



HORIZONT

SmartJCL – An introduction to IWS variables

SmartJCL and IWS Job Tailoring

About this Presentation

- This presentation gives you a short introduction to IWS job tailoring. It will also show, how HORIZONT's JCL checker "SmartJCL" handles IWS job tailoring.
- This presentation does not replace or extend any existing IWS manuals. It is an introduction, but not a reference!
- Please read the IWS manual "Planning and Scheduling the Workload" before using variables

At some slides this button can be used to jump to specific slides in this presentation.

e.g. jump to slide Agenda

Click on this symbol to jump to the overview slide

AGENDA

SmartJCL – An introduction to IWS variables

SmartJCL and IWS Job Tailoring

Overview

Variables

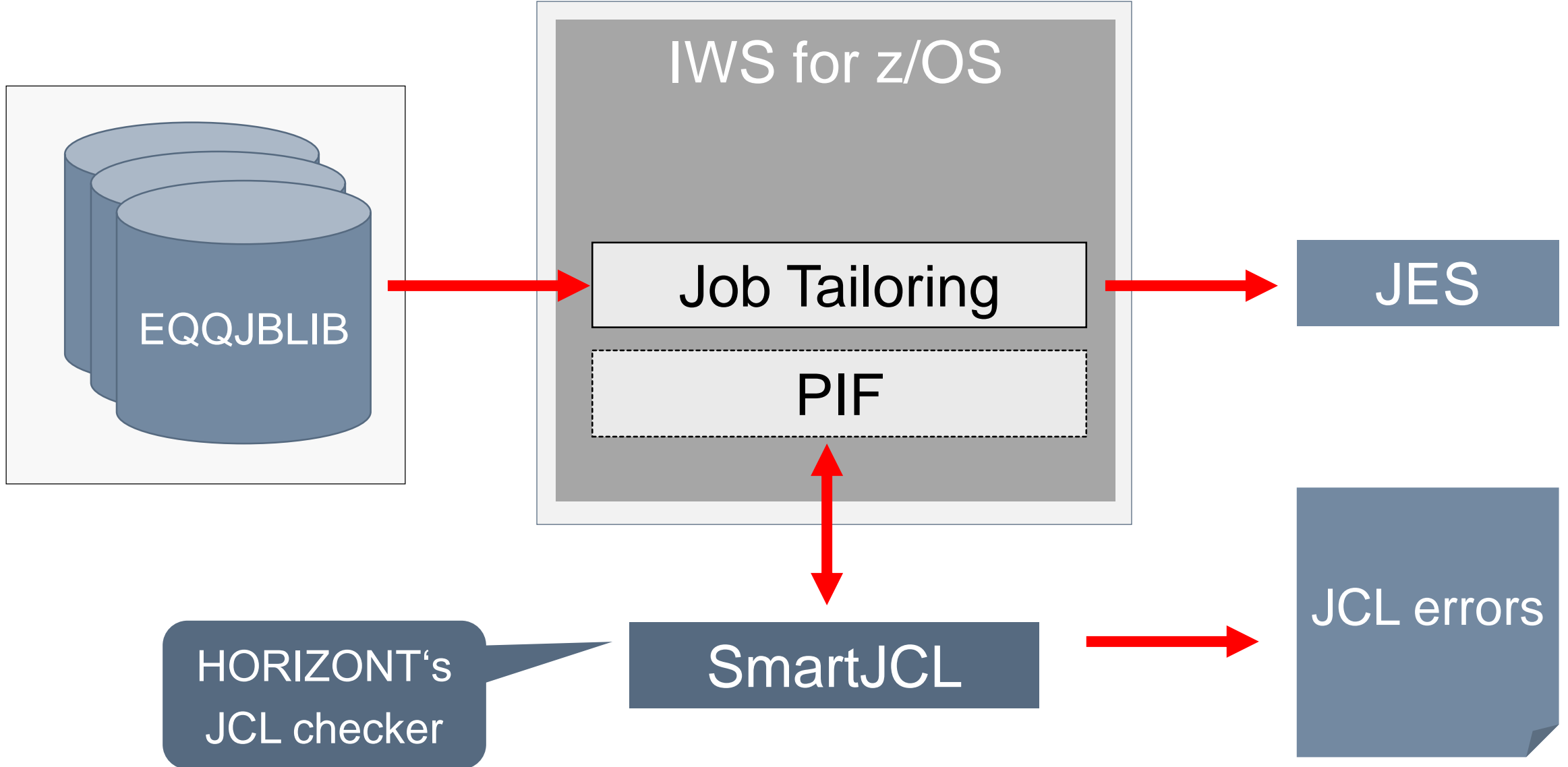
Search sequence

Directives

IWS Job Tailoring

SmartJCL

Overview



Job Tailoring

Job tailoring enables jobs to be automatically edited. Job tailoring provides:

- Variables
- Inclusion or exclusion of inline job statements
- Inclusion of job statements provided by
 - EQQJBLIB members
 - User-written exits

Variable Substitution

The values of IWS variables can be supplied:

- Automatically by IWS
- By a user-written exit
- Manually by an operator (promptable variables)

Types of Variables

IWS has different types of variables:

- Standard Variables, e.g.
 - DSN=&OADID..LIST (Application Id)
 - DSN=&OWSID..LIST (Workstation Id)
 - DSN=D&CDAY..LIST (Current day)
- User defined variables, e.g.
 - DSN=&MYHLQ..IWS.AD
 - DSN=&DATA%SET

User Variables

Typically, variables are defined by using the IWS dialogs:

```
EQQJVMAP ----- MAINTAINING OPC JCL VARIABLE TABLES
```

```
Option ==>
```

```
Select one of the following:
```

- 1 BROWSE - Browse variable tables
- 2 MODIFY - Create a new JCL variable table
or modify, copy, browse or delete
existing variable tables
- 3 PRINT - Print variable tables

See IWS manuals for more
information

The Variable Prefix

There are three different prefixes for variables. Each prefix causes IWS to perform variable substitution in a certain way.

- Ampersand &
- Percent %
- Question mark ?

The same variable can be used with different prefixes!

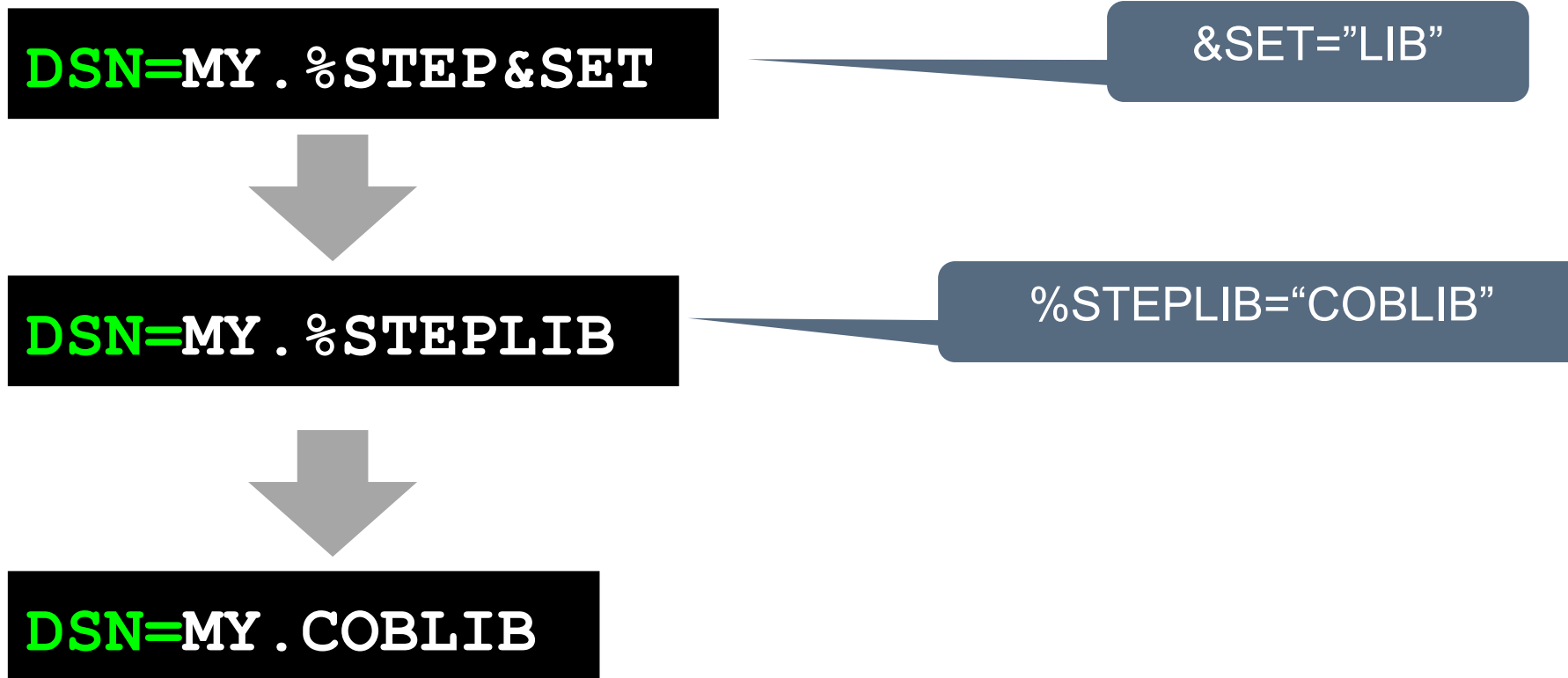
Ampersand (&)

Ampersand variables correspond to standard variables in JCL procedures and behave accordingly.

- && is not substituted (because of temp files)
- Variables which are defined “blank” are not substituted by IWS, they are treated as “normal” JCL variables

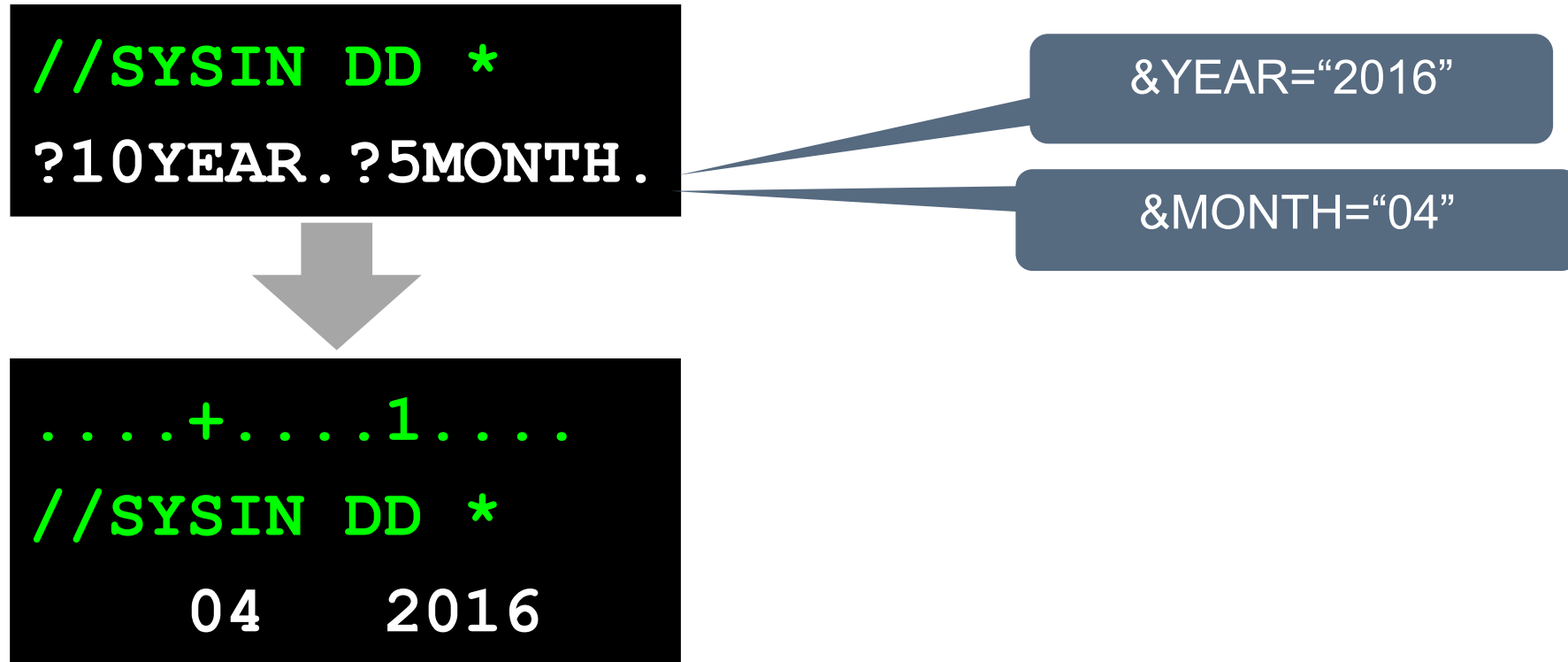
Percent (%)

Percent variables can be used to form simple variables (&) and compound variables. An example:



Question mark (?)

The position at which the value is placed can be defined in the variable table, or can be specified in the job where the variable is used. For example:



Variable Tables

Variables can be defined in different tables by using IWS dialogs, or by user written PIF-Programs:

- In tables specified for the operation
 - by using the TABLE directive in the job
 - by using the SEARCH directive in the job
- In tables specified for the application
 - by using the Modify Current Plan panel
 - by using the Long Term Plan panel
 - in the variable table associated with the run cycle
 - in the variable table associated with the period
- In the global variable table

Search sequence

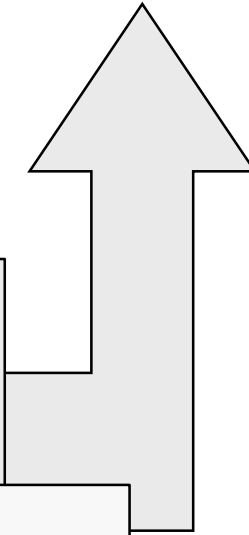
The “normal” search sequence is:

```
//SYSIN DD DISP=SHR,DSN=%MYDSN
```

JCL TABLE or SEARCH directives
&ABC=TEST . DATA (JCXB9)

Application run cycle and periods
&MYDSN=PROD . DATA (JCXB9)

Global table
&XYZ=SYS . DATA (JCXB9)



Directives

IWS uses “directives” to manage variable substitution. The directives are:

- SCAN
- SEARCH
- SETFORM
- SETVAR
- TABLE
- BEGIN and END
- FETCH

SCAN

If VARSUB (keyword in OPCOPTS) is set to SCAN, the directive informs IWS, that variable substitution should start from this line.

```
//TESTJOB JOB (ACCOUNT) ...  
//STEP1 EXEC PGM=MYPGM  
//*%OPC SCAN  
//STEPLIB DD DSN=OPC.LOAD.%LIBRARY.,  
// DISP=SHR
```

TABLE NAME

It defines the variable table that is used when attempting to assign a value to a variable.

```
//TESTJOB JOB (ACCOUNT) . . .  
//STEP1 EXEC PGM=MYPGM  
//* OPC TABLE NAME=(DAYTAB&CDAY)  
//STEPLIB DD DSN=OPC.LOAD.%LIBRARY.
```

- TABLE is often used with date variables, e.g. to use a dedicated table for every day of the week
- The variable table will be searched before the variable tables in any existing concatenation!

SEARCH NAME

It defines the variable tables that are searched when attempting to assign a value to a variable.

```
//TESTJOB JOB (ACCOUNT) ...  
//STEP1 EXEC PGM=MYPGM  
//* OPC SEARCH NAME=(TABLE1, TABLE2)  
//STEPLIB DD DSN=OPC.LOAD.%LIBRARY.
```

SEARCH NAME

Up to 16 tables, including the application and global tables, can be specified.

- The GLOBAL or NOGLOBAL keyword specifies if the global variable table will be searched.
- The APPL nor NOAPPL keyword specifies if the application variable table will be searched.
- The SEARCH statement must not contain another IWS variable (which is allowed for the TABLE statement)

SETFORM

SETFORM defines the format of dynamic variables. Dynamic variables are, e.g.

- CDATE, CTIME (Current date and time)
- ODATE, OTIME (Occurrence date and time)
- ...see IWS documentation for complete listing

```
//TESTJOB JOB (ACCOUNT) ...  
//STEP1 EXEC PGM=MYPGM  
//*%OPC SETFORM CDATE=(YYYYDDD)  
//SYSIN DD DSN=TEST.D
```

DSN=TEST.D2003058

SETVAR

It creates a temporary variable using an arithmetic expression together with supplied date variables.

```
//STEP1      EXEC PGM=MYPGM  
//*%OPC SETFORM CDATE=(YYYYDDD)  
//*%OPC SETVAR TVAR=(CDATE+1WD)  
//SYSIN      DD DSN=TEST.D%TVAR
```



```
//SYSIN      DD DSN=TEST.D2003059
```

2003058 + 1

BEGIN and END

BEGIN and END, used in pairs, denote the following, depending on the value of the ACTION keyword:

- The start and end of variable substitution
- The start and end of lines to be included or excluded from the tailored job

ACTION

BEGIN must have a matching END specifying the same ACTION. Example:

```
//STEP1      EXEC PGM=MYPGM  
//  
//STEPLIB   DD DSN=SYS.DB2.LOAD  
//*%OPC END ACTION=EXCLUDE  
//SYSIN     DD *
```



```
//STEP1      EXEC PGM=MYPGM  
//SYSIN     DD *
```


FETCH

FETCH lets you include lines from a PO member or which are supplied by an exit.

```
//STEP1      EXEC PGM=MYPGM  
//*%OPC     FETCH MEMBER=MYMEM  
//SYSIN     DD *
```



```
//STEP1      EXEC PGM=MYPGM  
//*        MYMEM  
//STEPLIB   DD DSN=SYS.DLIB  
//SYSIN     DD *
```

A comparison expression lets you specify conditions when BEGIN and FETCH directives will be honored.

```
///%OPC BEGIN ACTION=INCLUDE ,  
///%OPC COMP= ( ( &OADID . .EQ. (A1 ,A2 ,A3) ) ,  
///%OPC  
          ( &CDAY . .NE . (1 ,2) ) )
```

Logical OR

EQ,NE,GT,
GE,LT,LE

Newline:
Logical AND

READ: Include subsequent lines, if the Application Id is A1, A2 or A3
and it is not Monday or Tuesday

PHASE

Specifies whether the BEGIN/END pair should take effect during the setup or submit phase of the operation

```
// *%OPC BEGIN ACTION=EXCLUDE , PHASE=SETUP
```

- Possible values: SETUP or SUBMIT
- Default is SUBMIT

IWS Job Tailoring - Summary

IWS job tailoring provides:

- Predefined variables for date, time etc.
- User specific variables
- &, % and ? variables
- Directives to manage variable substitution, e.g. SCAN, SEARCH, TABLE
- Inclusion or exclusion of job statements

IWS job tailoring and JCL checkers

The previous slides shows you how Job Tailoring can be used to make your JCL more flexible. However, this has an impact on JCL checkers:

- The JCL checker must be able to correctly handle IWS variable and directives
- Due to date related variables, the JCL checker should be able to “simulate” IWS job tailoring for any day in the future.

SmartJCL – JCL Validation

One major benefit of SmartJCL is its integrated IWS Interface, but it also performs the basic tasks of a JCL checker:

- SmartJCL checks syntax (parameters, quotes, etc.)
- SmartJCL checks referbacks
- SmartJCL checks JCL overrides
- SmartJCL checks catalog actions
- SmartJCL checks GDG processing
- SmartJCL checks if program, dataset or proc exist

SmartJCL – Interface to IWS

One of the benefits of SmartJCL is the integrated IWS Interface. It enables you to:

- Simulate IWS job tailoring
- Check JCL in production sequence to simulate catalog actions
- Select the jobs which should be checked by Application Id, Owner Id, Input Arrival etc.

SmartJCL – Variables, Example (1)

```
File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          P390P.TEST.CNTL(SIMJCL01) - 01.04
COMMAND ==>
*****
***** Top of Data *****
000001 //SIMJCL01 JOB                ,M IWS Variables ... :SYSUID
000002 //*%OPC SCAN
000003 //*****
000004 //DELETE EXEC PGM=IEFBR14
000005 //DD1          DD DSN=P390A.&OJOBNAM IWS Variables ... IR
000006 //*****
000007 //ALLOC EXEC PGM=IEFBR14,COND=(4,LT)
000008 //DD1          DD DSN=P390A.&OJOBNAME..N&OOPNO,
000009 //              DISP=(,CATLG),SPAC IWS Variables ... ),
000011 //*****
000012 //WRITER EXEC PGM=REPORT01,COND=(1,LT)
000013 //OUT1          DD DISP=SHR,DSN=P390A.&OJOBNAME..N&OOPNO
000014 //IN1           DD DISP=SHR,DSN=P390A. IWS Variables ...
*****
***** Bottom of Data *****
```


SmartJCL – Variables, Example (2)

```
File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT      P390P.TEST.CNTL(SIMJCL01) - 01.04
COMMAND ==>
*****  ***** Top of Data *****
000001 //SIMJCL01 JOB                ,MSGCLASS=T,NOTIFY=&SYSUID
000002 //*%OPC SCAN
000003 //*****
000004 //DELETE EXEC PGM=IEFBR14
000005 //DD1 DD DSN=P390A.&OJOBNAME..N&OOPNO,DISP=SHR
000007 //*****
000008 //ALLOC EXEC PGM=IEFBR14,COND=(4,LT)
000009 //DD1 DD DSN=P390A.&OJOBNAME..N&OOPNO,
.HAAA // DISP=(,CATLG),SPACE=(CYL,(2,5),RLSE)
==MSG> JCK0316E - INVALID DATA SET NAME FORMAT
000010 //*****
000011 //WRITER EXEC PGM=REPORT01,COND=(4,LT)
.HAAI //OUT1 DD DISP=SHR,DSN=P390A.&OJOBNAME..N&OOPNO
```

... must be substituted before JCL validation.

==MSG> JCK0316E - INVALID DATA SET NAME FORMAT

SmartJCL – Variables, Example (3)

```
File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          P390P.TEST.CNTL(SIMJCL02) - 01.00
COMMAND ==>
***** ***** Top of Data *****
000001 //S                                $YSUID
=NOTE= --S                                $90G
000002 //*
000003 //*                               *****
000004 //D
000005 //D
000007 //*                               *****
000008 //Z
000009 //DD1          DD DSN=P390A.SIMJCL0
000010 //              DISP=(,CATLG),SPA
000011 //*****
000012 //WRITER EXEC PGM=REPORT01,CONI
000013 //OUT1          DD DISP=SHR,DSN=P390
```

Jobname found in IWS-CP

Additional search in IWS-AD?

Yes or No? Y

SmartJCL has an integrated IWS dialog: First the AD and the CP are scanned for the job

SmartJCL – Variables, Example (4)

```
File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          P390P.TEST.CNTL(SIMJCL02) - 01.00
C
*----- OPC/SimJCL ----- Row 1 of 35 *
* Command ==> _____ Scroll ==> PAGE *
*
* Select line with S to start Simulation *
*
* Simulation Date      ==> 20011231 T *
* Input Arrival Date ==> _____ Time ==> _____ *
*
* Application ID      Jobname  Input Arrival  OpNo  WsId *
* HELMUTTIME         P390K2   20020422  0800  020  CPU1  CP
* - HELMUTTIME         P390K2   20020422  0900  020  CPU1  CP
* - HELMUTTIME         P390K2   20020422  1000  020  CPU1  CP
* - HELMUTTIME         P390K2   20020422  1100  020  CPU1  CP
* - HELMUTTIME1        P390K2   20020422  0700  020  CPU1  CP
* -
```

... the User has to select the desired Application ...

SmartJCL – Variables, Example (5)

```
File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          P390P.TEST.CNTL(SIMJCL02) - 01.00
COMMAND ==>
***** Top of Data *****
000001 //SIMJCL01 JOB 'HORIZONT' MSCCLASS=T NOTIFY=CSYSUID
=NOTE= --SIMJCL01 JOB 'HORIZONT'
000002 //*%OPC SCAN
000003 /*******
000004 //DELETE EXEC PGM=IEFBR14
000005 //DD1 DD DSN=P390A.SIMJCL01.N010,DISP=SHR
000007 /*******
000008 //ALLOC EXEC PGM=IEFBR14,COND=(4,LT)
000009 //DD1 DD DSN=P390A.SIMJCL01.N010,
000010 // DISP=(,CATLG),SPACE=(CYL,(2,5),RLSE),
000011 /*******
000012 //WRITER EXEC PGM=REPORT01,COND=(4,LT)
000013 //OUT1 DD DISP=SHR,DSN=P390A.SIMJCL01.N010
```

... and finally all Variables are substituted!

SmartJCL – ISPF Dialog

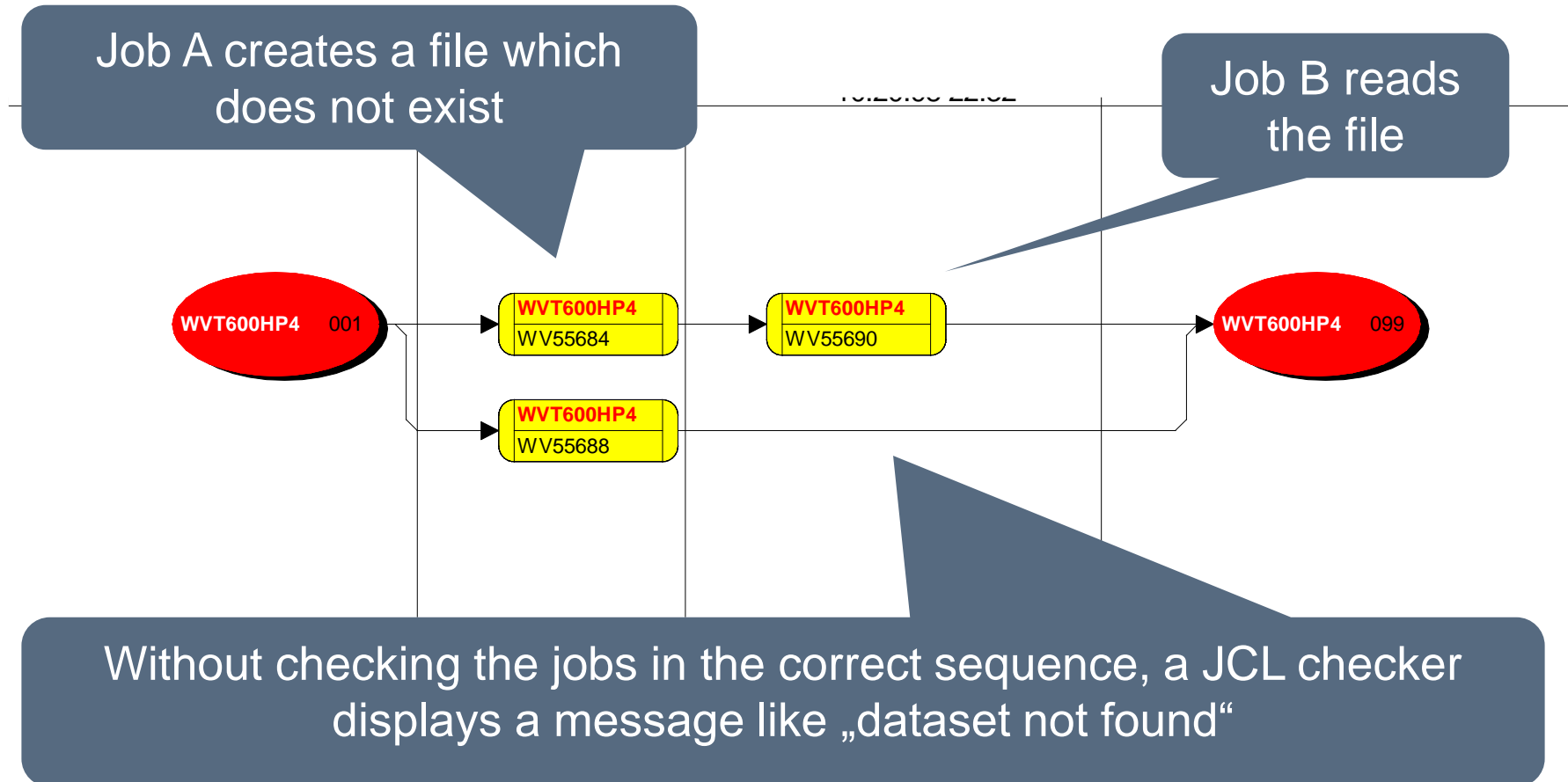
```
----- OPC/SimJCL Startpanel -----  
Command ===>  
  
Select one of the following options and press  
1  AD      - Select jobs from Application Description  
2  CP      - Select jobs from Current Plan  
3  FORECAST - Select jobs from LTP or future planning  
  
4  SIMULATE - Start JCL simulation  
  
5  CHECK   - Start JCL check based on simulation  
6  CHECK   - Start JCL check without simulation  
  
Note:      Options 5 and 6 are only available  
           if a JCL checker is installed
```

Select jobs from AD, CP and LTP

Check JCL for any day in the future

SmartJCL – Checking jobs in sequence

The integrated IWS interface enables SmartJCL to check jobs in a correct sequence:



SmartJCL - Summary

SmartJCL is a JCL checker with an integrated IWS interface, some functions are:

- Standard JCL validation, such as syntax checks, catalog actions etc.
- Simulation of IWS job tailoring
- Check JCL in production sequence to simulate catalog actions
- Select jobs which should be checked by Application Id, Owner Id, Input Arrival etc.

Thanks for your attention! Do you have any questions?

The logo for 'HORIZONT' features the word in a bold, sans-serif font. The letter 'O' is highlighted in red, while the other letters are in dark grey. The background is light grey with a red curved shape on the left and several faint geometric shapes (squares and a triangle) scattered around the text.

HORIZONT

Please feel free to visit us in Munich or send an email to
info@horizont-it.com