IBM Education Assistance for z/OS V2R3

New Function: HC SYS Filter

Element/Component: BCP Health Checker

Trademarks

- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.
- The term Health Checker is used as short form of “IBM Health Checker for z/OS” in this presentation.
- The term “health check” or just “check” is used as short form of “health check for the IBM Health Checker for z/OS” in this presentation.
- Additional trademarks: None

© 2017 IBM Corporation

Session Objectives

Learn about

- The new **filter support for HZSPRMxx** (Health Checker) parmlib members
- A new option to **“syntax check”** whole HZSPRMxx parmlib members

© 2017 IBM Corporation

Overview

Problem Statement / Need Addressed

- Health Checker supports customization of individual health checks via POLICY statements. Typical overrides specified in such a POLICY are a check's PARM string, the SEVERITY of a check, or the check INTERVAL
- Health Checker POLICY statements are recommended to be stored in, and specified via, HZSPRMxx PARMLIB members.
- In a SYSPLEX environment customers like to share such HZSPRMxx members between systems in the SYSPLEX.
- **While a majority of POLICY statements might apply equally to all systems in the SYSPLEX, there are always a few exceptions, i.e. where a particular system, or a subset of systems, requires slightly different health check overrides.**

© 2017 IBM Corporation

Overview – Solution

Solution

- HC Sys Filter introduces the capability to provide **conditional phrases,** “filters” (for example by system name), for POLICY statements as well as other statements allowed in HZSPRMxx.

Benefit / Value

- Allows one HZSPRMxx PARMLIB member to contain **special cases** for some systems in the SYSPLEX where that one PARMLIB member is being shared.

- **Eliminates the need to maintain multiple**, yet very similar, sets of parmlib members, for both HZSPRMxx and IEASYSxx (where the HZS system parameter determines which HZSPRMxx members the system should initially use).

© 2017 IBM Corporation

Overview – Added Benefit

An additional function provided by this lineitem is the option to **check whole HZSPRMxx** parmlib members **for syntax**.

© 2017 IBM Corporation

Usage & Invocation

- The filtering within HZSPRMxx parmlib members is done via "**WHEN**" statements, setting a scope for the to be filtered other statements

- For example, to mark checks CHECK1 and CHECK3 inactive only on the systems named SY07, SY08, and SY09, and to also mark CHECK2 inactive for system SY08 only, you could use:

```
WHEN(&SYSNAME. IN ('SY07','SY08','SY09'))
   DO
    ADDREP POLICY UPDATE CHECK(OWNER1,CHECK1) INACTIVE
           REASON('Not needed on system 7,8, and 9')
           DATE(20150811)
    WHEN(&SYSNAME. = 'SY08')
     DO
      ADDREP POLICY UPDATE CHECK(OWNER2,CHECK2) INACTIVE
             REASON('Not needed on system 8') DATE(20150811)
     END /* SY08 */

    ADDREP POLICY UPDATE CHECK(OWNER3,CHECK3) INACTIVE
           REASON('Not needed on system 7,8, and 9')
           DATE(20150811)

  END /* SY07, SY08, SY09 */
```


© 2017 IBM Corporation

Usage & Invocation – WHEN readability

- Indentation can be used for readability, and is recommended, but it is not required
- Adding comments after an END can help to document and understand the scope, but there is no formal “matching label” concept for DO and END
- Debug aid / completion message in hardcopy log:
- **HZS0118I** `WHEN EVALUATED TO {TRUE|FALSE} IN PARMLIB MEMBER=HZSPRMxx ON LINE n`

© 2017 IBM Corporation

Usage & Invocation – WHEN high-level syntax

- The general syntax is **`WHEN ( [<WhenExprList>] ) [DO] <stmts> [END]`**
  - You specify a **conditional expression** which will be dynamically evaluated when HZSPRMxx is being read/parsed by Health Checker
  - If the condition is evaluated to be **false**, the subsequent statements will only be checked for syntax, but otherwise will be “ignored” (= not used to change anything within the Health Checker framework)
  - If the condition is evaluated to **true**, the subsequent statements will be processed “as usual”
- Note that WHEN statements **can be nested**, so the `<stmts>` from above includes WHEN statements, just like any other, existing top-level HZSPRMxx statements (ADD POLICY statements for example)

© 2017 IBM Corporation

Usage & Invocation – WHEN scope

- The scope of a WHEN can be explicitly specified via **DO … END**
- If no DO … END is specified, the scope starts after the conditional expression and ends in one of the **three following cases**

© 2017 IBM Corporation

Usage & Invocation – WHEN scope ends... (1)

- At the end of the current HZSPRMxx parmlib member

```
WHEN(&SYSNAME. = 'SY08')
    ADDREP POLICY UPDATE CHECK(OWNER2,CHECK2) INACTIVE
           REASON('Not needed on system 8')
           DATE(20150811)
    ADDREP POLICY UPDATE CHECK(OWNER3,CHECK3) INACTIVE
           REASON('Not needed on system 8')
           DATE(20150811)

<end of HZSPRMxx>
```


© 2017 IBM Corporation

Usage & Invocation – WHEN scope ends... (2)

- When the **next WHEN** statement is encountered

```
WHEN(&SYSNAME. IN ('SY07','SY08','SY09'))
   ADDREP POLICY UPDATE CHECK(OWNER1,CHECK1) INACTIVE
          REASON('Not needed on system 7,8, and 9')
          DATE(20150811)
WHEN(&SYSNAME. <> 'SY08')
   ADDREP POLICY UPDATE CHECK(OWNER2,CHECK2) INACTIVE
          REASON('Not needed on any system, but system 8')
          DATE(20150811)
```


© 2017 IBM Corporation

Usage & Invocation – WHEN scope ends... (3)

- When the **END** statement of an “outer”, **containing WHEN** is reached

```
WHEN(&SYSNAME. IN ('SY07','SY08','SY09'))
  DO
      ADDREP POLICY UPDATE CHECK(OWNER1,CHECK1) INACTIVE
             REASON('Not needed on system 7,8, and 9')
             DATE(20150811)

      WHEN(&SYSNAME. = 'SY08')
        ADDREP POLICY UPDATE CHECK(OWNER2,CHECK2) INACTIVE
               REASON('Not needed on system 8')
               DATE(20150811)

        ADDREP POLICY UPDATE CHECK(OWNER3,CHECK3) INACTIVE
               REASON('Not needed on system 8')
               DATE(20150811)

  END /* SY07, SY08, SY09 */
```


© 2017 IBM Corporation

Usage & Invocation – Condition Syntax

Coming back to the syntax and the condition/“expression” part

- `WHEN ( [<WhenExprList>] ) [DO] <stmts> [END]`

The `WhenExprList` can be zero, one, or more individual `WhenExpr`:

- `<WhenExpr> = <WhenVal> <WhenCmpOp> <WhenValList>`

in which a single value is compared, as a text value, to one or more values via one of the supported comparison operators.

© 2017 IBM Corporation

Usage & Invocation – WHEN value

- A single “WHEN value” can be a predefined **variable**, or a simple, user specified **text string**:

```
<WhenVal>  =  SYSNAME | SYSPLEX | HWNAME | LPARNAME |
              VMUSERID | <text-string>
```

- The system substitutes a predefined variable with the associated “live” value (from the (E)CVT) at the time the HZSPRMxx member is parsed.
- For example, instead of SYSNAME the system will use ‘SY03’ on a system named ‘SY03’.
- Note the special case of <text-string> where you can use system symbols, which the already existing HZSPRMxx support substitutes with the symbol value before starting to parse the HZSPRMxx member:
  - For example using &SYSNAME. is equivalent to using SYSNAME
  - But, you can use many other useful system symbols; for example &SYSOSLVL. allows to code release specific conditions

© 2017 IBM Corporation

Usage & Invocation – Compare operators

The following **text comparison** operators can be used between WHEN values

- equals: ‘=‘
- not-equals: ‘<>’
- greater-than, greater-than-or-equal: ‘>’, ‘>=‘
- less-than, less-than-or-equal: ‘<‘, ‘<=‘
- List operator IN and NOTIN:
  - The IN operator compares the left-side WHEN value to the one or more right side WHEN values and evaluates to “true” if at least one of the right-side values is equal to the left-side value.
  - The NOTIN operator evaluates to “true”, if none of the right-side values is equal to the left-side value.

© 2017 IBM Corporation

Usage & Invocation – Synonyms, wildcards, restrictions

- The equals (=) and not-equals (<>) operators can be used as **synonyms** for IN and NOTIN.
- Only equals, not-equals, IN, and NOTIN are allowed to be specified with **more than one right-side** WHEN values (a “list”).
- **Wildcard characters** can be used:
  - An asterisk (*) will match any sequence of zero or more characters
  - A question mark (?) will match a single arbitrary character
  - For example: `WHEN(SYSNAME=SY1*) /* SY10, SY11... (and SY1)*/`
- Restrictions for wildcard characters:
  - Only one side of such a comparison is allowed to contain any wildcard characters
  - Only '=', '<>', 'IN', or 'NOTIN' are allow with wildcards

© 2017 IBM Corporation

Usage & Invocation – One WHEN, multiple conditions

As hinted at before, you can specify multiple conditions for one WHEN statement

- `WHEN(SYSPLEX='PLEX1'  SYSNAME='SY07')`

Logically they are **AND-ed** together.

As an alternative to nesting “single condition” WHENs

```
WHEN(SYSPLEX='PLEX1')
  DO
   WHEN(SYSNAME='SY07')
    DO ...
```


© 2017 IBM Corporation

Usage & Invocation – Whole HZSPRMxx Syntax Check

- With WHEN you now could add a single WHEN statement with a FALSE condition at the beginning of an HZSPRMxx member and let Health Checker process it, for syntax checking only:

```
WHEN(0=1)
```

- Easier: Use new option on existing “ADD parmlib member” command, the “SET HZS=” equivalent for Health Checker:

```
F HZSPROC,ADD,PARMLIB=(suffix1,...,suffixn[,{C|CHECK}])
```

Note: The `CHECK` is inside the suffix list

- With **summary** messages, per member:

```
ASA020I - "SYNTAX CHECKING IS COMPLETE FOR PARMLIB
MEMBER=memname. ERROR(S) WERE FOUND“
```

or

```
ASA021I - "SYNTAX CHECKING IS COMPLETE FOR PARMLIB
MEMBER=memname. NO ERRORS WERE FOUND"
```


© 2017 IBM Corporation

Migration & Coexistence Considerations

- Rollback to z/OS V2R2 is provided via APAR OA49807
- Without this APAR, any system with z/OS V2R2 and earlier will reject WHEN statements with standard syntax messages and will try to recover and try to parse any remaining statements within the HZSPRMxx member
  - Not recommended to rely on this “recovery” due to likely undesired effects

© 2017 IBM Corporation

Installation

- No explicit installation steps are required
- The new functions are shipped with the Health Checker portion of the z/OS base for V2R3 (BCP/HBB77B0)
- The z/OS V2R2 APAR comes with activation instructions (“restart HZSPROC…”)
- For actual exploitation:
  - Add WHEN statements to your HZSPRMxx parmlib members as appropriate
  - Possibly eliminate “redundant” copies of HZSPRMxx and IEASYSxx members which only differ in small sections, now covered by WHEN statements in a common (set of) HZSPRMxx member(s)

© 2017 IBM Corporation

Appendix

Related Publications

- “IBM Health Checker for z/OS User's Guide” (SC23-6843)
  - Guide and Reference
  - Includes all the details for any new function
  - Includes an inventory of IBM supplied health checks
- “Exploiting the Health Checker for z/OS infrastructure”
  - Health Checker “hands-on” Redpaper 4590

© 2017 IBM Corporation