a description of existing behavior

How will a local downlevel system react?

- Firstly, at least the current **and** intended new MAIN must be at V2R2 in order to use the new function
  - Otherwise, you will use the 11-step process, or a modified (and slightly simplified) version of it.
- Local peers will need their configuration explicitly changed in order to reflect the new MAIN.
  - TARGET NODE(*thisnode*) SYSNAME(*newmain*) MAIN
  - Note this isn't strictly necessary from a functional perspective. It just insures that if somebody issues TARGET LIST on this peer, they will get output which accurately identifies the current MAIN system

Mixed-release networks: the bottom line

- You will probably not have the luxury of upgrading all of your systems at the same time
- But as you do, you will be able to get some benefit. The benefit will increase as you upgrade more systems
- Downlevel systems are no worse off than they are today
  - There will be some noise as handshaking errors are generated across the switch
- As long as you have downlevel systems, you should view a MAIN switch as temporary, switching back to the original at your earliest convenience so that remote OUTMSG files do not fill up.
- When all systems are at V2R2, you can fire away at will, making MAIN switches as temporary or permanent as you desire.

Usage & Invocation...OPERCMDS protection of TARGET command

- All TARGET keywords are protected at the keyword level:
  - Subsys-name.TARGET.keyword
    - RACF.TARGET.MAIN
- For PLEX/NEWMAIN, we add the RRSF node and system name:
  - RACF.TARGET.NEWMAIN.POKNODE.POKSYS1
- Controls not only who can initiate MAIN switches, but which systems they are allowed to switch to
- Node name qualifier allows for same system name in different nodes when OPERCMDS is being propagated
- Protection at the keyword-only level requires a generic profile
  - RACF.TARGET.NEWMAIN.*

Usage & Invocation...obtaining RRSF information using APIs

- The R_admin callable service (IRRSEQ00) has a new function code AMN_XTR_RRSF to extract
  - Configuration settings made by the SET command related to RRSF
  - Some general RACF subsystem information (e.g. command prefix)
  - Configuration and operational information on all nodes/systems, local and remote
    - Values of various TARGET keywords used to define the node
    - Operational state, connection statistics, number of checkpointed requests
    - Partner handshake data, such as its release, template, and dynamic parse levels
- Authorization is the same OPERCMDS access you would need in order to issue SET LIST and TARGET LIST commands

Usage & Invocation...obtaining RRSF information using APIs ...

- IRRXUTIL is updated to make the R_admin data available to REXX callers
- Uses fake class and profile names of "_RRSFEXTR"

```
rc=irrxutil("EXTRACT","_RRSFEXTR","_RRSFEXTR","STEM")
```

- Contains variables for all the data returned by R_admin
- Plus additional convenience variables
  - Ability to directly access local node
  - Ability to directly access any given node based on node name/system name/protocol instance

Example: which remote targets support dynamic MAIN switches?

invoke from console with: F AXR,*exec-name*

```
myrc=IRRXUTIL("EXTRACT","_RRSFEXTR","_RRSFEXTR","RRSF")
Do i = 1 to RRSF.0
  If ¬RRSF.i.LOCAL Then
    Do
      /*************************************************************/
      /* Construct the part of the message that identifies the     */
      /* remote node name, and if applicable, system name.         */
      /*************************************************************/
      target = "NODE("||RRSF.i.NODE||")"
      If RRSF.i.MSN Then
        target = target || " SYSNAME("||RRSF.i.SYSNAME||")"
      /*************************************************************/
      /* If the partner parse level (a character string) is null,  */
      /* then we have never connected to the node/system. This exec */
      /* is most helpful if all connections have been established  */
      /* at some point in the past so that handshake data was      */
      /* exchanged.                                                */
      /*************************************************************/
      If Length(RRSF.i.PARTNER_PARSE) = 0 Then
        rc = AXRWTO(target "release level is unknown")
      Else Do
        If RRSF.i.PARTNER_OS_VERSION /= 0 Then
          rc = AXRWTO(target "supports dynamic MAIN switches")
        Else
          rc = AXRWTO(target "does not support dynamic MAIN switches")
      End
    End
End
```

Migration & Coexistence Considerations

- Note that the 11-step process was not originally supported for TCP/IP when it was introduced as an RRSF communication protocol in V1R13
- Support was added in V2R1
- So if you have TCP/IP connections and wish to start using a modified 11-step process, those systems must be at V2R1
  - Or you could convert them back to APPC temporarily

Installation

- You could consider optionally, proactively, adding SET FULLRRSFCOMM to the RACF parameter library used by your MSNs, so that they are ready to go when you IPL V2R2
  - Downlevel systems will receive harmless error message when executing the command
- Consider the additional VSAM files that will be allocated, and network sockets that will be opened, even if you never perform a switch
- We already recommend sharing the same RACF parameter library among all systems in an MSN. This becomes even more crucial with this support. If this is not currently the case, consider making this consolidation right now, on your pre-V2R2 systems.

Presentation Summary

- Switching the MAIN system today is manual, complicated, error-prone, and should only be undertaken when a permanent change is required
- Dynamic MAIN switch via PLEX/NEWMAIN vastly simplifies/automates this process
- The benefits of dynamic MAIN switch will increase over time as you upgrade systems to V2R2. In the interim, you are no worse off than you currently are.
- When all of your systems are at V2R2, you can switch as frequently as you'd like
- Using enhancements to IRRXUTIL, you might be able to automate switches based on your own criteria. It certainly allows for easier scripting to obtain information about your live network
  - e.g. How many of my remote systems are uplevel?

Appendix

- *RACF: System Programmer's Guide* (SA23-2287)
- *RACF: Command Language Reference* (SA23-2292)
- *RACF: Security Administrator's Guide* (SA23-2289)
- *RACF: Callable Services* (SA23-2293)
- *RACF: Macros and Interfaces* (SA23-2288)
- *RACF: Messages and Codes* (SA23-2291)
- *RACF: Data Areas* (GA32-0885)
- *RACF: Diagnosis Guide* (GA32-0886)