

## Overview

CBROAMVT is an OAM initialization simulation tool that can be used to validate a CBROAMxx parmlib member with the purpose of detecting errors before an OAM initialization or restart is issued. The overall design of the tool is made to mimic the OAM initialization process as closely as possible. The tool can be executed from TSO or batch with the user providing input for the CBROAMxx parmlib member, output dataset, and (if applicable) ONLYIF parameters through either text prompts from the tool itself or directly as input parameters. This tool will check for any syntax errors and invalid values for each statement within the CBROAMxx parmlib member and, if any errors or warnings are found, the tool will output a report to the given output dataset for review by the user. The error messages are formatted to match those given by OAM initialization but include additional messages with information to help the user determine how to fix the errors/warnings (such as indicating if a keyword is too long, the wrong type, or an invalid value for the given parameter).

## Invocation

### 1. Via TSO

- a. Go to the TSO command line and issue 'CBROAMVT'
- b. Follow prompts to enter the **CBROAMxx parmlib member** and the name of an **output dataset** that will contain the results of the check. The dataset names must be fully qualified, and the output dataset must refer to a new, unallocated dataset (the tool will handle the allocation of the output dataset).
- c. If any information provided is incorrect (refer to table below) the user will be re-prompted for the information.
- d. If ONLYIF statements are present in the CBROAMxx parmlib member follow the prompts to choose how to handle the processing of ONLYIF paths (see fig. 2 and fig. 5).
- e. Input parameters

<b>cbroam</b>	The fully qualified dataset name of the CBROAMxx parmlib member.
<b>outds</b>	The fully qualified dataset name of the output dataset, must refer to a new, unallocated dataset.
<b>crosscheck</b>	"Y" or "N", indicates if crosschecking should occur if ONLYIF statements are found.
<b>sysname</b>	SYSNAME parameter for ONLYIF statements. Cannot be blank.
<b>db2id</b>	DB2ID parameter for ONLYIF statements. Cannot be blank.
<b>oamvar</b>	OAMVAR parameter for ONLYIF statements. Cannot be blank.

## 2. Via batch

### a. Sample JCL

```
//CBROAMVT JOB , 'user ', CLASS=A,
//          MSGCLASS=H, NOTIFY=&SYSUID
// *
//STEP1     EXEC PGM=IKJEFT01
//SYSPROC   DD DISP=SHR, DSN=SYS1.CLIST
//SYSTSPRT  DD SYSOUT=*
//SYSTSIN   DD *
           CBROAMVT cbroam outds crosscheck sysname db2id oamvar
/*
```

### b. Input parameters

<b>cbroam</b>	Required, the fully qualified dataset name of the CBROAMxx parmlib member.
<b>outds</b>	Required, the fully qualified dataset name of the output dataset, must refer to a new, unallocated dataset.
<b>crosscheck</b>	Optional, "Y" or "N", indicates if crosschecking should occur if ONLYIF statements are found. Default is "N".
<b>sysname</b>	Optional, SYSNAME parameter for ONLYIF statements, only applies if ONLYIF statements are found in the CBROAMxx parmlib member and crosschecking is enabled.
<b>db2id</b>	Optional, DB2ID parameter for ONLYIF statements, only applies if ONLYIF statements are found in the CBROAMxx parmlib member and crosschecking is enabled.
<b>oamvar</b>	Optional, OAMVAR parameter for ONLYIF statements, only applies if ONLYIF statements are found in the CBROAMxx parmlib member and crosschecking is enabled.

### c. If you receive a return code 8, it indicates an error with your input parameters.

## Behaviors and Limitations

1. If any errors were found in the CBROAMxx parmlib member they will be listed in the output dataset under its respective set statement, in order of occurrence (see fig. 3b). If no errors were found, the output dataset will not be allocated (see fig. 2)
2. Error and warning statements will resemble what would be output by OAM initialization. In some cases, extra information is provided to explain the reason for the error (see fig. 3b).
  - a. If a syntax error is due to an incorrect length or datatype, or what the expected value for a parameter is.
  - b. If a keyword/value has multiple errors associated with it (such as both an invalid value and a missing parenthesis) error messages for both will be issued.
3. Storage group behavior
  - a. The current version of the tool has no capability of validating storage groups, it assumes that all storage groups provided exist.

- b. Storage group types start as undefined when a storage group is first referenced. Once a parameter is referenced that applies to a specific storage group type, the storage group will have its type set to match. From that point on, that storage group will be referenced as that type.
  - i. For example, here SGBACK1 will be set as a BACKUP storage group type. This will cause the second line to produce an error because FIRSTBACKUPSTORAGEGROUP expects to be applied to an OBJECT storage group.

```
SETOSMC FIRSTBACKUPGROUP(SGBACK1)  
STORAGEGROUP(SGBACK1 FIRSTBACKUPGROUP(SGBACK2))
```

- 4. Crosschecking behavior
  - a. Crosschecking refers to checking a specific combination of ONLYIF parameters and only processing the set statements associated with that configuration.
  - b. Crosschecking is used when there is at least one ONLYIF statement in the CBROAMxx parmlib member that contains a specific value for any one of the three ONLYIF parameters: SYSNAME, DB2ID, and OAMVAR.
  - c. Only parameters found in the ONLYIF statements will be asked for (see fig. 2). These values cannot be null.
  - d. When crosschecking, only statements prior to the first ONLYIF statement, ONLYIF statements, and statements inside corresponding ONLYIF statements are checked for errors (see fig. 3b).
  - e. If crosschecking is not specified, when an ONLYIF statement is encountered all parameters are reset except those set before the first ONLYIF statement and final parameter checks are run (see fig. 4a and 4b).
- 5. This tool will not check that expiration dates are at or after the current date, only for the syntax and validity of the date.
- 6. This tool cannot verify if a Dataclass, Tapeunitname, or Tapecapacity name exist. It will still check the syntax of these parameters.
- 7. This tool does support system symbols. All symbols used must be defined on the system the tool is being run on. If a system symbol is undefined, the behavior matches OAM initialization in the fact that it is treated as an invalid keyword (see fig. 6).

## Example usage

**Figure 1:** Example CBROAMxx member. ONLYIF statement referring to DB2ID “DBC0” has statements that contain errors (circled in red).

```
SETOAM
  MAXTAPERETRIEVETASKS(20) MAXTAPESTORETASKS(15)
  MAXRECYCLETASKS(12)
  TAPEEXPIRATION(2023/&JDAY)
} Global Definitions

ONLYIF SYSNAME(PC8) DB2ID(NONE)
SETTLIB ENTRYIGNOREMSGTYPE(SUPPRESS)

ONLYIF SYSNAME(PC8) DB2ID(*ALL*)
SETOSMC
  STORAGEGROUP(CL1PL2P8 BACKUPTIER(CLOUD))
  STORAGEGROUP(CL2PL2P8 BACKUPTIER(CLOUD))
  STORAGEGROUP(FS1PL2P8 BACKUPTIER(FS))
  STORAGEGROUP(FS2PL2P8 BACKUPTIER(FS))

ONLYIF SYSNAME(PC8) DB2ID(DAC0)
SETOSMC
  FIRSTBACKUPGROUP(FS1PL2P8)
  SECONDBACKUPGROUP(CL2PL2P8)
SETOAM
  STORAGEGROUP(GROUP00
    TAPEUNITNAME(3490) L2TAPEUNITNAME(3480)
    MAXTAPERETRIEVETASKS(10) MAXTAPESTORETASKS(10)
    DATACLASS(DCBAR144) L2DATACLASS(DCBAR144)
    TAPEDRIVESTARTUP(8000) DEMOUNTWAITTIME(0045)
    TAPEPERCENTFULL(100) SGMAXRECYCLETASKS(8))

ONLYIF SYSNAME(PC8) DB2ID(DBC0)
SETOSMC
  FIRSTBACKUPGROUP(CL1PL2P8)
  SECONDBACKUPGROUP(FS2PL2P8)
SETOAM
  STORAGEGROUP(GROUP02
    TAPEUNITNAME(3490) L2TAPEUNITNAME(3480)
    MAXTAPERETRIEVETASKS(9999) MAXTAPESTORETASKS(10)
    DATACLASS(&ERRSYM) L2DATACLASS(DCBAR144)
    TAPEDRIVESTARTUP(8000) DEMOUNTWAITTIME(0045)
    TAPEPERCENTFULL(ERR) SGMAXRECYCLETASKS(15))
```

Error: This system symbol does not exist on the test system.

Error: The keyword TAPEPERCENTFULL only takes numeric data.

Error: The keyword MAXTAPERETRIEVETASKS has a maximum length of 3 for its value.

Error: The value for SGMAXRECYCLETASKS is greater than the previously defined global MAXRECYCLETASKS.

**Figure 2:** Example CBROAMVT call with crosschecking and no errors. CBROAMxx member used is shown in figure 1. Because no errors are found, no output data set is allocated. Note that even though there are errors in the fourth ONLYIF statement they aren't caught because the provided DB2ID is not DBC0.

```

CBROAMVT v1.0: CBROAMxx verification tool
-----
If the given CBROAMxx dataset is found to have errors
then an output data set will be created that contains
the error messages associated with the parsing errors.
This is a simulation of CBROAMxx processing. Initialization
will not occur.
-----
Enter dataset name of CBROAMxx member to be validated:
CBROAMVT.EXAMPLE(ONLYIFEX)
User input CBROAMxx dataset: CBROAMVT.EXAMPLE(ONLYIFEX)
Enter output data set name for validation result:
CBROAMVT.EXAMPLE.OUT
User output dataset name: CBROAMVT.EXAMPLE.OUT
Explicit ONLYIF Statements found. Would you like to do cross-checking? (Y/N)
Y
Enter SYSNAME:
PC8
User input sysname: PC8
Enter DB2ID:
DAC0
User input db2id: DAC0
CBROAMVT tool running...
CBROAMVT tool has ended
No errors found with CBROAM member. Output dataset not allocated.
***

```

Prompts to determine which ONLYIF statements to process.  
OAMVAR is not requested because as seen in fig. 1, there are no  
OAMVAR parameters defined.  
With this configuration only global statements and statements  
under the second and third ONLYIF statements will be read.

**Figure 3a:** Example CBROAMVT call with crosschecking and errors. CBROAMxx member used is shown in figure 1.

```
CBROAMVT v1.0: CBROAMxx verification tool
-----
If the given CBROAMxx dataset is found to have errors
then an output data set will be created that contains
the error messages associated with the parsing errors.
This is a simulation of CBROAMxx processing. Initialization
will not occur.
-----
Enter dataset name of CBROAMxx member to be validated:
CBROAMVT.EXAMPLE(ONLYIFEX)
User input CBROAMxx dataset: CBROAMVT.EXAMPLE(ONLYIFEX)
Enter output data set name for validation result:
CBROAMVT.EXAMPLE.OUT
User output dataset name: CBROAMVT.EXAMPLE.OUT
Explicit ONLYIF Statements found. Would you like to do cross-checking? (Y/N)
Y
Enter SYSNAME:
PC8
User input sysname: PC8
Enter DB2ID:
DBC0
User input db2id: DBC0
CBROAMVT tool running...
CBROAMVT tool has ended
Errors found with CBROAM member. Information written to CBROAMVT.EXAMPLE.OUT.
***
```

Prompts to determine which ONLIF statements to process.  
OAMVAR is not requested because as seen in fig. 1, there are no  
OAMVAR parameters defined.  
With this configuration only global statements and statements  
under the second and fourth ONLYIF statements will be read.

**Figure 3b:** Output dataset produced by figure 3a. Extra information provided by CBROAMVT is circled in red.

```

----- SETOAM -----
No errors found

=====
----- ONLYIF SYSNAME(PC8) DB2ID(NONE) -----
No errors found

=====
----- ONLYIF SYSNAME(PC8) DB2ID(*ALL*) -----
No errors found

----- SETOSMC -----
No errors found

=====
----- ONLYIF SYSNAME(PC8) DB2ID(DBC0) -----
No errors found

----- SETOSMC -----
No errors found

----- SETOAM -----
CBR0303I Data for keyword MAXTAPERETRIEVETASKS in a SETOAM statement is invalid
- 9999.
Data is invalid length.
CBR0303I Data for keyword DATACLASS in a SETOAM statement is invalid - &ERRSYM.
First character of data is invalid type.
CBR0303I Data for keyword TAPEPERCENTFULL in a SETOAM statement is invalid -
ERR.
Data is invalid type.

----- FINAL PARAMETER CHECK -----
CBR0342I STORAGEGROUP GROUP02 SGMAXRECYCLETASKS value (15) is greater than
SETOAM MAXRECYCLETASKS value (12).

```

This SETOAM statement is not associated to any ONLYIF statements so it is always checked.

This ONLYIF refers to the first ONLYIF statement in the CBROAMXX member seen in fig. 1. Because the DB2ID provided is not NONE the following SETLIB statement is not checked.

This ONLYIF refers to the second ONLYIF statement in the CBROAMXX member seen in fig. 1. Because the SYSNAME and DB2ID provided match the ones in the statement the following SETOSMC statement is checked.

This system symbol is processed as is because it does not exist on the testing system.



*Figure 4a: Example CBROAMVT call where user indicates no crosschecking.*

```
CBROAMVT v1.0: CBROAMxx verification tool
-----
If the given CBROAMxx dataset is found to have errors
then an output data set will be created that contains
the error messages associated with the parsing errors.
This is a simulation of CBROAMxx processing. Initialization
will not occur.
-----
Enter dataset name of CBROAMxx member to be validated:
CBROAMVT.EXAMPLE(ONLYIFEX)
User input CBROAMxx dataset: CBROAMVT.EXAMPLE(ONLYIFEX)
Enter output data set name for validation result:
CBROAMVT.EXAMPLE.OUT
User output dataset name: CBROAMVT.EXAMPLE.OUT
Explicit ONLYIF Statements found. Would you like to do cross-checking? (Y/N)
N
Only syntax checking will be done.
CBROAMVT tool running...
CBROAMVT tool has ended
Errors found with CBROAM member. Information written to CBROAMVT.EXAMPLE.OUT.
***
```



Figure 4b: Output dataset provided by figure 4a.

```
----- SETOAM -----
No errors found
=====
----- ONLYIF SYSNAME(PC8) DB2ID(NONE) -----
No errors found
----- SETTLIB -----
No errors found
-----FINAL PARAMETER CHECK-----
No errors found
=====
----- ONLYIF SYSNAME(PC8) DB2ID(*ALL*) -----
No errors found
----- SETOSMC -----
No errors found
-----FINAL PARAMETER CHECK-----
No errors found
=====
----- ONLYIF SYSNAME(PC8) DB2ID(DAC0) -----
No errors found
----- SETOSMC -----
No errors found
----- SETOAM -----
No errors found
-----FINAL PARAMETER CHECK-----
No errors found
=====
----- ONLYIF SYSNAME(PC8) DB2ID(DBC0) -----
No errors found
----- SETOSMC -----
No errors found
----- SETOAM -----
CBR0303I Data for keyword MAXTAPERRETRIEVETASKS in a SETOAM statement is invalid
- 9999.
Data is invalid length.
CBR0303I Data for keyword DATACLASS in a SETOAM statement is invalid - &ERRSYM.
First character of data is invalid type.
CBR0303I Data for keyword TAPEPERCENTFULL in a SETOAM statement is invalid -
ERR.
Data is invalid type.
-----FINAL PARAMETER CHECK-----
CBR0342I STORAGEGROUP GROUP02 SGMAXRECYCLETASKS value (15) is greater than
SETOAM MAXRECYCLETASKS value (12).
```

Values set before any ONLYIF statements are kept during the checking of all other statements in the CBROAMxx member.

Before a new ONLYIF statement is read, a final parameter check is issued and all values set in the previous ONLYIF block are cleared.

**Figure 5:** Example CBROAMVT call with no crosschecking. CBROAMxx member contains no ONLYIF statements so no prompts are given.

```
CBROAMVT v1.0: CBROAMxx verification tool
-----
If the given CBROAMxx dataset is found to have errors
then an output data set will be created that contains
the error messages associated with the parsing errors.
This is a simulation of CBROAMxx processing. Initialization
will not occur.
-----
Enter dataset name of CBROAMxx member to be validated:
CBROAMVT.EXAMPLE(NOERROR)
User input CBROAMxx dataset: CBROAMVT.EXAMPLE(NOERROR)
Enter output data set name for validation result:
CBROAMVT.EXAMPLE.OUT
User output dataset name: CBROAMVT.EXAMPLE.OUT
CBROAMVT tool running...
CBROAMVT tool has ended
No errors found with CBROAM member. Out dataset not allocated.
***
```