

IBM z/OS Change Tracker
Version 2.5

Guide and Reference



Note

Before using this information and the product it supports, read the information in [“Notices” on page 209](#).

This edition applies to Version 2 Release 5 of z/OS® (5650-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2023-03-28

© **Copyright International Business Machines Corporation 2022, 2023.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

© **Teracloud ApS 2022, 2023.**

Contents

Figures.....	vii
Tables.....	xi
About this information.....	xiii
Who should use this information.....	xiii
z/OS information.....	xiii
How to send your comments to IBM.....	xv
If you have a technical problem.....	xv
Summary of changes.....	xvii
Summary of changes for IBM z/OS Change Tracker for z/OS Version 2 Release 5 (V2R5).....	xvii
Chapter 1. Overview.....	1
z/OS Change Tracker functionality.....	2
Use cases.....	4
Change tracking.....	4
Data set protection.....	7
Tokenization.....	12
Tokenization of z/OS UNIX System Services directories.....	16
Chapter 2. Configuration and implementation.....	19
90-day trial.....	19
Security checklist.....	20
Configuring z/OS Change Tracker.....	20
z/OSMF plug-in considerations.....	22
z/OS Change Tracker maintenance and housekeeping tasks.....	22
Chapter 3. Using z/OS Change Tracker with ISPF.....	25
z/OS Change Tracker options.....	26
Administration.....	27
Check out.....	29
Check in.....	30
Protect a resource.....	31
Modify protection.....	34
Remove protection.....	35
Backup.....	36
Delete or archive backups.....	37
Checked out resources.....	37
Show archive reports.....	39
Show logs.....	40
Recover archived backups.....	41
Prepare tokens.....	43
Identify MODS.....	44
Audit.....	45
Recover	46
Disaster recovery.....	50
Tokenize your Software.....	50

Report differences.....	51
Utilities.....	52
SuperC member compare.....	52
Macro compare.....	53
SuperC compare.....	56
Group compare.....	58
Reconcile two libraries.....	61
Reconcile groups of libraries.....	63
Tokenize directory files.....	65
Report changed files in a directory	67
Macro compare files.....	69
List a token file.....	70

Chapter 4. Using z/OS Change Tracker batch commands..... 73

How to read the syntax diagrams.....	73
ARCHIVEBACKUP command.....	75
ARCHIVERECOVER command.....	76
ARCHIVERESOURCE command.....	77
AUTOARC command.....	78
BEGINDATE, ENDDATE commands.....	79
CHECKIN command.....	80
CHECKOUT command.....	81
COMPARE command.....	82
COPYDIFF command.....	84
DATE command.....	84
DD1SHOW, DD2SHOW commands.....	85
DEACTIVATESTC command.....	86
DELETEBACKUP command.....	87
DSN1, DSN2 commands.....	88
ENDCHECK command.....	89
EXCLUDE command.....	90
EXPLICIT command.....	91
GROUPCOMPARE command.....	92
INCLUDE command.....	93
LIST command.....	94
LOADDD command.....	94
LOGCOMMENT command.....	95
MEMBERBACKUP command.....	96
MEMBERCOMMENT command.....	97
MEMBERRECOVER command.....	97
MEMLIST command.....	99
OBJECT command.....	100
OPTION command.....	101
PATHCOMPARE command.....	103
PATTERN command.....	105
PROTECT command.....	105
REACTIVATESTC command.....	109
REJECT command.....	110
REMOTECOMPARE command.....	111
REMOTEGROUPCOMP command.....	112
REMOTEOBJECT1, REMOTEOBJECT2 commands.....	113
REMOVECHECKOUT command.....	114
REPROTECT command.....	114
RESYNC command.....	116
SAVEDD command.....	117
SHOW=ACTIONS command.....	117
SHOW=ARCHIVES command.....	119

SHOW=BACKUP command.....	119
SHOW=CHECKOUT command.....	120
SHOW=LIST command.....	120
SHOW=LOG command.....	121
SHOW=PROTECTIONS command.....	122
SHOW=TOKENS command.....	122
SHOW=UNIVERSE command.....	123
SHOW=USSENV command.....	124
STRINGREPLACE command.....	124
SURVEILLANCE command.....	125
SYSUT1, SYSUT2 command.....	127
TESTRUN command.....	127
TOKEN command.....	128
TRIGGERREFRESH command.....	132
UNLOADTOKENS command.....	133
UNPROTECT command.....	134
USSEXCLUDE command.....	135
USSPATHREPL command.....	136
USSENV command.....	137
USSREJECT command.....	139
USSREPORT command.....	140
USSTOKENIZE command.....	141
VERIFY command.....	142
VOL1, VOL2 commands.....	143
VOLREP command.....	144
WRITETOFILE command.....	144
Chapter 5. Sample jobs.....	147
Chapter 6. z/OS Change Tracker messages.....	153
Appendix A. Accessibility.....	207
Notices.....	209
Terms and conditions for product documentation.....	210
IBM Online Privacy Statement.....	211
Policy for unsupported hardware.....	211
Minimum supported hardware.....	211
Trademarks.....	212

Figures

1. Typical z/OS Change Tracker sysplex configuration.....	21
2. Primary option menu.....	25
3. Options and Parameters.....	26
4. Administration menu.....	27
5. First Check Out panel.....	29
6. Second Check Out panel.....	30
7. Check in a member (1 of 2).....	30
8. Check in a member (2 of 2).....	31
9. Protect a resource (1 of 3).....	32
10. Protect a resource (2 of 3).....	32
11. Protect a resource (3 of 3).....	32
12. Protecting an uncataloged data set.....	34
13. Modify resource protection (LOCK Status).....	35
14. Remove resource protection.....	35
15. Backup members - protected resource list.....	36
16. Second Backup panel.....	36
17. Delete or archive backups (1 of 2).....	37
18. Delete or archive backups (2 of 2).....	37
19. List of backups for the member.....	37
20. Report on resources (1 of 2).....	38
21. Report on resources (2 of 2).....	38
22. Example of Checked-Out-Components report.....	39
23. Show archived backups (1 of 3).....	39

24. Show archived backups (2 of 3).....	39
25. Show archived backups (3 of 3).....	39
26. Show archived actions (1 of 2).....	40
27. Show archived actions (2 of 2).....	40
28. Show administrator logs	40
29. Example of the List of TRACE activities report.....	41
30. Recover archived backups (1 of 3): Archived Resources.....	41
31. Recover archived backups (2 of 3): Recover or View Backups.....	42
32. Recover archived backups (3 of 3).....	42
33. Prepare tokens (2 of 3).....	43
34. Prepare tokens (3 of 3).....	43
35. Tokenization JCL.....	44
36. Identify MODS selection.....	44
37. Identify MODS (1 of 2).....	44
38. Identify MODS (2 of 2).....	45
39. Audit.....	46
40. Example audit results.....	46
41. Recover (1 of 3).....	46
42. Recover or view member versions online (2 of 3).....	47
43. Recover - backup versions of member (3 of 3).....	47
44. Comparing pairs of versions online.....	48
45. Chronological change details (1 of 3).....	49
46. Chronological change details (2 of 3).....	49
47. Chronological change details (3 of 3).....	50
48. Disaster recovery functions.....	50

49. Prepare tokens.....	50
50. Tokenize production (1 of 3).....	51
51. Tokenize (2 of 3).....	51
52. Tokenize (3 of 3).....	51
53. Report differences.....	51
54. Utilities menu.....	52
55. SuperC of protected members.....	52
56. Select a resource for SuperC.....	52
57. Select members for SuperC comparison.....	53
58. Macro compare new data set.....	54
59. Macro compare old data set.....	54
60. Macro compare statistics.....	54
61. Macro compare result.....	55
62. SuperC Compare Report.....	55
63. Difference report when “Consider Directory” is requested.....	56
64. SuperC compare new data set.....	57
65. SuperC compare old data set.....	57
66. SuperC compare result.....	57
67. Group compare - defining the new set (1 of 2).....	58
68. Group compare - resolved libraries for the new set (2 of 2).....	59
69. Group compare - defining the old set (1 of 2).....	59
70. Group compare - resolved libraries for the old set (2 of 2).....	59
71. Group compare JCL.....	60
72. Example group compare report (1 of 2)	60
73. Example group compare report (2 of 2).....	61

74. Reconcile two libraries (specifying the from data set).....	62
75. Reconcile two libraries (specifying the To (Old) data set).....	62
76. Sample reconcile JCL.....	63
77. Reconcile groups of libraries - specifying the From data sets.....	63
78. Reconcile groups of libraries - the resolved From data sets.....	64
79. Reconcile groups of libraries - specifying the To data sets.....	64
80. Reconcile groups of libraries - resolved To list.....	65
81. Group reconciliation JCL	65
82. Tokenize files in a path (1 of 2).....	66
83. Tokenize files in a path (2 of 2).....	66
84. JCL to tokenize files in a z/OS UNIX path.....	67
85. Report changed files in a directory.....	67
86. JCL to report changed files over time.....	68
87. Report of changed files (1 of 2).....	68
88. Report of changed files (2 of 2).....	68
89. Macro compare files in two Paths.....	69
90. JCL to compare two paths.....	69
91. Path compare report	70
92. List a token file.....	70
93. JCL to list content of a Token File.....	70
94. Example contents of a token file	71

Tables

1. Primary menu options.....	25
2. Description of input fields in Options and Parameters panels.....	26
3. z/OS Change Tracker administration menu options.....	27
4. Description of output fields.....	30
5. Description of fields.....	31
6. Description of output fields.....	38
7. Description of input fields (CYGUCMP1).....	56
8. Example of data sets paired for Group Compare.....	58
9. Description of input parameters for reconciliation (CYGUCOP1).....	62
10. Data sets paired for group reconciliation.....	63
11. Description of input fields (CYGOCHG).....	69
12. Description of input fields (CYGOLIST).....	71
13. Summary of jobs in the samples library.....	147
14. CYGA* jobs.....	147
15. CYGB* jobs.....	148
16. CYGC* jobs.....	148
17. CYGD* jobs.....	148
18. CYGH* jobs.....	149
19. CYGK* jobs.....	149
20. CYGR* jobs.....	149
21. CYGS* jobs.....	150
22. CYGT* jobs.....	151
23. CYGU* jobs.....	151

24. CYGV* jobs.....	151
25. CYGX* jobs.....	152

About this information

This information contains information you need to configure and use z/OS Change Tracker, a paid feature of z/OS 2.5.

Who should use this information

This document is intended for people who are responsible for implementing and administering z/OS Change Tracker, including system programmers, managers, and auditors.

Different readers may be responsible for tasks such as configuration and setup, change management, compliance management, security administration and auditing as well as backup and recovery.

z/OS information

This information explains how z/OS references information in other documents and on the web.

When possible, this information uses cross document links that go directly to the topic in reference using shortened versions of the document title. For complete titles and order numbers of the documents for all products that are part of z/OS, see *z/OS Information Roadmap*.

To find the complete z/OS library, go to [IBM Documentation \(www.ibm.com/docs/en/zos\)](http://www.ibm.com/docs/en/zos).

How to send your comments to IBM

We invite you to submit comments about the z/OS product documentation. Your valuable feedback helps to ensure accurate and high-quality information.

Important: If your comment regards a technical question or problem, see instead [“If you have a technical problem”](#) on page xv.

Submit your feedback by using the appropriate method for your type of comment or question:

Feedback on z/OS function

If your comment or question is about z/OS itself, submit a request through the [IBM RFE Community](#).

Feedback on IBM® Documentation

If your comment or question is about the IBM Documentation functionality, for example search capabilities or how to arrange the browser view, send a detailed email to IBM Documentation at ibmdocs@us.ibm.com.

Feedback on the z/OS product documentation and content

If your comment is about the information that is provided in the z/OS product documentation library, send a detailed email to mhvrcfs@us.ibm.com. We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information.

To help us better process your submission, include the following information:

- Your name, company/university/institution name, and email address
- The following deliverable title and order number: IBM z/OS Change Tracker Guide and Reference, SC31-5707-50
- The section title of the specific information to which your comment relates
- The text of your comment.

When you send comments to IBM, you grant IBM a nonexclusive authority to use or distribute the comments in any way appropriate without incurring any obligation to you.

IBM or any other organizations use the personal information that you supply to contact you only about the issues that you submit.

If you have a technical problem

If you have a technical problem or question, do not use the feedback methods that are provided for sending documentation comments. Instead, take one or more of the following actions:

- Go to the [IBM Support Portal](#).
- Contact your IBM service representative.
- Call IBM technical support.

Summary of changes

This information includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations for the current edition are indicated in the PDF by a vertical line to the left of the change.

Note: IBM z/OS policy for the integration of service information into the z/OS product documentation library is documented on the z/OS Internet Library under [IBM z/OS Product Documentation Update Policy \(www-01.ibm.com/servers/resourcelink/svc00100.nsf/pages/ibm-zos-doc-update-policy?OpenDocument\)](http://www-01.ibm.com/servers/resourcelink/svc00100.nsf/pages/ibm-zos-doc-update-policy?OpenDocument).

Summary of changes for IBM z/OS Change Tracker for z/OS Version 2 Release 5 (V2R5)

Changed

The following content is changed.

March 2023 refresh

- Information to describe support for the new 90-day trial license (APAR PH51954). See “90-day trial” on page 19 and the new messages “CYG111I” on page 155, “CYG117I” on page 156, and “CYG412W” on page 181.

February 2023 refresh

- New z/OSMF plug-in provides a graphical interface to monitor and manage resources on the protection list (APAR PH49337). See “z/OSMF plug-in” on page 1 and “z/OSMF plug-in considerations” on page 22.

December 2022 refresh

- New messages “CYG210W” on page 159, “CYG411W” on page 181, and “CYG427I” on page 181 and updated message “CYG266W” on page 171 relate to improved access and refresh handling (APAR PH49225).

September 2022 refresh

- Enhancements to several panels in the ISPF interface (APAR PH48359):
 - “Check out” on page 29
 - “Backup” on page 36
 - “SuperC member compare” on page 52
- Clarifications and other updates to messages: Chapter 6, “z/OS Change Tracker messages,” on page 153

June 2022 refresh

- Clarification of the NOTIFY=Y feature in “Data set protection” on page 7
- Updates to the Security checklist before configuring z/OS Change Tracker in “Security checklist” on page 20
- Updates to z/OSMF configuration process in “Configuring z/OS Change Tracker” on page 20
- Updates to the LOCK parameter in “Locking” on page 33
- Updates to Email alerts for critical protected libraries in “Email Alert” on page 33
- Updates to reflect zFS as the primary file system and withdrawal of support for HFS in z/OS 2.5:
 - Chapter 1, “Overview,” on page 1

- [“z/OS Change Tracker functionality” on page 2](#)
- [“Recover ” on page 46](#)
- [“Tokenize directory files” on page 65](#)
- [“Report changed files in a directory ” on page 67](#)
- [“Macro compare files” on page 69](#)
- [“List a token file” on page 70](#)
- [“LIST command” on page 94](#)
- [“OPTION command” on page 101](#)
- [“PATHCOMPARE command” on page 103](#)
- [“PROTECT command” on page 105](#)
- [“SAVEDD command” on page 117](#)
- [“SYSUT1, SYSUT2 command” on page 127](#)
- [“CYG257E” on page 168, “CYG310W” on page 177, “CYG326W” on page 179, “CYG821E” on page 194, “CYG882E” on page 200](#)
- Updates to Messages: [Chapter 6, “z/OS Change Tracker messages,” on page 153](#)

May 2022 refresh

- Updates to support for Japanese in [Chapter 1, “Overview,” on page 1.](#)
- Updates to compatibility with existing SYSChange installations in [“z/OS Change Tracker functionality” on page 2.](#)
- Updates to shutting down the started task in [“z/OS Change Tracker functionality” on page 2](#) and [“z/OS Change Tracker maintenance and housekeeping tasks” on page 22](#)
- Updates to UNPROTECT replacing DELETERESOURCE in [“Data set protection” on page 7](#) and [“z/OS Change Tracker maintenance and housekeeping tasks” on page 22.](#)
- Updates to recursively resolving a directory for tokenization in [“Tokenization of z/OS UNIX System Services directories” on page 16.](#)
- Updates to "About this Task" concerning CYG.** access in [“Configuring z/OS Change Tracker” on page 20.](#)
- Updates to the description of input fields in the Option and Parameters panel [“z/OS Change Tracker options” on page 26.](#)
- Updates to descriptions of menu options Check In and Recover Archive Backups [“Administration” on page 27.](#)
- Updates to the LOCK request in [“Protect a resource” on page 31.](#)
- Updates to the PATTERN syntax in example 3 for [“GROUPCOMPARE command” on page 92.](#)
- Updates to the VER parameter and usage section of [“MEMBERRECOVER command” on page 97.](#)
- Updates to the QUICKMODE parameter in [“PROTECT command” on page 105.](#)
- Updates to usage information in [“TOKEN command” on page 128.](#)
- Updates to usage information in [“TRIGGERREFRESH command” on page 132.](#)
- CYGK* and CYGH* job detail added to [Chapter 5, “Sample jobs,” on page 147.](#)
- Added [Chapter 6, “z/OS Change Tracker messages,” on page 153.](#)

Chapter 1. Overview

IBM z/OS Change Tracker is a change management tool for tracking, controlling, and managing changes in software, enhancing the security of the IBM z/OS operating environments.

z/OS Change Tracker is used to establish base versions for library members and automatically generates member versions as users introduce changes. All z/OS file structures are supported, including PDS, PDSE, Physical Sequential files, VSAM files, Direct Access, z/OS UNIX System Services, and zFS File Structures.

Key features

Key features in z/OS Change Tracker include:

- Real-time monitoring of all change activities in critical system software and mission-critical application software automatically.
- Detection of the key elements of changes.
 - Who made the change?
 - When was the change made?
 - What lines in the source member have changed, or which bytes were changed in load modules?
 - Why was the change made?
 - Which programs introduced the change?
- Automated member-level backup. Members of protected resources are automatically backed up creating stacked versions.
- Member-level recoverability. z/OS Change Tracker provides a list of stacked versions in an ISPF table. In this list, users can view contents of the member versions online, compare pairs of versions for details of the change, and recover the required version to resolve an unwanted change.
- Member-level control feature controls library changes. Selected groups of members can be made exclusively available to a list of authorized TSO IDs or a RACF® group.
- Enables documentation of changes introduced to critical members of a protected resource. When the user edits and saves a member, the user is prompted to document the reason for the change. This feature can be activated for selected resources when the data set is defined for protection.
- Change detection applied to production load libraries. This ensures system software and application software availability and integrity, using the z/OS Change Tracker tokenization technology.
- Comparing multiple local or remote software environments. z/OS Change Tracker identifies changes between any two z/OS environments. The report highlights missing data sets and missing or changed members. For z/OS UNIX or zFS directories, the report contains missing paths and missing or changed files in the two environments. If the two z/OS or zFS environments have the same configuration and the same contents, RC=0 is returned.
- z/OS Change Tracker is set up to use English as a supplied default but can be customized to use Japanese.

z/OSMF plug-in

The IBM z/OS Change Tracker z/OSMF plug-in provides an interactive GUI front-end for IBM z/OS Change Tracker. This plug-in is accessed from the z/OSMF desktop.

The plug-in includes a Monitored Resources view for managing resources on the protection list, and other features such as viewing the change history of data sets, member-level backup and recovery, and checkin and checkout of members.

z/OS Change Tracker functionality

z/OS Change Tracker features key functionality to assist in safely managing changes within the z/OS and z/OS UNIX environments.

Protecting critical libraries and automatic creation of member versions

The z/OS Change Tracker started task (STC) monitors data sets for changes. When a change is introduced to a protected member, the started task takes a backup. As a result, you can reliably recover from unwanted changes if necessary. Member backups are stored in the *data file*, and an associated comment describing the incident is recorded in the *control file*. These VSAM files, along with the archive data file and archive control file, are collectively referred to as the *repository*.

The started task is critical for the functioning of z/OS Change Tracker. To ensure the correct functioning and integrity of the repository, always shut down the started task in the recommended fashion. See [“z/OS Change Tracker maintenance and housekeeping tasks”](#) on page 22 for more information.

Using the CSAVE feature backs up members at the time the change is committed. Automatically generated comments allow for simple version identification in case a future recovery is required. Archived backups are made according to the schedule set by the user. Both Archived and Live backups are available online for viewing, comparing and recovery purposes using the ISPF interface.

Note:

- For the started task to be able take a member backup, it must have READ access to the data set containing the member. It is recommended that your security be set up so that the started task has READ access to all data sets that are protected by z/OS Change Tracker.
- z/OS Change Tracker will not perform a backup of any PDS or PDSE member that has more than 500,000 records.

Auditing and checking software integrity

There are two approaches available for auditing software and checking the integrity of the software environments. These approaches can be used concurrently to achieve your goals.

Approach 1:

Protect your critical libraries using the Protect a Resource function, and then audit the changes recorded real-time by the started task. This approach enables you to audit critical libraries by identifying who made the change, what action was involved (ADD, UPDATE, DELETE, RENAME, or ZAP), when the change was made, and which program was responsible for the change.

Approach 2:

Use tokenization to establish reference tokens representing the contents of data set members in your software environments. A token is a unique 8-byte binary representation of a data set member or non-PDS data set and can be stored in the z/OS Change Tracker repository or in a sequential data set. Periodically re-tokenize the data sets in your software environments and compare the new tokens with the old to identify and report any changed members.

Important: Where a large number of data sets might be involved, use a sequential data set to store the tokens. SAF-protect the sequential data set to minimize the possibility of corruption.

Integrity verification for a sizeable local environment

To perform Integrity Verification for a large environment such as the libraries in a production site, tokenize the data sets of interest in that environment. This process establishes a 4-byte hex number that represents the content of each tokenized library member. One token is generated for each member of a partitioned data set (PDS, PDSE source and load), and one token is produced for an entire physical sequential or a direct access file.

The same TOKENIZE function with the MODS option specified can be used to detect if the environment has changed. The MODS parameter identifies any changed members or files since the tokens were initially generated. Lack of change (RC=0) indicates that the environment has remained intact, and hence the integrity is assured.

Integrity verification between two environments (local or remote)

To perform integrity verification between two environments, for example to verify if production software and disaster recovery software are identical, the approach is similar. For any two environments, z/OS Change Tracker quickly identifies data sets which have differences, such as mismatched members, or members with different contents. To perform integrity verification between two remote environments such as production and disaster recovery, tokenize the entire production volumes storing the tokens into an external flat file, and transfer this token representation file to the target environment.

On the target environment, tokenize the similar environment to create a second token representation file. Compare the two token representation files using the COMPARE command. The compare report identifies and reports mismatched data sets. For matching data set names, the mismatched member names, as well as the matching member names with different contents are identified and reported.

Using the tokenization process to synchronize the DR system with the production system regularly will ensure the latest level of software. The process can be performed as often as necessary.

Protecting software assets

z/OS Change Tracker protects your critical data sets and library members from unauthorized change. Users can protect their software assets at the data set level, or at the member level (once critical libraries are protected with LOCK=YES). In the case of member level protection of protected libraries, member updates are allowed only if Checkout standards are used. Hence, the combination of data set level protection and member level protection provides the data center with complete and continuous protection.

Member level protection:

When the Administrator locks a data set using the LOCK=YES parameter, none of the members of that library can be changed unless the Checkout standards are used. This feature provides organizations with the assurance that their critical libraries and members are protected from the risks associated with unauthorized change and downtime. In order to allow update access to the specific members in that library, those members can be explicitly checked out by the Administrator. The checkout process makes one or more members available to the Administrator. Once the changes to the member have been implemented, the member is checked in by the Administrator. As an added safety feature, every time a checked out member is updated, the z/OS Change Tracker Started Task, monitoring the protected resources, automatically records the incident and backs up the member.

Utilities

z/OS Change Tracker provides a robust set of utilities. The user may require high-level comparison facilities for comparing pairs of PDS or PDSEs to determine similarities or differences at the member-level. Users may also wish to reconcile one or more pairs of partitioned data sets, to ensure that their member configurations are exactly alike.

Users can examine byte-level differences for members of libraries which are identified as being different by the z/OS Change Tracker internal tokenization operation (the DIFF members). To report the byte-level differences, once the z/OS Change Tracker tokenization has filtered out those members with identical contents (SAME members), the SuperC compare utility is invoked for DIFF members only. This way users can obtain a concise report of only the members which have some differences in their contents eliminating the need to spend time on the ones which are identical in their content.

Additionally, z/OS Change Tracker performs processing of the IBM z/OS File System (zFS) and z/OS UNIX. z/OS UNIX directories may be tokenized recursively by the z/OS Change Tracker tokenization process. The same z/OS UNIX environments can be re-tokenized at a later time, thus the z/OS UNIX files with different

contents will be identified and reported as being DIFF. Same process works to identify and report the DIFF between two remote environments.

Installing and running z/OS Change Tracker alongside SYSChange

z/OS Change Tracker can be installed and run alongside an existing installation of SYSChange in your environment, with certain limitations.

- Both z/OS Change Tracker and SYSChange must use different data files and control files.
- Data migration between z/OS Change Tracker and SYSChange is not supported.

Use cases

This section documents z/OS Change Tracker use cases.

Note: All jobs in this section require the following JCL unless stated otherwise.

```
//JOB
//STEP1 EXEC PGM=CYGMAIN,REGION=0M
//STEPLIB DD DISP=SHR,DSN=node.CYG100.LINKLIB
//CYGPARM DD DISP=SHR,DSN=node.CYG100.PARMLIB(CYGPARMS)
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CYGIN DD *
Commands
//*
```

To avoid complexities, only the CYGIN data is provided in the sample jobs.

Change tracking

There are two methods of change identification and automated member backup in z/OS Change Tracker.

- Method 1: The z/OS Change Tracker started task identifies updates to members of protected data sets as they occur, and creates versions of the updated members after each update operation. Assuming the data set is protected and the started task is running on the LPAR, the updated members are captured and backed up automatically.
- Method 2: The Surveillance function identifies the changed members and creates versions.

If a data center relies on using batch functions to introduce changes to their protected resources, they can use the second method to record their changes and to back up the changed members. z/OS Change Tracker allows you to eliminate the impact of batch operation on the started task. You can use the commands DEACTIVATESTC or OPTION=HALTOBJECT to temporarily stop the function of the started task.

Example 1: Running IEBCOPY while the started task has become deactivated

In this example, the started task has been deactivated in the first step of a job. While the started task is inactive, all the IEBCOPY steps are performed. In the last step of the job, the SURV function is run to identify the changed members, to record them, and to back them up. Finally, the started task is reactivated again.

```
//step0
DEACTIVATESTC
//step1 IEBCOPY
//step2 IEBCOPY
.
.
.
//step99
SHOW=PROTECTIONS
SHOW=LIST
SURV=(LIST,TTR=YES)
REACTIVATESTC
//*
```

Example 2: Running IDCAMS while the started task has become deactivated

In this example, the started task has been deactivated in the first step of an IDCAMS job. While the started task is inactive, all the IDCAMS DELETE steps are performed. The SURV function is run to record the deleted members. Finally, the started task is reactivated again.

Note: Because the started task role is temporarily eliminated, it is not necessary to run the IDCAMS on the target libraries with DISP=OLD.

```
//step0
DEACTIVATESTC
//step1 IDCAMS DELETE using DISP=OLD
//step2 IDCAMS DELETE using DISP=OLD
.
.
.
//step99
SHOW=PROTECTIONS
SHOW=LIST
SURV=(LIST,TTR=YES)
REACTIVATESTC
//*
```

Example 3: Expanding the size of a protected library

In this example, the procedure to expand the library is explained. It is assumed that the library A is a protected library with LOCK=YES.

1. Stop the started task.
2. Copy library A to an expanded library B.
3. Rename library A to C, and then B to A.
4. Run the RESYNC function while the started task is still deactivated.

```
//CYGIN DD *
RESYNC=(DSN=library-a)
//*
```

5. COPY additional members to the newly allocated library A.
6. Run the SURV to find the copied members.

```
//CYGIN DD *
SURV=(DSN=library-a,TTR=YES)
//*
```

7. Start the started task.

Note: Because the started task has been restarted, even though the VOLSER of the newly allocated library A may be different from the original library A, the command TRIGGERREFRESH is not required. When the started task builds its “protection list” it will find the correct VOLSER of the library A.

Note: Because the started task was stopped, the LOCK attribute has no impact. Thus, it is not needed to change the protection attribute of the library from LOCK=YES to LOCK=NO and reverse.

Example 4: Re-allocating a protected library on a different VOLUME

If a protected resource is relocated to a different volume, this data set relocation must be conveyed to z/OS Change Tracker. There are two cases to consider:

1. If the relocation was done while the started task was down, no action is required. When the started task restarts, the correct volume will be found.
2. If the relocation was done while the started task was active, the following command must be executed. Otherwise, tracking of this object will stop until the started task is recycled.

```
//CYGIN DD *
TRIGGERREFRESH
//*
```

Example 5: Procedure to force creation of member backups

If a member-level backup of a library is missed, you may run the following procedure to find the changed members and back them up.

1. Stop the started task or run the DEACTIVATESTC command.
2. Run the SURV function on the protected resource.

```
//CYGIN DD *  
SURV=(DSN=library,TTR=YES)  
/*
```

3. Confirm that the backups are generated.

```
//CYGIN DD *  
SHOW=BACKUPS=(MEM=*, DSN=your-library)  
/*
```

4. Start the started task or run the REACTIVATESTC command.

Example 6: Avoiding QSL0T error message being issued by the started task

Occasionally, if a protected PDSE was a target of a very large IEBCOPY operation, an internal condition was detected by the started task indicating that the QSL0T memory has no room. Although the IEBCOPY job runs successfully and all the members are successfully copied, the started task had stopped working.

To prevent this error, specify OPTION MASSCOPY in the CYGPARMs member. If this option is specified, when a mass copy or mass delete is detected, z/OS Change Tracker will temporarily HALT the operation of its started task for this target data set. As a result of the halt, member backups are not generated in real-time and activity recording is not performed; these activities are delayed to a later time when the SURV function is run. The temporary halt will be removed and the normal tracking will be resumed when the IEBCOPY or the IDCAMS jobs ends.

If this condition is detected, stop the started task, run the SURV command for all data sets in the protection list, and restart the started task. If the QSL0T error is encountered during a batch function, use the DEACTIVATESTC command, run the function, run SURV for the object, and then run REACTIVATESTC.

Example 7: Can multiple started tasks run on a system?

Because z/OS Change Tracker uses the SVC table to anchor its own intercepts, only one started task can run on a system at one time. However, for testing purposes you can stop the current started task that refers to the CYGPARMs member and start another one that reads CYGPARMt instead. The CYGPARMt member might refer to a TEST control file and data file, whereas the CYGPARMs refers to the PROD control file and data file.

Example 8: Using DEACTIVATESTC and REACTIVATESTC when deleting a large number of members

When deleting a large number of members from a library, usually when deleting more than 100 members, it is recommended to use the commands DEACTIVATESTC before starting the operation. The following is an example:

```
//step0  
DEACTIVATESTC  
//step1 IDCAMS DELETE using DISP=OLD  
//step2 IDCAMS DELETE using DISP=OLD  
.  
.  
.  
//step99  
SHOW=PROTECTIONS  
SHOW=LIST  
SURV=(LIST,TTR=YES)  
REACTIVATESTC  
/*
```


Data set protection

This section discusses some of the questions and answers regarding data set protection, real-time auditing of changes in critical system libraries, automatic backup of changed members, documenting changes introduced and other related functions.

How do I protect a resource?

There are two ways to protect data sets:

1. Using explicit data set names
2. Using data set patterns

How do I protect multiple explicit data sets?

The following statements protect 3 data sets while specifying LOCK=NO. The EXPLICIT commands build the list of data sets in memory. The PROTECT command protects all data sets in the LIST.

```
//CYGIN DD *  
LOGCOMMENT='PROTECT A LIST OF EXPLICIT DSN - CICS GROUP'  
*  
EXPLICIT=CICS.PROCLIB  
EXPLICIT=CICS.PARMLIB  
EXPLICIT=CICS.LOADLIB  
*  
SHOW=LIST  
*  
PROTECT=(LIST,  
LOCK=NO,  
INITIAL=NO,  
COPIES=20)  
/*
```

Refer to the sample job CYGA1PRO in the [Chapter 5, “Sample jobs,” on page 147](#).

How do I protect a pattern of data sets?

The following statements protect a pattern of data sets. The PATTERN commands resolve data sets from the catalog. The EXCLUDE commands exclude the data sets from the resolved list. The SHOW=LIST command shows the list of data sets to be protected. Finally, the PROTECT command operates on all data sets in the LIST.

```
//CYGIN DD *  
LOGCOMMENT='PROTECT A PATTERN OF DATA SETS'  
*  
PATTERN=SYS1.*  
PATTERN=SYS2.*  
EXCLUDE=SYS1.DUMP*  
EXCLUDE=SYS1.BROADCAST  
*  
SHOW=LIST  
*  
PROTECT=(LIST,  
LOCK=NO,  
INITIAL=NO,  
COPIES=20)  
/*
```

Refer to the sample job CYGA1PRO in the [Chapter 5, “Sample jobs,” on page 147](#).

What is the difference between LOCK=YES and LOCK=NO?

Protecting a data set with LOCK=NO provides a freehand mode operation. All members of the resource can be updated by any user observing the rules of RACF. z/OS Change Tracker monitors member-level activities of such a resource, records the changes, and automatically backs up the changed members to its repository.

LOCK=YES prevents users from updating members of a protected resource until a checkout is performed. This member-level update prevention is performed by the z/OS Change Tracker started task.

How does NOTIFY=Y work for documenting changes during an EDIT session?

When a data set is protected with NOTIFY=Y, the user will be reminded to use the CSAVE command when the edit of the member is finished. This feature is important for critical libraries such as SYS1.PARMLIB and SYS1.PROCLIB where everyone is supposed to use the CSAVE command. To implement this feature refer to Chapter 2, “Configuration and implementation,” on page 19.

```
PROTECT=(DSN=SYS1.PARMLIB,
  NOTIFY=Y,
  CSAVE=Y,
  LOCK=NO,
  INITIAL=NO,
  COPIES=20)
```

NOTIFY=Y will not work for a PDSE when enforced CSAVE or LOCK=Y is set. If either are specified with NOTIFY=Y, NOTIFY=Y will be ignored.

How do I generate a report of change activities in a protected resource?

The following statements report all change activities in SYS1.PARMLIB and SYS1.PROCLIB protected resources.

```
//CYGIN DD *
  SHOW=ACTIONS=(DSN=SYS1.PARMLIB, MEM=*)
  SHOW=ACTIONS=(DSN=SYS1.PROCLIB, MEM=*)
//*
```

How do I generate a report of DELETE activities in any of the protected data sets?

In the following statements, the first SHOW=PROTECTIONS command generates a list of all protected resources in memory. The second SHOW=ACTIONS=LIST command examines all data sets in the list to report the change activities. The change activities are filtered to only show the DELETE activities since the OPTION=DEL is specified.

```
//CYGIN DD *
  OPTIONS=DEL (other options are: ADD, UPD, REN, ALL)
*
  SHOW=PROTECTIONS
  SHOW=ACTIONS=LIST
//*
```

Can I generate a report of all changes that occurred yesterday?

Yes. Using the DATE command you may request for the change activities of a specific day in the past.

```
//CYGIN DD *
  DATE=TODAY-1
*
  SHOW=PROTECTIONS
  SHOW=ACTIONS=LIST
//*
```

Can I generate a report of all changes that occurred in the past 7 days?

Yes. Using the SINCE command you may request for the change activities since *N* days ago.

```
//CYGIN DD *
  SINCE=TODAY-7
*
  SHOW=PROTECTIONS
```

```
SHOW=ACTIONS=LIST
//*
```

Can I generate a report of all changes that occurred within a date range?

Yes. You can use BEGINDATE and ENDDATE commands. In the following example all change activities from midnight of January 1, 2022 until midnight of May 10, 2022 will be reported.

```
//CYGIN DD *
BEGINDATE=2022/01/01_00:00
ENDDATE=2022/05/10_23:59
SHOW=PROTECTIONS
SHOW=ACTIONS=LIST
//*
```

Can I back up a specific member of a data set with my own documentation?

Yes. Once a data set has become a protected resource by z/OS Change Tracker, the started task automatically backs up any changed members immediately after it has been saved. Users can initiate their own member backups with appropriate documentation for the backed-up member.

```
//CYGIN DD *
MEMBERBACKUP=(DSN=SYS1.PARMLIB,VOL=RES001,
MEM=MEM1,
COMMENT='Backup created by John before vacation')
//*
```

Refer to the sample job CYGA2BKP in the samples library.

How can I recover a previous version of a member?

To recover a version in batch, run the following statements:

```
...
//DD1 DD DISP=SHR,DSN=IBMU11.RECOVERY.PDS01
//CYGIN DD *
MEMBERRECOVER=(DSN=SYS1.PARMLIB,VOL=RES001,
MEM=MEM1,
VER=-1, RECOVERS THE PREVIOUS VERSION
OUTDD=DD1)
```

Refer to the sample job CYGA3REC in the samples library.

Can I recover all previous versions of a member?

Yes. To recover all versions of a specific member, run the following statements:

```
...
//DD1 DD DISP=SHR,DSN=IBMU11.RECOVERY.PDS01
//CYGIN DD *
MEMBERRECOVER=(DSN=SYS1.PARMLIB,VOL=RES001,
MEM=MEM1,
VER=*, RECOVERS ALL VERSIONS
OUTDD=DD1)
```

Refer to the sample job CYGA3REC in the samples library.

Can I recover a specific version of a member backup using ISPF?

Yes. Use option 3 Recover from the z/OS Change Tracker ISPF main menu to list all versions of a specific member. You can then select one version for recovery or view the contents online without recovering it. Pairs of versions can be selected to report the individual lines of difference. For load modules, the differences between versions are reported on a byte level.

Can I generate a list of all available backups for all members?

Yes. To obtain a list of all versions available in the z/OS Change Tracker repository, the following sample job may be run:

```
//CYGIN DD *  
SHOW=BACKUPS=(DSN=SYS1.PARMLIB,  
MEM=*)  
//*
```

Refer to the sample job CYGA4DEL in the samples library.

How do I remove a data set from the z/OS Change Tracker protection list?

To remove a data set from being protected by z/OS Change Tracker, use the following statements:

```
//CYGIN DD *  
UNPROTECT=(DSN=SYS1.PARMLIB,VOL=RES001)  
//*
```

In this example the VOL parameter has been specified to refer to an uncataloged protected resource. Refer to the sample job CYGA5REM in the samples library.

How do I update a member in a locked (LOCK=YES) data set?

A member of a locked data set must be checked out to a user or group before it can be updated. It can then be updated only by that user or members of that group. The started task will prevent any update attempt by other users. The CHECKOUT command performs this function and may only be run by the administrator.

```
//CYGIN DD *  
CHECKOUT=(DSN=SYS1.PROCLIB,  
MEMBER=IPCS,  
COMMENT=('CHECKOUT TO IBMU11'),  
USERID=IBMU11)  
ENDCHECK  
//*
```

Refer to the sample job CYGA6CKO in the samples library.

Can I check out a member to a RACF group?

Yes. The administrator can use the following statements to check out a member to a predefined RACF group. Multiple users in the RACF group will have update access to the checked out member.

```
//CYGIN DD *  
CHECKOUT=(DSN=SYS1.PROCLIB,  
MEMBER=IPCS,  
COMMENT=('CHECKOUT to multiple users in a RACF group'),  
RACFGROUP=CICSTEAM)  
*  
ENDCHECK  
SHOW=UNIVERSE  
//*
```

How do I relinquish ownership of a member?

When you no longer need the member for your exclusive control, an Administrator can check in that member using the following statements:

```
//CYGIN DD *  
CHECKIN=(DSN=SYS1.PROCLIB,  
MEM=IPCS,  
COMMENT='MODIFY WAS APPROVED BY CICS TEAM LEADER')  
*
```

```
ENDCHECK
//*
```

Refer to the sample job CYGA7CKI in the samples library.

Can I see which members have already been checked out from a library?

Yes. To view which members have been checked out from a specific resource, use the Check In option from the Administration ISPF menu. In batch, you can use the SHOW=CHECKOUT command to list all checked out members in all protected libraries.

```
//CYGIN DD *
SHOW=CHECKOUT
//*
```

Can I change the LOCK status of a resource which has already been protected?

Yes. Using the following job an Administrator can change the lock status of a resource. If it has already been protected with LOCK=NO, this job will change it to LOCK=YES. Keep in mind that by setting the LOCK status to YES, none of the members in the library can be updated by any user unless a checkout is performed by an Administrator.

```
//CYGIN DD *
REPROTECT=(DSN=SYS1.PROCLIB,
LOCK=YES)
//*
```

Refer to the sample job CYGA8REP in the samples library.

What would happen if the started task is stopped? Will introduced changes be lost if the started task is not active?

No, the Surveillance feature will capture changes that occur while the started task is inactive. The next time the task restarts all changes introduced will be picked up and the changed members are backed up. Such recorded changes collected by the SURV function are identified in the change report as being collected by the CYGMAIN program. The z/OSMF supplied started task job and procedure have a Surveillance step which runs before the started task is started. It is recommended that they not be modified.

```
//STEPSURV
//CYGIN DD *
SHOW=PROTECTIONS
SURV=(LIST,CHG=YES)
//*
//STEPREP
//CYGIN DD *
SHOW=PROTECTIONS
SHOW=ACTIONS=LIST
//*
```

Refer to the sample job CYGA9SRV in the samples library.

Can I generate a list of protected data sets being controlled by z/OS Change Tracker?

Yes. Using the statements below, an Administrator can generate a list of currently protected resources.

```
//CYGIN DD *
SHOW=PROTECTIONS
//*
```

What needs to be done if an already protected resource is moved to a different DASD volume?

If a cataloged data set is already protected by z/OS Change Tracker, and it is moved to a DASD volume other than the one already known by the started task, either the task must be recycled, or the TRIGGERREFRESH command must run by an Administrator to trigger a protection list refresh.

```
//CYGIN DD *  
  TRIGGERREFRESH  
//*
```

Tokenization

This section discusses some of the questions and answers regarding tokenization.

What does tokenization mean for z/OS Change Tracker?

A token represents the content of a PDS member or the content of a PS or DA file. For a PDS or PDSE one token is generated for each member.

Note: Tokenization is not supported for encrypted data sets.

How do I compare pairs of data sets when all data sets are available on the same processor?

There is no limit to the number of pairs that can be compared using z/OS Change Tracker. Data sets in the pair can be cataloged or uncataloged.

```
//CYGIN DD *  
  OPTION=DIFF (MISMATCH, SAME, MODS ARE THE OTHER OPTIONS)  
*  
  NEWDSN=SYS1.PARMLIB  
  OLDDSN=SYS2.PARMLIB  
  COMPARE  
*  
  NEWDSN=SYS1.LINKLIB  
  OLDDSN=SYS2.LINKLIB  
  COMPARE  
//*
```

Refer to the CYGT* sample jobs in [Chapter 5, “Sample jobs,” on page 147](#).

How do I compare two groups of data sets when the data set names do not directly match?

Use the STRINGREPLACE (STRREPL) command. In this example, you build two lists of data sets in memory. The first list refers to a group of cataloged data sets starting with SYS1, and the second list refers to another group of data sets starting with SYS7. Using the STRREPL command, the lists are properly matched in memory. Once two data sets are matched by name, the member-by-member compare operation begins. This time, member names are matched and their tokens are evaluated to decide whether they have the same contents.

```
//CYGIN DD *  
  OPTION=DIFF (MISMATCH, SAME, MODS ARE THE OTHER OPTIONS)  
*  
  STRREPL=('SYS1','SYS7')  
*  
  EXPLICIT=SYS1.PARMLIB  
  EXPLICIT=SYS1.PROCLIB  
  EXPLICIT=SYS1.VTAMLIB  
  SHOW=LIST  
*  
  EXPLICIT=SYS7.PARMLIB  
  EXPLICIT=SYS7.PROCLIB  
  EXPLICIT=SYS7.VTAMLIB  
  SHOW=LIST
```

```
*
GROUPCOMPARE
//*
```

How do I compare two groups of data sets when one group is cataloged and the other group is not cataloged?

This example builds two lists of data sets and then uses the GROUPCOMPARE command to properly match the data set names. The first LIST refers to a group of cataloged data sets and the second LIST refers to a group of uncataloged data sets. After the two lists are created in memory, the data sets are matched. Once two data sets are matched, compare of the two data sets begin by matching the member names and evaluating their tokens to decide whether they have the same contents.

```
//CYGIN DD *
OPTION=MODS (MISMATCH, SAME, DIFF ARE THE OTHER OPTIONS)
*
EXPLICIT=SYS1.PARMLIB
EXPLICIT=SYS1.PROCLIB
EXPLICIT=SYS1.VTAMLIB
SHOW=LIST
*
INCLUDE=(VOL=RES001,DSN=SYS1.PARMLIB)
INCLUDE=(VOL=RES001,DSN=SYS1.PROCLIB)
INCLUDE=(VOL=RES001,DSN=SYS1.VTAMLIB)
SHOW=LIST
*
GROUPCOMPARE
//*
```

Refer to the CYGT* sample jobs in [Chapter 5, “Sample jobs,”](#) on page 147.

Can I compare all data sets on two DASD volumes?

Yes. You can build two lists and then compare the data sets on both lists. Using the INCLUDE commands, data sets on two DASD volumes are placed in the lists in memory. Then, the GROUPCOMPARE is used to match the data set names and compare them.

```
//CYGIN DD *
OPTION=DIFF (MISMATCH, SAME, MODS ARE THE OTHER OPTIONS)
*
INCLUDE=(VOL=RES001,DSN=*)
EXCLUDE=SYS1.BROADCAST
EXCLUDE=SYS1.DUMP*
SHOW=LIST
*
INCLUDE=(VOL=RES002,DSN=*)
EXCLUDE=SYS1.BROADCAST
EXCLUDE=SYS1.DUMP*
SHOW=LIST
*
GROUPCOMPARE
//*
```

How do I tokenize a group of explicit data sets?

The following statements tokenize a group of data sets and then store the tokens in the Control File. Storing the tokens in the Control File (the default method) provides an effective solution to identifying and reporting changes introduced to an environment over time.

```
//CYGIN DD *
LOGCOMMENT='tokenize a list of explicit data sets - CICS GROUP'
*
EXPLICIT=CICS.PROCLIB
EXPLICIT=CICS.PARMLIB
EXPLICIT=CICS.LOADLIB
*
SHOW=LIST
TOKEN=LIST
//*
```

Refer to the sample job CYGB1TOK in [Chapter 5, “Sample jobs,” on page 147](#).

How do I tokenize uncataloged data sets?

The following statements show the use of multiple TOKEN commands. For uncataloged data sets, the VOL parameter must be specified. For cataloged data sets, the VOL parameter must be omitted.

```
//CYGIN DD *
LOGCOMMENT='TOKENIZE MULTIPLE DATA SETS'
*
TOKEN=(DSN=SYS1.PARMLIB,VOL=RES001)
TOKEN=(DSN=SYS1.PARMLIB,VOL=RES002)
TOKEN=(DSN=SYS1.PROCLIB)
TOKEN=(DSN=SYS1.LINKLIB)
TOKEN=(DSN=SYS1.VTAMLIB)
//*
```

Refer to the sample job CYGB1TOK in [Chapter 5, “Sample jobs,” on page 147](#).

Can I tokenize patterns of data sets?

Yes. The following statements tokenize multiple patterns of data sets and then store the tokens in the Control File. Unwanted data set patterns can be excluded from the LIST using EXCLUDE commands.

```
//CYGIN DD *
LOGCOMMENT='TOKENIZE A GROUP OF DATA SETS; MVS TEAM'
*
PATTERN=SYS1.*
PATTERN=SYS2.*
EXCLUDE=SYS1.DUMP*
EXCLUDE=SYS1.BROADCAST
*
SHOW=LIST
TOKEN=LIST
//*
```

Refer to the sample job CYGB1TOK in [Chapter 5, “Sample jobs,” on page 147](#).

How do I identify and report changes in a list of previously tokenized data sets?

The following statements identify the changes in 3 data sets that were previously tokenized. This example assumes that the tokens have been stored in the Control File.

```
//CYGIN DD *
OPTION=MODS
*
EXPLICIT=CICS.PROCLIB
EXPLICIT=CICS.PARMLIB
EXPLICIT=CICS.LOADLIB
*
SHOW=LIST
TOKEN=(LIST,MODS)
//*
```

Refer to the sample job CYGB2MOD in [Chapter 5, “Sample jobs,” on page 147](#).

Can I tokenize a group of libraries and store the tokens into an external file?

Yes. The following statements tokenize multiple data sets and then store the tokens to an external token file. By default, tokens are stored in the Control File unless the NOWRITE parameter in the TOKEN command is specified.

```
...
//SYSUT2 DD DISP=SHR,DSN=node.OLDTOKEN
//CYGIN DD *
*
* A LIST OF EXPLICIT OR DATA SET PATTERNS TO BE TOKENIZED
*
EXPLICIT=DEV.ATM.INSTALL
```



```

EXPLICIT=DEV.ATM.LINKLIB
EXPLICIT=DEV.ATM.SAMPLES
SHOW=LIST
*
TOKEN=(LIST,NOWRITE)
CLOSE
//*

```

In this case, the tokens are stored in a PS file referenced by //SYSUT2 DD. This method is recommended if the tokens need to be transferred to a remote environment as a representation of the first environment.

The attributes of the token file should be:

```

DCB=(DSORG=PS,RECFM=FB,LRECL=4096,BLKSIZE=4096)

```

Refer to the sample job CYGD1TOK in [Chapter 5, “Sample jobs,” on page 147](#).

How do I compare two groups of software not available to the same processor?

As shown in the previous example, two environments can be tokenized and the external token files can be brought together on the same processor. Token files can be transferred between systems using protocols such as NDM, XMIT, or FTP, or they can be sent as e-mail attachments.

The following statements show how two token files can be compared to identify and report the differences in the configuration and the contents.

```

...
//DD1 DD DISP=SHR,DSN=node.NEWTOKEN <== NEW TOKEN FILE
//DD2 DD DISP=SHR,DSN=node.OLDTOKEN <== OLD TOKEN FILE
//CYGIN DD *
OPTION=MODS
*
DD1SHOW
SHOW=LIST BUILDS THE LIST1 IN MEMORY
*
DD2SHOW
SHOW=LIST BUILDS THE LIST2 IN MEMORY
*
REMOTEGROUPCOMP
//*

```

Refer to the sample job CYGD3CMP in [Chapter 5, “Sample jobs,” on page 147](#).

Can I synchronize two groups of local libraries?

Yes. The following statements refer to multiple pairs of libraries. Each pair is internally tokenized and then the tokens are compared in memory to create a list of members being different in their contents (the DIFF members). Then, using the COPYDIFF command these members are copied from the source PDS to the target PDS. Hence, if there are only a few members different only those members are copied rather than copying the entire PDS. Moreover, the members existing in the target PDS but missing from the source PDS are optionally deleted (DELETEMEM=YES) from the target to achieve identical pairs of PDS. Only the target PDS is the subject of change and the source PDS remains unaltered.

The TESTRUN=YES command provides a simulation run showing the scope of change between the two environments without an actually copying or deleting any members.

```

//CYGIN DD *
TESTRUN=NO
COMPRESS=YES
*
DELETEMEM=YES
DSN1=IBMU11.TEST.PDS1
DSN2=IBMU12.TEST.PDS1
COPYDIFF
*
DELETEMEM=NO
DSN1=IBMU11.TEST.PDSE2
DSN2=IBMU12.TEST.PDSE2

```

```
COPYDIFF
//*
```

Refer to the sample job CYGB5CPY in [Chapter 5, “Sample jobs,” on page 147](#).

Can I reconcile differences between pairs of uncataloged data sets?

Yes. The first pair of partitioned data sets in the sample job below is uncataloged. The VOL1 and VOL2 commands must refer to the volumes where these data sets are located on. In the second pair, only the first data set in the pair is uncataloged and the second one is a cataloged data set.

```
//CYGIN DD *
TESTRUN=NO
COMPRESS=YES
*
VOL1=VOLSER
VOL2=VOLSER
DSN1=IBMU11.TEST.PDS1
DSN2=IBMU12.TEST.PDS1
COPYDIFF
*
VOL1=VOLSER
DSN1=IBMU11.TEST.PDSE2
DSN2=IBMU12.TEST.PDSE2 /* ONLY DSN2 IS CATALOGED */
COPYDIFF
//*
```

Refer to the sample job CYGB6CPY in [Chapter 5, “Sample jobs,” on page 147](#).

Can I use z/OS Change Tracker to synchronize our production software with the Disaster Recovery software?

Yes. To learn more about this capability refer to the CYGD* jobs in [Chapter 5, “Sample jobs,” on page 147](#).

How do I verify whether two software environments are identical in their configuration and their contents?

Using the z/OS Change Tracker tokenization capability, this task is quite simple. By tokenizing each environment and storing the tokens on external token files, a representation of each environment is established. Then, these token files must be made available on the same processor for comparison by the REMOTEGROUPCOMP command.

Refer to the CYGT* jobs in [Chapter 5, “Sample jobs,” on page 147](#).

Tokenization of z/OS UNIX System Services directories

How do I tokenize one or multiple z/OS UNIX directories?

Using the z/OS Change Tracker tokenization technology, one or multiple z/OS UNIX directories may be tokenized. In the following example all regular files in both directories are tokenized. z/OS Change Tracker opens the specified directory and makes a list of all regular files sorted according to the suffixes of the files. For example, all files with the suffix of .src will be grouped together. For each file a token is generated and stored in the external token file referred to by the //DD1.

```
..
//DD1 DD DISP=SHR,DSN=node.USS.OLDTOKEN
//CYGIN DD *
*
SYSUT1=PATH='/u/jones/*.*'
SYSUT1=PATH='/u/smith/*.*'
SAVEDD=DD1
//*
```

Refer to the sample job CYGU2TOK in [Chapter 5, “Sample jobs,” on page 147](#).

Can I compare files in two z/OS UNIX directories without first tokenizing them?

Yes. Using the following sample job you may compare regular files in two z/OS UNIX directories. Since both directories are available on the same processor tokenization is not required. Note that the USSPATHREPL command is required to establish matching path names.

```
//CYGIN DD *
OPTION=MODS
*
SYSUT1=PATH='/u/home/base/*.*'
SYSUT2=PATH='/u/home/target/*.*'
*
USSPATHREPL=(' /u/home/base' , ' /u/home/targ')
PATHCOMPARE
//*
```

Other possible reporting options are the following:

- OPTION=CHANGEONLY Shorter report of changes only
- OPTION=MISMATCH report mismatch file names ADD or DEL
- OPTION=DIFF If two files are matched in name, DIFF examined
- OPTION=SAME Sameness examined

Can I tokenize specific file types in a directory (specific suffixes)?

Yes. z/OS Change Tracker allows users to indicate which file types need to be tokenized. In the following example, multiple specific file types are tokenized rather than all files in the specified directory.

```
...
//DD1 DD DISP=SHR,DSN=node.USS.OLDTOKEN
//CYGIN DD *
*
SYSUT1=PATH='/u/jones/*.jcl'
SYSUT1=PATH='/u/jones/*.src'
SYSUT1=PATH='/u/jones/*.txt'
SAVEDD=DD1
//*
```

What is the advantage of the z/OS Change Tracker tokenization for the z/OS UNIX files?

Similar to the discussions for z/OS environment tokenization, the tokenization technology for z/OS UNIX environments enables users to identify changes and report the changes to other systems. To accomplish this, the same environment may be tokenized over a period of time. By comparing the OLDTOKEN file and the NEWTOKEN file, all changes are identified and reported.

How do I compare the z/OS UNIX token files to identify and report the changes?

The following statements compare two pre-generated token files using the PATHCOMPARE command.

```
...
//DD1 DD DISP=SHR,DSN=node.USS.OLDTOKEN
//DD2 DD DISP=SHR,DSN=node.USS.OLDTOKEN
//CYGIN DD *
LOADDD=DD1
LIST
LOADDD=DD2
LIST
*
PATHCOMPARE
//*
```

Can I compare two z/OS UNIX environments where the beginning pathnames are different?

Yes. Let us assume that the following two directories are tokenized, and the tokens are now stored in the files OLDTOKEN and NEWTOKEN.

```
SYSUT1=PATH='u/home/base/*.*'    tokens are stored in OLDTOKEN
SYSUT1=PATH='u/home/target/*.*'  tokens are stored in NEWTOKEN
```

Using the USSPATHREPL command, associate the path names to cause a matching path between the two environments and then compare the files tokens.

```
...
//DD1 DD DISP=SHR,DSN=node.USS.OLDTOKEN
//DD2 DD DISP=SHR,DSN=node.USS.NEWTOKEN
//CYGIN DD *
  USSPATHREPL=(' /u/home/base' , ' /u/home/target')
*
LOADDD=DD1
LIST
LOADDD=DD2
LIST
*
PATHCOMPARE
//*
```

How do I resolve directories recursively for z/OS Change Tracker to tokenize?

See sample jobs [CYGU2TBF](#) and [CYGU3TAF](#) for examples of recursive tokenization.

Chapter 2. Configuration and implementation

z/OS Change Tracker is configured using IBM z/OS Management Facility (z/OSMF). This chapter provides information on accessing the SMP/E workflow and product implementation guidelines.

Detailed information and the core instructions for configuring z/OS Change Tracker are presented in the Configure zOS Change Tracker workflow.

The z/OS Change Tracker plug-in is installed by using an optional step in the Configure zOS Change Tracker workflow. Information and instructions for configuring the plug-in are presented in the workflow step.

90-day trial

You can try z/OS Change Tracker using the 90-day trial. You have 90 days from the day the trial begins to use and evaluate the feature.

Setting up the trial

Activating the trial is a simple process that does not require assistance from IBM. To begin your trial, update the dynamic element parmlib member:

1. Create a copy of IFAPRD00 or your currently active IFAPRDxx member to enable the trial.
2. Update the new member with the following definitions:

```
PRODUCT OWNER('IBM CORP')
NAME('z/OS')
ID(5650-ZOS)
VERSION(*)
RELEASE(*)
MOD(*)
FEATURENAME('CHNGTRKR_TRIAL90')
STATE(ENABLED)
```

3. Issue the SET PROD command to specify the new member as active on the system.
4. Update IEASYSxx to point to the new member.

Trial duration

Every time a z/OS Change Tracker started task is started it will display a z/OS console message indicating the number of days left in the trial. No disablement is required when the trial period ends. z/OS Change Tracker will not start, any active started tasks will be shut down, and a console message will be issued.

To end the trial early, use either of the following steps to prevent z/OS Change Tracker from starting:

- Edit the IFAPRDxx member that enabled the trial to remove the entire section associated with FEATURENAME('CHNGTRKR_TRIAL90').

Or:

- Change STATE(ENABLED) to STATE(DISABLED).

While these steps are not required, they are necessary if you want to prevent z/OS Change Tracker being restarted during the 90-day duration. You might also choose to perform them for system hygiene after the trial ends.

The trial cannot be paused. The original start time is used to calculate the 90-day interval. If you mistakenly remove or disable the IFAPRDxx statement during the trial, the 90-day duration continues. Replace or enable the IFAPRDxx statement as soon as possible to continue the trial.

When the trial is over

After 90 elapsed days, you must purchase a license to continue to use z/OS Change Tracker.

Security checklist

Before you configure z/OS Change Tracker, a security review is suggested for your environment.

Roles

z/OS Change Tracker grants different levels of access and involvement depending on user role: *administrator, user, or auditor.*

Administrators can define data sets to be monitored and protected, and can grant individuals access to update locked members.

Users can verify that software on multiple local or remote systems is identical. Users can view and compare backups. If any differences are found, z/OS Change Tracker identifies which members have different contents.

Auditors are participants who are able to receive and view output that is relevant to auditing, such as member update actions.

Security considerations

- Set up security profiles for the Auditor and the Administrator.
- Separate the roles of Auditors and Administrators.
- Limit the number of people who can act as Administrators or Auditors.
- RACF protect the started task procedure and job.
- RACF protect the PARMLIB that is used to configure the product.
- Ensure only authorized users are able to start and stop the started task.
- Ensure that the started task has READ access to all data sets that are protected by z/OS Change Tracker.
- RACF protect the repository control, data and archive files.
- Establish a procedure to SMS-manage and encrypt backups of repository control files, data files, and tokenization files.
- Consider operating with STCPRINTOFF for added security.

Configuring z/OS Change Tracker

z/OS Change Tracker comes preinstalled with z/OS. It is a priced feature of z/OS that must be enabled before it can be configured and used. Configuration and deployment is done using a z/OS Management Facility (z/OSMF) workflow.

Before you begin

- Ensure that the z/OSMF server is properly configured and that you have the necessary authorization to use it.

Tip: For a new installation of z/OSMF, begin by creating a base configuration.

- Complete the SMP/E installation outlined in *z/OS V2R5 Program Directory* (GI11-9848-05).
- Ensure that the z/OS Change Tracker administrator has update access to CYG.** data sets.
- If you use WLM, note the following recommendations.

1. Assign the z/OS Change Tracker started task (STC) to a WLM service class that has low or medium importance.

2. Do not assign the started task to a WLM service class that is associated with a discretionary goal.
- If z/OS Change Tracker will run in a sysplex environment, refer to [Figure 1 on page 21](#). Decide which system to configure z/OS Change Tracker on. Decide which other systems in the sysplex also need to run the z/OS Change Tracker started task.

About this task

In a sysplex environment, the z/OS Change Tracker repository, which contains the VSAM data files and control files, is on DASD shared by all systems in the sysplex. Each started task is configured to access the shared repository files.

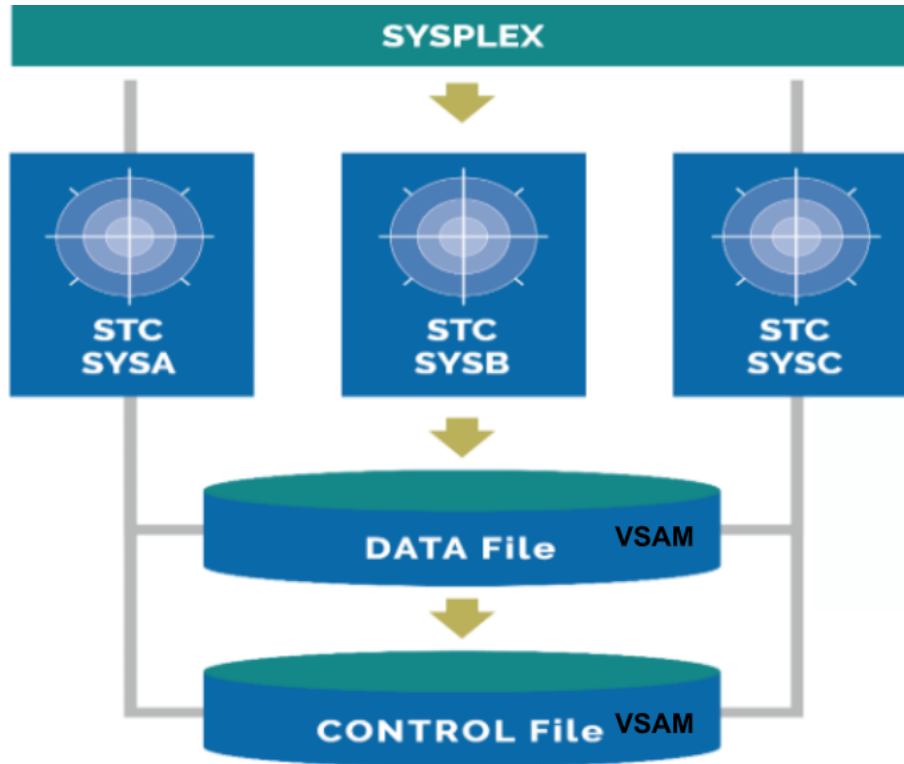


Figure 1. Typical z/OS Change Tracker sysplex configuration

Procedure

1. Load z/OS Change Tracker workflow definition file

- a) Log in to z/OSMF and open the **Workflows** window.
- b) From the Actions menu, select **Create Workflow**.
- c) Open the workflow definition file, which is located in: `/usr/lpp/cyg/zosmf/workflows/cygwflw.xml`
- d) Leave the variable input file field blank. Select **Next** to review the workflow.
- e) Select **Finish** to create the workflow.

2. Complete the z/OS Change Tracker workflow

Open the workflow and perform the steps in sequence.

Important: Ignoring prerequisites or performing steps out of sequence might result in an unstable configuration.

Results

The workflow creates a CYGPARMs data set member that contains configuration parameters which control the operation of z/OS Change Tracker. This member is referred to by all z/OS Change Tracker processes, tasks, and jobs.

When the workflow has successfully completed, z/OS Change Tracker is configured and ready to use.

z/OSMF plug-in considerations

The IBM z/OS Change Tracker z/OSMF plug-in provides an interactive GUI front-end from the z/OSMF desktop.

The plug-in includes a Monitored Resources view for managing resources on the protection list, and other features such as viewing the change history of data sets, member-level backup and recovery, and checkin and checkout of members.

The plug-in is installed as an optional step in the overall Configure z/OS Change Tracker workflow. For information and instructions, run the z/OSMF Plugin configuration step in the workflow.

If you do not yet have z/OSMF configured on your system, refer to the [z/OSMF Configuration Guide](#) in the IBM Documentation.

When z/OSMF has been configured and the plug-in is installed, open the Change Tracker task by double-clicking the IBM z/OS Change Tracker icon on the z/OSMF desktop.

For information about the available features and functions in each view, click the Help button to display the corresponding topic in the z/OSMF online help.

Diagnosing problems in z/OSMF

If you encounter any problems with the plug-in, try the following steps.

- To clear transient display problems, refresh the browser window.
- Close and restart the plug-in from the z/OSMF desktop.
- Check the History page for recent messages that might be relevant.
- Check the z/OSMF log file for possibly useful information. The z/OSMF log file is in a configurable location. For information about working with z/OSMF log files, see [z/OSMF log files](#) in *IBM z/OS Management Facility Configuration Guide*.

To determine the service level of the IBM z/OS Change Tracker plug-in, click Settings from the menu bar in the header. The service level is displayed under **Plug-in version**.

Removing the Change Tracker task from z/OSMF

To remove the Change Tracker task from z/OSMF, use the Import Manager task to import the properties file named `pluginDelete.properties`.

z/OS Change Tracker maintenance and housekeeping tasks

Housekeeping refers to tasks associated with freeing up space in the data and control files and improving run-time performance. These tasks include periodic backup and reorganization of the VSAM data sets and performing mass updates while the started task is inactive.

Shutting down the z/OS Change Tracker started task (STC)

To shut down the started task, use the MVS MODIFY command:

```
MODIFY cygstc,SHUTDOWN
```


or

```
MODIFY cygstc,STOP
```

where *cygstc* is the jobname of the started task. Avoid canceling the job as this may lead to incomplete updates and difficulties in restarting the started task.

Regular backup and reorganization of VSAM data sets

Regular maintenance of the z/OS Change Tracker VSAM files is required to ensure proper operation of the product. Perform the following steps weekly to keep the size of the VSAM files manageable, and to leave ample free space for the operations of the following week.

1. Shut down the started task on all LPARs that access the files.
2. Back up all four VSAM files (ARCHIVE . CONTROL, ARCHIVE . DATA, CONTROL, DATA) using IDCAMS REPRO or a similar product that does record-level I/O. Examples can be found in the CYGWK* members of the SCYGSAMP data set.
3. Run the following AUTOARC command in a job to examine all protected data sets and archive any extra member versions beyond those created by the “number of copies to keep” or “days to keep” parameters. (Note: The “number of copies to keep” and “days to keep” parameters are part of the “PROTECT command” on page 105.)
4. Run any other UNPROTECT and ARCHIVERESOURCE commands if necessary.
5. Back up and restore the 4 VSAM files. This will reorganize them.
6. Restart the started task on all LPARs.

Mass update in a shared environment

Some operations may generate a large number of updates to the z/OS Change Tracker VSAM files. For example, the PROTECT command with the INITIAL=YES parameter, if run against a data set with many members, will cause a large number of records to be inserted into the data file as a backup for each member is created. Other commands such as AUTOARC and ARCHIVERESOURCE can potentially do the same. In a sysplex environment, z/OS Change Tracker performs serialization to ensure the integrity of the VSAM Files, but performance issues may arise. If a performance issue does arise in these situations, it can be improved by running the commands when the started task is inactive on all LPARs.

Recovering from LOGICAL or PHYSICAL VSAM errors

In rare cases, the sharing of VSAM files in a sysplex environment may cause errors in the z/OS Change Tracker VSAM files. Two common errors are:

- Invalid Control Interval (R15=8 and the RPL feedback=X'9C')
- Error in the Sequence Set (R15=12 and the RPL feedback=X'08')

These errors indicate that the VSAM file is corrupted in some way and so must be corrected. Usually, a logical data backup (not a cluster backup) and restore of the file in error, as described in the following steps, corrects the problem.

1. Stop the started task on all LPARs.
2. Run the CYGEXAM job in the SCYGSAMP data set to run IDCAMS EXAMINE on all the VSAM files. This job identifies which files have logical or physical errors.
3. Perform a logical backup of the file in error using IDCAMS REPRO or a similar product that performs logical I/O.
4. If the REPRO runs successfully, the CYGRC1BK and CYGRC2RE jobs in the SCYGSAMP data set can be used to rebuild the clusters.

5. If the REPRO fails, the files can only be recovered by restoring the latest synchronized backup of the 4 VSAM files. After the restore is complete, run the VERIFY command as shown below to verify the integrity of the control file.

```
//CYGIN DD *  
SHOW=PROTECTIONS  
VERIFY=LIST  
//*
```

GRS setup: If Global Resource Serialization (GRS) is set up correctly, you do not need to perform additional configuration for VSAM file sharing. z/OS Change Tracker issues ENQ/DEQ using the SCOPE SYSTEMS and RNL=NO, and performs serialization to update VSAM files residing on a shared DASD correctly.

Chapter 3. Using z/OS Change Tracker with ISPF

This chapter describes the z/OS Change Tracker options and facilities available using the ISPF panels. Invoke **z/OS Change Tracker** from the initial ISPF menu or execute CYGISPF CLIST to display the Primary Option Menu.

IBM z/OS Change Tracker		
Option ==>		
0	Options	Set processing options and review the PARMs
1	Administration	Manage the protected resources
2	Audit	Audit change activities in protected resources
3	Recover	Browse, edit and recover backups
4	Disaster Recovery	Tokenize and compare PROD and DR environments
5	Utilities	Perform utility functions

Figure 2. Primary option menu

Note: Field-level help is available for all panels of the product. Press F1 key on any given field to display help.

The menu options are described in the following table:

Table 1. Primary menu options	
Option	Description
0	<p>Options</p> <p>This option allows you to review and modify z/OS Change Tracker parameters and options such as the user profile, and jobcard information. The parameters set in the CYGPARMs member are also displayed here as a table.</p>
1	<p>Administration</p> <p>The Administration option allows the Administrator to protect their critical data sets. The function of systems administrators, operations control, and z/OS subsystems managers such as CICS, Db2, and others, is to determine which libraries are considered critical.</p>
2	<p>Audit</p> <p>This option reports actions such as ADD, DEL, UPD (update), and REN (rename) that have been recorded for a protected resource. The user specifies a member name or pattern to view the list of actions for a selected resource.</p>
3	<p>Recover</p> <p>This option allows the selective recovery of member backups from the live system.</p> <p>When the Administrator initially protects a resource, all members in that library are backed up with the prefix STC, which indicates that the backup was initiated by the started task. During checkout, another backup is generated with the prefix CKO. When the member is checked in another automatic backup with the prefix CKI is generated. Every time a member of a protected resource changes, the changed member is backed up.</p>
4	<p>Disaster Recovery</p> <p>Select this option to perform the DR functions. Comparison of DR software with the PROD software, as well as regularly synchronizing the DR software with the PROD software is performed by selecting this option.</p>

Table 1. Primary menu options (continued)	
Option	Description
5	Utilities This option contains a set of functions, including <ul style="list-style-type: none"> • Software environment comparison • Reconciling pairs of libraries • Tokenizing software environments • Change identification in pre-tokenized environments

z/OS Change Tracker options

You can use the Options and Parameters panel to customize reporting options and library settings.

Options and Parameters

Command ==>

Specify your JCL library, report and submit options, and JOBCARD. Press Enter to save.

JCL Library Name 'JCLUSER.CYG.JCL'

Enter "/" to select option
Display report panel
Submit job automatically

Job Statement Information:

```

==> //JCLUSERX JOB (0),'',
==> //      CLASS=A,NOTIFY=&SYSUID,MSGCLASS=X
==> //*
==> //*

```

Parameters in Effect:

Parameter Data Set	=> SYS1.PARMLIB.SCB3.USER
Parameter Member	=> CYGPAMS
Control File	=> CYG.V100.BLD00017.CONTROL
Data File	=> CYG.V100.BLD00017.DATA
Archive Control File	=> CYG.V100.BLD00017.ARCHIVE.CONTROL
Archive Data	=> CYG.V100.BLD00017.ARCHIVE.DATA
LINKLIB (APF)	=> CYG.V100.BLD00017.SCYGLNK
Print Utility	=> VPSPRINT
USS Token File HLQ	=> CYGUSS
USS Target Dir	=> /u/changetracker
Options Settings	=> MASSCOPY
Panel National Lang	=> ENU
Email File Data Set	=>

Figure 3. Options and Parameters

The section titled **Parameters in Effect** is a scrollable table which can be navigated using your UP and DOWN commands.

Table 2. Description of input fields in Options and Parameters panels	
Field	Description
JCL library	The JCL library where the z/OS Change Tracker generated batch jobs are stored.

Table 2. Description of input fields in Options and Parameters panels (continued)	
Field	Description
Display report panel	Specify if ISPF generated reports should be printed or discarded. Select the option to display the print panel for reports. If the option is not selected, the reports will be deleted.
Submit job automatically	Select this option to submit generated batch jobs automatically. Leave this option blank to review batch jobs before submitting them.
Job statement information	Four lines are provided for specifying the jobcard. These four lines will be recorded in the ISPF batch jobs generated by z/OS Change Tracker.

Administration

The z/OS Change Tracker Administration menu features reporting options and resource management functions.

One of the primary administrative tasks is resource protection.

When a resource is protected, an initial backup of all members in the library is taken. This backup is recoverable. This feature can be skipped by specifying INITIAL=NO.

Protection provides a locking mechanism where all members in a library are locked. If LOCK=YES is specified during the protection, an update is disallowed on all the members unless checkout standards are used. To unlock members of the resource, use LOCK=NO. This setting allows updates to the members. Whenever a protected member is changed it is automatically backed up, irrespective of the LOCK value.

Administration	
Option ==>	
1 Check Out	Grant ownership of a set of members
2 Check In	Check in members previously checked out
3 Protect a Resource	Protect a PDS or PDSE
4 Modify Protection	Change the LOCK status of a resource
5 Remove Protection	Remove protection from a resource
6 Backup	Backup members of a protected resource
7 Delete or Archive Backups	Delete or archive member backups
8 Checked Out Resources	Display checked out members
9 Show Archive Reports	Show archived backups or actions
10 Show Logs	Show logged Administrator activities
11 Recover Archived Backups	Recover an archived member backup
12 Prepare Tokens	Reference Tokens to identify MODS
13 Identify MODS	MODS since Reference Tokens generated

Figure 4. Administration menu

Table 3. z/OS Change Tracker administration menu options	
Option	Description
1	<p>Check Out</p> <p>This option allows you to check out one or multiple members of a protected resource to a particular user, or a RACF group. At the time of checkout, update rights are granted to the designated user or group, and a backup of the member is generated. You can provide a comment explaining the reason for the checkout.</p>
2	<p>Check In</p> <p>This option allows an administrator to check in members that have been checked out. Checking in a member removes the user's update rights granted by Check Out.</p>

Table 3. z/OS Change Tracker administration menu options (continued)

Option	Description
3	<p>Protect a Resource</p> <p>Use this option to protect data sets (resources). Critical systems libraries such as PARMLIB, PROCLIB, and Db2-related or CICS-related libraries are good candidates for protection. If a resource is protected with LOCK=NO, the resource remains available for anyone to update.</p> <p>If the parameter INITIAL=YES is specified, a system-generated marker of SYS is stored in the repository along with the member backups, so that during recovery of a member, you can recognize the base version created when resource protection was enabled.</p>
4	<p>Modify Protection</p> <p>This option allows you to modify the initially defined protection attributes. For example, the Lock status can be changed from LOCK=YES to LOCK=NO.</p>
5	<p>Remove Protection</p> <p>This option lists all data sets that are currently protected, so that a specific data set can be selected to remove its protection. Backups for the data set can be either deleted or archived. (Archiving moves all of the backups from the live system to the archive system).</p>
6	<p>Backup</p> <p>This option allows you to initiate selective member backups of any members of a protected resource at any time to establish a personal reference copy. At the time of member backup, a comment may be supplied to help identify the version among other versions created by the started task or by CHECKOUT and CHECKIN.</p>
7	<p>Delete or Archive Backups</p> <p>Using this option, you can choose to DELETE or ARCHIVE one or multiple backup versions from the list of member versions presented online.</p>
8	<p>Checked Out Resources</p> <p>This option lists all of the currently protected data sets. Those with checked out members can be selected to display the list of checked out members.</p>
9	<p>Show Archive Reports</p> <p>This option provides a list of all protected resources currently residing in the z/OS Change Tracker Archival system. By selecting a resource from the list, all the archived backups or archived actions associated with the selected resource will be displayed.</p>
10	<p>Show Logs</p> <p>This function displays comments stored in the control file when various activities are performed on a resource, such as applying or removing protection and checking out or checking in a member. These comments are stored for auditing purposes.</p>
11	<p>Recover Archived Backups</p> <p>Displays versions of member backups stored in the z/OS Change Tracker archive files. The Administrator can view the contents of these versions, compare versions, and recover a version to a dynamically allocated recovery data set.</p>

Table 3. z/OS Change Tracker administration menu options (continued)	
Option	Description
12	<p>Prepare Tokens</p> <p>This utility tokenizes all members in one or more partitioned data sets (source/load, PDS, or PDSE). When a partitioned data set is tokenized, a hash token value is created for each member and stored in the control file as a reference token.</p>
13	<p>Identify MODS</p> <p>This option uses the previously created reference tokens to show any changes in the data set members.</p>
X	<p>Exit</p> <p>This option exits the Administration menu.</p>

Check out

This option allows an administrator to check out one or more members of a locked resource to a particular user or a RACF group. When a checkout is done, exclusive ownership rights are granted to the designated user. The member being checked out is backed up to establish a reference copy at the time of check out.

The Administrator can supply a comment explaining the reason for the checkout. The supplied comment is automatically associated with the created version along with the marker CKO: appended to the beginning of the supplied comment. During an online recovery, the visible marker and the comment help users to identify the version as generated by the Check Out function.

When a data set has been locked with LOCK=YES, none of the members may be updated by any user. Members can be checked out to a user or RACF group to enable them to be updated by that user or any member of that group.

Figure 5 on page 29 shows the initial Check Out panel, which you can use to select the data set containing the member to check out.

Command ==>

Check Out (1 of 2)

Scroll ==> CSR

Job Name CHECKOUT

S Protected Resource	Volser	Protection Attributes	Check Out Y/N	Count
- CYGUSER.TEST.PRODUCT.CNTL	-	- - L - -	N	-
- CYGUSER.TEST.PRODUCT.INSTALL	-	- - - - -	N	-
- CYGUSER.TEST.PRODUCT.JCL	-	- - L - -	N	-
- CYGUSER.TEST.PRODUCT.LINKLIB	-	- - - - -	N	-
- CYGUSER.TEST.PRODUCT.LINKLIB.DIST	-	- - - - -	N	-
- CYGUSER.TEST.PRODUCT.MAC	-	- - - - -	N	-
- CYGUSER.TEST.PRODUCT.PANELS	-	- - - - -	N	-
- CYGUSER.TEST.PRODUCT.PARMLIB	-	- - - - -	N	-
- CYGUSER.TEST.PRODUCT.SOURCE	-	- - L - -	N	-

Figure 5. First Check Out panel

A second panel displays a list of the members in the selected data set. You can select one or more members to check out.

Command ==>

Check Out (2 of 2)

Scroll ==> CSR

Selected Resource . . : CYGUSER.TEST.PRODUCT.INSTALL

Checker User ID . . . _____ or RACF Group _____

Comment _____

S	Name	VV	MM	Created	Changed	Size	Init	Mod	ID
-	SYCAASM	26.02	18/08/08	20/10/29	14:38	55	55	0	CYGUSER
-	SYCACOB	26.00	18/08/08	18/08/08	21:17	404	404	0	SCM4226
-	SYCALLOC	01.03	15/06/18	22/09/02	10:50	56	56	0	CYGUSER
-	SYCARC	48.03	15/06/18	15/06/18	13:05	49	49	0	SCM4200
-	SYCCHECK	01.03	15/06/18	15/06/18	13:05	33	33	0	SCM4200

Figure 6. Second Check Out panel

Press Enter and z/OS Change Tracker generates a batch job for performing the checkout. The batch job makes use of the CHECKOUT command. Refer to the [“CHECKOUT command” on page 81](#) for details of using this function in batch mode.

Note: This procedure allows multiple members to be checked out to one specific user. Other users cannot make changes to a checked out member until it is checked in.

Check in

Administrators use the Check In panel to check in members that have previously been checked out.

Checking in a member removes the checked out user's exclusive ownership, allowing it to be checked out to other users.

Command ==>

Check In (1 of 2)

Scroll ==> CSR

Job Name CHECKIN

S	Protected Resource	Volume	Protection Attributes	Check Out Y/N	Count
-	DEV.PRODUCT.INSTALL	-	- - - I	N	-
-	DEV.PRODUCT.LINKLIB	-	- - - I	N	-
-	DEV.PRODUCT.MAC	-	- - - I	N	-
-	DEV.PRODUCT.PANELS	-	- - - I	N	-
-	DEV.PRODUCT.PAMRLIB	-	- - - I	N	-
-	DEV.PRODUCT.SOURCE	-	- - - I	N	-

Figure 7. Check in a member (1 of 2)

The check in process

The Check In panel shown above provides an overview of all the protected resources in the system. Only resources that have some members already checked out are candidates for a CHECKIN operation.

If Y is specified under the column heading Check Out, that indicates that there are some members checked out from the resource. The Count field indicates how many members have been checked out. Only resources with checked out members may be selected for a checkin.

The following table describes the protection attributes and Check Out fields that are displayed for each protected resource.

Table 4. Description of output fields	
Field	Description
Protected Resource	Identifies protected resources that are currently defined to z/OS Change Tracker.
Protection Attributes	A (Alert), N (Notify), L (Locked), and I (Initial)
Check Out: Y/N	Indicates whether any members are checked out from the protected resource.
Check Out: Count	Indicates how many members remain checked out from the protected resource.

Selecting members for check in

On the first check in panel, the Administrator selects the last resource on the panel. The following panel is displayed where the two checked out members are displayed. On this panel, one or more members can be selected for check in.

Command ==>

Check In (2 of 2)

Scroll ==> CSR

Selected Resource

Volume Serial . . :

Comment

Checked Out Member

Checker ID

S CYGMAIN IBMUSER

- CYGMOD IBMUSER

Figure 8. Check in a member (2 of 2)

Comment

It is recommended that the Administrator provides a descriptive comment. The check-in comment is stored with the member backup in the z/OS Change Tracker repository, providing easy identification of the version among all versions created by the STC. The comment is prefixed by the marker CKI to identify the version as the one created at check in.

Output fields

In the second Check In panel, the Checked Out Member column lists members that are currently checked out from the resource. These are the members which can be checked in. In this example, two members CYGMAIN and CYGMOD are candidates for check-in. They are checked out to the user whose ID is IBMUSER.

Automatic backup at the time of checkin

When a member is checked in, a backup of that member is generated in the repository. Along with the member backup, a description containing the special marker CKI is stored to indicate that this backup was caused by CHECKIN. Later, during selective member recovery, the CKI marker helps identify the backup member as the one created at the time of check in.

When the user presses Enter, a job is generated. When this job is submitted, it checks in the member CYGMAIN and generates a backup of the member. The following table describes the output fields.

Table 5. Description of fields	
Field	Description
Selected Resource	The selected data set.
CHECKIN Comment	A description or note about the member that is being checked in.
Name	Members that are currently checked out from the selected resource.
Checker User-ID	The TSO ID of the user who checked out the member.

Protect a resource

Protecting resources (PDS, PDSE, source or load) is one of the main reasons for implementing z/OS Change Tracker. Specific critical libraries, which the health of the z/OS system depends on, can be protected by z/OS Change Tracker. The list of such data sets is determined by operations control and systems managers.

Libraries such as PARMLIB, PROCLIB, VTAMLIB, vendor software, and your subsystems are good candidates for protection by z/OS Change Tracker. After a library is protected, the started task monitors

it continuously for any updates introduced in batch or ISPF. Each time a member of a protected resource is updated, an event is generated indicating who changed the member, what has changed, and the date and time of the change. The started task generates a version of the updated member in the z/OS Change Tracker repository, which provides the ability to revert to this version in future.

See “[PROTECT command](#)” on page 105 for details of this function in batch mode.

After selecting Protect a Resource, you must indicate if the data set to be protected is Cataloged or Uncataloged.

Command ==>		Protect (1 of 3)	Scroll ==> CSR
Cataloged or uncataloged data set			
1	1. Cataloged		
	2. Uncataloged		

Figure 9. Protect a resource (1 of 3)

z/OS Change Tracker requires the explicit name or pattern of the data sets to be protected. You may exclude data sets returned in this search from protection.

Command ==>		Protect (2 of 3)	Scroll ==> CSR
Specific DSN or Dsname Level			
-	DEV.PRODUCT		

Figure 10. Protect a resource (2 of 3)

Command ==>		Protect (3 of 3)	Scroll ==> CSR
Protection Options:			
Enter "/" to select option		Locking	
-	CSAVE Notify at Edit	1	1. None
		2	2. Lock
Email Alert NONE			
Processing Options:			
Comment . . .			
Job Name PROTECT			
Backup Options:			
Backups		Keep backups	
1	1. Suppress initial versions	1	1. Max copies 30
	2. Generate initial versions	2	2. Max days 0
	3. None	3	3. Always
X	Data Set Name	Prompt	
-	DEV.PRODUCT.SOURCE		

Figure 11. Protect a resource (3 of 3)

CSAVE Notify at Edit

An up-front notification for data sets may be requested by indicating YES for this field.

For specific critical libraries (such as SYS1.PROCLIB or SYS1.PARMLIB) you may enforce documentation requirements for changes being introduced. This facility enables users to be notified at the time of editing that the data set is marked as CSAVE enforced, and changes to the members of the edited PDS should be documented using the CSAVE command.

CYG8803W You are about to EDIT/BROWSE a Protected library. For EDIT use the CSAVE command.

Locking

This input field allows the Administrator to protect a resource.

Enter 1 to unlock all the members. This means that any member can be updated, but changes will be monitored.

Enter 2 to lock all the members. This means that no member can be updated unless it is checked out.

If a member is checked out it can only be updated by the user or group to which it is checked out.

Email Alert

Critical libraries can optionally be subject to email alert when protected. If selected, recipients designated during the z/OSMF Configuration workflow are notified via email when protected libraries are modified.

Email notifications must be enabled as part of the z/OS Change Tracker z/OSMF Configuration to function.

Comment

You can enter a comment when protecting a resource. These comments are stored in the control file and can be viewed by using the Show Logs function or the batch reports.

All comments must start in column 1.

Backups

Use this field to specify when z/OS Change Tracker should back up members in the protected resource.

1 Suppress initial versions

Back up members whenever their content changes. Do not take an initial set of backups.

Tip: This option is recommended for large data sets with hundreds or thousands of members.

2 Generate initial versions

Generate an initial set of backups for all members when the library is protected. These are the base versions. Also back up members whenever their content changes.

3 None

Never back up members. This option records changes in the system log without generating any backups.

Keep backups

For this input field the user must specify either a positive integer value or zero. A positive integer value represents the maximum number of backup versions that the user wishes to be kept in the live system before the additional versions are archived by the [AUTOARC](#) function. Hence, if the number of backup versions for a given member exceeds this positive integer value, the AUTOARC command (a weekly batch job) archives the extra versions to the z/OS Change Tracker archive system. If zero is specified, there will be no limit on the number of backup versions stored in the live system. In the example above, we see that the user has specified 30. Once the number of current versions exceeds 30, the extra versions, from the bottom of the list, will be archived.

Days to Keep

Days to keep versions when AUTOARC is run: For this field, the user must specify either a positive integer value or zero. A positive integer value indicates the maximum number of days that the user wishes to keep the versions in the live system. When the age of a backup version exceeds the specified age, that member becomes a candidate for z/OS Change Tracker archival.

If zero is specified, then z/OS Change Tracker will not impose such age requirements. In the above example, days to keep must be zero, as copies to keep is already specified.

Note: The archival rules may be controlled by either copies to keep or days to keep versions. If one has a positive value specified, the other must be set to zero.

Protecting uncataloged resources

z/OS Change Tracker can also protect uncataloged data sets. If Uncataloged is selected on the initial Protect panel, an alternative panel is displayed. You can enter the volume serial and the data set name to identify an uncataloged data set for protection.

```

                                Protect (3 of 3)

Command ==>

ISPF Library:
  Project . . . _____
  Group   . . . _____ . . . _____ . . . _____
  Type    . . . _____
Other Partitioned Data Set:
  Data Set Name _____
  Volume Serial _____ (required)

Protection Options:
  Enter "/" to select option          Locking
  _ CSAVE Notify at Edit              1 1. None
                                       2. Lock
  Email Alert NONE

Processing Options:
  Comment . . _____
  Job Name   PROTECT

Backup Options:
  Backups                                Keep backups
  1 1. Suppress initial versions      1 1. Max copies 30
    2. Generate initial versions        2. Max days 0
    3. None                             3. Always

```

Figure 12. Protecting an uncataloged data set

Note: The generated batch job for the protection of an uncataloged data set can be tailored to include multiple uncataloged data sets for protection.

Modify protection

The z/OS Change Tracker Administrator can use the modify protection function to change the protection status and auto archive policies of one or more selected resources.

For example, if a particular resource no longer needs protection from updates, Locking can be set to None.

Note that when Locking is set to None, the started task still monitors the resource, records all actions on its members, and backs up the members each time they are updated using a batch job or ISPF.

Select **Modify Protection** to display the following panel.

Command ==>

Modify Protection

Scroll ==> CSR

Protection Options:

Email Alert NONE

Enter "/" to select option

- CSAVE Notify at Edit

Locking

2 1. None

2. Lock

Backup Options:

Keep backups

1 1. Max copies 30

2. Max days 0

3. Always

Processing Options:

Comment . . .

Job Name REPROTCT

	Protected Resource	Volume	Protection Attributes	Check Out Y/N	Count
S	DEV.PRODUCT.INSTALL	-	- - - I	N	-
-	DEV.PRODUCT.LINKLIB	-	- - - I	N	-
S	DEV.PRODUCT.MAC	-	- - - I	N	-
-	DEV.PRODUCT.PANELS	-	- - - I	N	-
-	DEV.PRODUCT.PARMLIB	-	- - - I	N	-
-	DEV.PRODUCT.SOURCE	-	- - L I	Y	002

Figure 13. Modify resource protection (LOCK Status)

In this example, the resource DEV.PRODUCT.MAC is selected. The third attribute in the Protection Attributes section is not set, which indicates that this resource is not currently locked. The Locking field has been set to 2, which indicates a request to lock the resource.

Pressing Enter generates a batch job that applies the changes to the selected resource. Within one minute of job completion, the protection list is refreshed automatically and the new LOCK setting takes effect.

The output fields are discussed in [Table 4 on page 30](#).

Remove protection

The z/OS Change Tracker Administrator can use this option to remove protection from a resource.

Before the resource is removed from the protection list, the user can choose whether to delete or archive member backups for that resource. Deletion permanently removes the backups from the live system. Archiving moves all of the backups and the STC-recorded actions from the live system to the archive system.

Select Remove Protection from the Administration menu to display the following panel.

Command ==>

Remove Protection

Scroll ==> CSR

Comment

Backups

1 1. Archive

2. Delete

	Protected Resource	Volume	Protection Attributes	Check Out Y/N	Count
-	DEV.PRODUCT.INSTALL	-	- - - I	N	-
-	DEV.PRODUCT.LINKLIB	-	- - - I	N	-
S	DEV.PRODUCT.MAC	-	- - - I	N	-
-	DEV.PRODUCT.PANELS	-	- - - I	N	-
-	DEV.PRODUCT.PARMLIB	-	- - - I	N	-
-	DEV.PRODUCT.SOURCE	-	- - - I	Y	002

Figure 14. Remove resource protection

In the above example, the Administrator has decided to remove the protection from DEV.PRODUCT.MAC library. Note that the requested operation is archive. Once this operation completes, the user will no longer be able to recover members from the live system. Instead, they will need to use [recover archived backups](#) from the Administration menu if they need to recover any archived versions.

Pressing Enter generates a batch job that applies the changes to the selected resource. Within one minute of job completion, the protection list is refreshed automatically and the resource will no longer be monitored by the started task.

Note: In [Modify Protection](#) option, the resource library has two checked out members and protection cannot be removed until the members are checked in.

Backup

The Backup feature enables the user to back up a member of a protected resource to establish a checkpoint version in the z/OS Change Tracker repository.

The STC automatically takes backups whenever members are changed. This option allows additional backups to be taken at any time. For these backups, a comment can be supplied to differentiate them from the automatic STC backups.

Use the initial backup panel to select the resource to back up.

Backup Members - Protected Resource List										Scroll ==> CSR	
Command ==>											
Job Name BACKUP											
	Protected Resource				Volume	Protection Attributes			Check Out		
									Y/N	Count	
-											
S											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-											
-							</				

Figure 15. Backup members - protected resource list

After a resource is selected, a second panel displays a list of the members in the selected resource. You can select one or more members to back up.

SCB2 CYGCBKPS		Backup Members - Member List					Row 1 to 7 of 62	
Command ==>							Scroll ==> CSR	
Comment _____								
S	Name	VV	MM	Created	Changed	Size	Init	Mod ID
-	CYGCAPP	36.02	15/06/18	22/08/10	12:44	98	98	0 JENNICK
-	CYGCBKP	10.00	16/09/09	16/09/09	17:25	74	74	0 USR4210
-	CYGCBKPL	01.00	15/06/18	15/06/18	13:06	48	48	0 USR4200
-	CYGCBKPS	01.00	15/06/18	15/06/18	13:06	104	104	0 USR4200
-	CYGCBUND	01.00	15/06/18	15/06/18	13:06	95	95	0 SCM9200
-	CYGCBUN1	52.00	15/06/18	15/06/18	13:06	39	39	0 SCM9200
-	CYGCBUN2	01.00	15/06/18	15/06/18	13:06	38	38	0 SCM9200

Figure 16. Second Backup panel

Documenting the user created backup

A comment can be entered in the Comment field, which is stored in the control file associated with the backup. This comment helps to identify the backup during member recovery. See [“Recover”](#) on page 46 for more information.

Delete or archive backups

Using this option, the z/OS Change Tracker Administrator can delete or archive selected member versions from the live system.

Upon selection of the delete or archive backups function a list of protected resources is displayed as shown below. The Administrator must first select the resource containing the member which they wish to delete or archive. In this example, the Administrator has selected the resource DEV.PRODUCT.MAC.

Command ==>

Delete or Archive Backups (1 of 2)

Scroll ==> CSR

Protected Resource	Volume	Protection Attributes	Check Out Y/N	Count
- DEV.PRODUCT.INSTALL	-	- - - I	N	-
- DEV.PRODUCT.LINKLIB	-	- - - I	N	-
<u>S</u> DEV.PRODUCT.MAC	-	- - - I	N	-
- DEV.PRODUCT.PANELS	-	- - - I	N	-
- DEV.PRODUCT.PARMLIB	-	- - - I	N	-
- DEV.PRODUCT.SOURCE	-	- - - I	Y	002

Figure 17. Delete or archive backups (1 of 2)

When the user presses Enter the following panel is displayed.

Command ==>

Delete or Archive Backups (2 of 2)

Protected Resource	: DEV.PRODUCT.MAC
Volume	: - ("-" indicates cataloged)
Member	: CYGARC-----

Backups

1. Archive

2. Delete, comment Archive a version-----

Figure 18. Delete or archive backups (2 of 2)

By specifying an explicit member on the previous panel, z/OS Change Tracker displays all the backup versions currently existing in the live system for the member CYGARC. On this panel, one or multiple versions may be selected. As a result, a job is created to DELETE or ARCHIVE the selected versions.

Command ==>

List of Backups for the Member (CYGARC)

Select one or more versions to ARCHIVE, then press Enter.

Job name ARCHIVE

Mem/Ver	BackupDate	Time	Changed	UserID	Backup Description
<u>S</u> CYGARC	2021/08/30	04:13	21/08/30	USER12	STC: -0TH IKJEFT01 USER12
<u>S</u> -1	2021/08/30	04:11	21/08/30	USER12	STC: -0TH IKJEFT01 USER12

Figure 19. List of backups for the member

Checked out resources

When the z/OS Change Tracker Administrator needs to identify all the protected resources in the system along with their status, the report on resources option is invoked.

The Administrator can determine whether or not any members are checked out by using this option. Selecting this option from the Administration Menu displays the Checked Out Resources panel.

Command ==>		Checked Out Resources				Scroll ==> CSR	
		Volume	Protection Attributes			Check Out Y/N	Count
S	Protected Resource						
-	DEV.PRODUCT.INSTALL	-	-	-	I	N	-
-	DEV.PRODUCT.LINKLIB	-	-	-	I	N	-
-	DEV.PRODUCT.MAC	-	-	-	I	N	-
-	DEV.PRODUCT.PANELS	-	-	-	I	N	-
-	DEV.PRODUCT.PARMLIB	-	-	-	I	N	-
S	DEV.PRODUCT.SOURCE	-	-	-	I	Y	002

Figure 20. Report on resources (1 of 2)

The initial panel identifies all protected resources along with their checkout status. The protection attributes and Check Out fields are discussed in [Table 4 on page 30](#).

Display of checked out members for a selected resource

On the above panel, the Administrator can press Enter to display the status of all members. Alternatively, the Administrator can select a particular resource to display a list of checked-out members and the ID of the user who checked them out.

Command ==>		Checked Out Members		Scroll ==> CSR	
Selected Resource DEV.DEMO.INSTALL					
Checked Out Member		Checker ID	Check Out Comment		
CYGMAIN		IBMUSER	CKO: Checkout for exclusive cont		
CYGMOD		IBMUSER	CKO: Checkout for exclusive cont		

Figure 21. Report on resources (2 of 2)

Table 6. Description of output fields	
Input	Output Field
Selected Resource	The protected resource that was selected from the first panel. A protected resource can have one or more members checked out.
Checked Out Member	A member that is checked out from the protected resource.
Checker ID	The user that the member is checked out to.
Check Out Comment	A comment supplied by the administrator when the member was checked out.

Report the status of all resources (without selecting a particular resource)

If the Administrator presses Enter on the initial panel without selecting a particular resource, the Checked-Out-Components report is generated for all the resources. This report provides a convenient way to gather detailed information about all of the protected resources defined to z/OS Change Tracker without having to select them one by one. The batch command [SHOW=UNIVERSE](#) provides another way to create the same report.

Protected-Resources				Checked-Out-Components			2022/09/07 11:38:58	
CPY	DAY	ATTRIBUTES	Cat	Member	Checker	R	Volser	DSN
10	0	- - L I -	C	SYCACOB	JENNICK	-	WORK01	DEV.PRODUCT.INSTALL
10	0	A N L I -	C	-	-	-	WORK01	DEV.PRODUCT.LINKLIB
30	0	- - - I -	C	-	-	-	WORK01	DEV.PRODUCT.MAC
10	0	- - - I -	C	E	-	-	WORK01	DEV.PRODUCT.PANELS
30	0	- N L - -	C	DEM0DEL	IBMUSER	-	WORK01	DEV.PRODUCT.PARMLIB
30	0	- - L - -	C	MYMEM	XOR2	-	WORK01	DEV.PRODUCT.SOURCE

Legend:
A(Alert) N(Notify/CSAVE) L(Lock) I(Initial) V(Never) Q(Quick)

Figure 22. Example of Checked-Out-Components report

Show archive reports

This option enables the z/OS Change Tracker Administrator to view the archived backups or the archived actions for protected resources.

Select Show Archive Reports from the Administration menu.

Show the archived backups

From the Show Archive Reports panel you can select whether to report on archived backups or archived actions.

```

Command ==>                               Show Archive Reports

Make a selection below and then press Enter.

Type of report to show
1  1. Backups report
   2. Actions report

```

Figure 23. Show archived backups (1 of 3)

Select the Backups report, and then press Enter to display the Show Archived Backups panel.

```

Command ==>                               Show Archived Backups                               Row 1 to 2 of 2
                                                                 Scroll ==> CSR

Member  . . . *_____ (Name or pattern)

S  Archived Protected Resources                Volser
-  DEV.PRODUCT.INSTALL                        -
-  DEV.PRODUCT.PANELS                        -

```

Figure 24. Show archived backups (2 of 3)

On this panel, specify a member name or a member pattern using an asterisk (*) as a wildcard. Type S next to the name of a protected resource to view a list of all archived backups for that resource.

Press Enter to generate a list of all archived versions for matching members.

```

Archives for: VOL= catlg DSN=DEV.PRODUCT.INSTALL
Name/Ver  VV MM  Created      Changed      Size  Init  Mod   ID      Backup-Comment
CYGARC    01.01 21/03/10 22/07/01 19:57 39   38   0   IBMUSER -
-1        01.02 21/03/10 22/08/24 04:54 40   38   0   WSER12 -
-2        01.03 21/03/10 22/08/24 04:55 41   38   0   WSER12 -

```

Figure 25. Show archived backups (3 of 3)

Show the archived actions

If the Actions report is selected on the first panel, the following panel is displayed.

```

                                Show Archived Actions
Command ==>                               Scroll ==> CSR
Member . . . cygarc (name or pattern)

Actions
1 1. ALL
  2. ADD
  3. UPD
  4. DEL
  5. REN

S Archived Protected Resources          Volser
S DEV.PRODUCT.INSTALL                  -

```

Figure 26. Show archived actions (1 of 2)

Upon pressing Enter, the following online report is displayed to the user, with information on all archived Actions for the specified member.

Member	OPTIONS=	ALL:	Archived Actions	for:	VOL=	catlg	DSN=DEV.PRODUCT.INST
	By-User	ACT	Action-Date-Time	SYS/Job	Prog/Job2		Backup-Comment.....
CYGARC	WSER12	UPD	2021/08/24 04:54	P390		IJKEFT01	-
CYGARC	WSER12	UPD	2021/08/24 04:55	P390		IJKEFT01	-

Figure 27. Show archived actions (2 of 2)

As shown above, there were two update activities in the archive system for the member CYGARC.

Show logs

For auditing purposes and increased accountability, z/OS Change Tracker provides the means to display most of the activities performed by administrators.

Log information for the following administrator activities is stored in the control file and can be displayed by using the Show Logs menu option.

Comments are stored for the following functions:

- Protecting a resource, modifying resource protection, or removing protection from a resource
- Deleting or archiving a protected resource
- Checking out or checking in a member
- Deleting a backup version

Select Show Logs from the Administration menu to display the Log of Administrator Activities panel.

```

                                Log of Administrator Activities
Command ==>

ISPF Library:
  Project . . . _____
  Group . . . _____ . . . _____ . . . _____
  Type . . . _____

Other Partitioned Data Set:
  Data Set Name *_____

Reporting Options
Activities of last nnn days ALL

```

Figure 28. Show administrator logs

The administrator can specify the name of a single data set, or else enter an asterisk (*) to display the complete log of activities on all protected resources. The following report shows an example of the type of information that is extracted from the log.

List of TRACE activities 2022/12/08 10:43			
DATE=ALL			
	Action	Resource-name	Volume
	-----	-----	-----
2022/07/12 17:10	PROTECT	SYS1.PARMLIB	-
	Admin:	IBMUSR12 Lock-status: UNLK	
	Comment:	Protection requested by MVS team	
2022/07/14 13:10	PROTECT	SYS1.PROCLIB	-
	Admin:	IBMUSR12 Lock-status: UNLK	
	Comment:	Protection Requested by John	
2022/07/25 15:26	MODIFY	SYS1.PROCLIB	-
	Admin:	IBMUSR13 Lock-status: LOCK	
	Comment:	Change requested to LOCK=YES	
...			

Figure 29. Example of the List of TRACE activities report

In this example the administrator (IBMUSR12) protected two resources (Action=PROTECT) with LOCK=NO (UNLK). The Resource-name field identifies the protected system libraries. The Comment field displays the log message from when the protection was applied.

Recover archived backups

Use the Recover Archived Backups function to restore a prior version of an archived backup.

When a systems administrator needs to revert to an older version of a member and the required version no longer exists in the live system, the older version must be recovered from the z/OS Change Tracker archive.

Selecting Recover Archived Backups from the Administration menu displays the Archived Resources panel.

Command ==>		Archived Resources	Scroll ==> CSR
		Archived Protected Resources	Volume
S		DEV.PRODUCT.INSTALL	-
-		JENNICK.DEV.DEMO.INSTALL	-
-		JENNICK.DEV.DEMO.JCL	-
-		JENNICK.TEST.DEMO.INSTALL	-
-		TSTRAUB.TEST.JCL	-

Figure 30. Recover archived backups (1 of 3): Archived Resources

The Archived Protected Resources field lists all of the protected resources in the archive. These are the resources containing member backup versions that can be recovered from the archive system.

Specifying a member or a member pattern

After selecting an archived resource, use the second panel to enter selection criteria.

```

                                Recover or View Backups
Command ==>

Archived Resource . : DEV.PRODUCT.INSTALL
Volume . . . . . : -             ("- indicates cataloged)

Archived Member Backup cygarc
Job Name . . . . . ARECOVER

```

Figure 31. Recover archived backups (2 of 3): Recover or View Backups

Enter a member name or pattern in the Archived Member Backup field. If an explicit member such as CYGARC is specified, a list of all available versions for that member is displayed from which one or more versions can be selected for recovery. If a pattern such as CYG* is specified, all versions of members whose name starts with CYG are listed. If * is specified, all versions for all members in the archive are listed.

```

                                Archived Backups of Member (CYGARC)
                                SCROLL==> CSR
Command =====>

Select one or multiple versions for RECOVERY.
Lcmds: V (View), E (Edit), C (Changes), S (Recover)

  Mem/Ver  BackupDate Time    Changed  UserID  Backup-Description
-  -      2019/08/22 23:14    21/07/01 IBMUSER SYS:  -Initial Version
-  -1     2019/08/24 04:54    21/08/24 USER12  No comment.
-  -2     2019/08/24 04:55    21/08/24 USER12  No comment.

```

Figure 32. Recover archived backups (3 of 3)

In the above panel, select one or multiple versions to be recovered to a recovery PDS compatible with the original PDS to which this member had belonged.

Line commands: The following line commands are available:

V (View)

View the contents of the version without actually recovering the version.

E (Edit)

Edit the member and save it into a library.

C (Changes)

Enter C next to any two backup versions to view the differences between them.

S (Select for recovery)

Select one or more backup versions for recovery.

Recovery PDS naming convention

The recovery data set is used to store the recovered versions from the archive. These recovered member backups can later be copied to the appropriate libraries as needed. From the list of available backups, when one version is selected for recovery, the recovery data set is allocated dynamically by z/OS Change Tracker, and its name is displayed in an ISPF message. If more than one version is being recovered from the archive system, then a batch job is generated, and in that job is a reference to this already pre-allocated recovery PDS. The recovery PDS has a name in the following form:

```
IBMUSER.RECOVERY.CYG13208.T2218370
```

The first two nodes are the user's ID and the word RECOVERY. The third and fourth nodes are a datestamp in the form CYGYDDDD and a timestamp.

Naming conventions for the recovered versions

Suppose there are a total of three archived backup versions for the member CYGARC. When these versions are recovered, version 0 is recovered under its original name. The other versions use generated names as shown in this example:

```
CYGARC
ZZZZV001
ZZZZV002
```

Prepare tokens

The Prepare Tokens utility tokenizes all members that are stored in one or more partitioned data sets (source/load, PDS or PDSE). When a partitioned data set is tokenized, a hash token value is created for each member and stored in the control file as a reference token.

Preparing the Tokens

In the first Tokenize panel, specify whether the data set is cataloged or uncataloged.

In the Prepare Tokens panel, specify one or more library names or patterns. Matching libraries will be resolved from the system catalog and tokenized.

Prepare Tokens (2 of 3)

Command ==>

To Prepare Tokens enter one or more specific DSNs or dsname levels on each line, then press ENTER. DSNs can be PS, DA, PDS, PDSE (source or load).

Specific DSN or Dsname Level

- DEV.PRODUCT

-

-

Figure 33. Prepare tokens (2 of 3)

In this example, the user has specified the pattern DEV.PRODUCT as the data sets to resolve from the catalog.

Uncataloged data sets are also supported. For a full description on how to tailor the batch job for tokenizing uncataloged data sets, refer to [“TOKEN command” on page 128](#).

Data sets resolved from the catalog

The next panel lists the matching data sets. You can review the list and use the X line command to exclude data sets that are not to be tokenized. Press Enter to generate the batch job for submission.

Prepare Tokens (3 of 3)

Command ==>

Specify the comment and the JOBNAME. Undesired data sets may be excluded from the list using the line command X. Press ENTER to build the JCL for submission. Tokens are stored in the Control File.

Comment for tokenize Tokenize the PROD Software Environment

Job Name PREPARE

X Data Sets Resolved from the Catalog (3.4)

- DEV.PRODUCT.INSTALL

- DEV.PRODUCT.LINKLIB

- DEV.PRODUCT.MAC

- DEV.PRODUCT.PANELS

- DEV.PRODUCT.PARMLIB

- DEV.PRODUCT.SOURCE

Figure 34. Prepare tokens (3 of 3)

```
LOGCOMMENT='Tokenize the DEV environment'
*
TOKEN=(DSN=DEV.PRODUCT.INSTALL)
TOKEN=(DSN=DEV.PRODUCT.LINKLIB)
TOKEN=(DSN=DEV.PRODUCT.MAC)
TOKEN=(DSN=DEV.PRODUCT.PANELS)
TOKEN=(DSN=DEV.PRODUCT.PARMLIB)
TOKEN=(DSN=DEV.PRODUCT.SOURCE)
```

Figure 35. Tokenization JCL

A batch job containing the above statements is created. When this job is submitted, the TOKEN commands tokenize each data set. One token is generated per member and the tokens, along with some other information, are stored in the control file for future reference.

Identify MODS

MODS Identification identifies data set modifications such as member add, delete, update and rename using tokens stored in the repository. It requires that the data sets have been previously tokenized.

To identify the MODS for a group of data sets, first specify if they are cataloged or uncataloged.

```

                                Identify MODS
Command ==>

To identify MODS in a group of pre-Tokenized data sets specify the following
option, then press Enter.

Cataloged or uncataloged data sets
1  1. Cataloged
   2. Uncataloged
```

Figure 36. Identify MODS selection

Next, enter the specific data set names or patterns.

```

                                Identify MODS (1 of 2)
Command ==>

To identify the MODS since the reference tokens were generated, enter one or
more DSNs or Dsname levels on each line, then press Enter.

    Specific DSNs or Dsname Levels
-  DEV.PRODUCT-----
-  -----
-  -----
-  -----
-  -----
-  -----
```

Figure 37. Identify MODS (1 of 2)

The next panel displays the data set names that have been resolved. Names can be excluded from the list by using the X line command.

Select a report option:

MODS

All changes to members.

ADD

Only added members.

DEL

Only deleted members.

UPD

Only updated members.

REN

Only renamed members.

SAME

Matching member names with the same contents.

Enter a name for the batch job being created. The job will be saved in your JCL library under this name.

When Enter is pressed a batch job will be built for you to submit.

```
Identify MODS (2 of 2)

Command ==>

Specify the report option and the jobname. You may exclude some data sets
from the list using the line command X. Press Enter to build the job.
Report Option
1 1. MODS
  2. ADD
  3. DEL
  4. UPD
  5. REN
  6. SAME

Job Name . . SHOWMODS

X Data Sets Resolved from the Catalog (3.4)
- DEV.PRODUCT.INSTALL
- DEV.PRODUCT.LINKLIB
- DEV.PRODUCT.MAC
- DEV.PRODUCT.PANELS
- DEV.PRODUCT.PARMLIB
- DEV.PRODUCT.SOURCE
```

Figure 38. Identify MODS (2 of 2)

Audit

The option is used to display activities on the protected resources monitored by the z/OS Change Tracker STC. When a protected resource is updated, whether in batch or ISPF, the update activity at the member-level is recorded in the z/OS Change Tracker Control File, and the updated member is backed up automatically to the z/OS Change Tracker repository. As a result, member versions are created and stacked for any future online recovery.

The update activities create Actions of type ADD, UPD (update), DEL (delete) and REN (rename). For all action types, except for the DEL action, a version is created by the STC. For the DEL activity, however, only the Action is recorded indicating the date it was deleted, the ID of the TSO user who committed the delete along with the program name, and the job name.

On the first Audit panel, the Auditor specifies a member name or pattern and the type of actions to report, and then selects a protected resource.

The Protection Attributes and other fields are discussed in [Table 4 on page 30](#).

Audit						
Command ==>						
Member CYGSTC_____						
Actions						
1. ALL						
2. ADD						
3. DEL						
4. REN						
5. UPD						
	Protected Resource	Volume	Protection Attributes		Check Out Y/N Count	
S	DEV.ATM.INSTALL	-	-	-	I	N -
-	DEV.ATM.LINKLIB	-	-	-	I	N -
-	DEV.ATM.PANELS	-	-	-	I	N -
-	DEV.ATM.SAMPLES	-	-	-	I	N -

Figure 39. Audit

Pressing Enter generates a SYSPRINT report for online review.

Auditors can request a specific action type to see the Actions for the member CYGARC. If a member pattern such as * is specified, all member-level update activities in the library are reported.

List of Activities in Protected Libraries						
2020/08/30 08:41						
DATE=ALL						
OPTIONS=ALL						
	By-User	Act	Action-Date-Time	SYS/Job	Prob/Job2	Oldname
			DSN=DEV.PRODUCT.INSTALL		-	
CYGARC	WSER12	UPD	2020/08/30 04:13	P390	IKJEFT01	STC: -OTH IKJEFT01
CYGARC	WSER12	UPD	2020/08/30 04:11	P390	IKJEFT01	STC: -UPD IKJEFT01
***** Bottom of Data *****						

Figure 40. Example audit results

Recover

The Recover function is used to recover prior versions of backed-up members from source or load libraries.

There may be incidents where a user has introduced an undesirable change to a critical resource member. In this case, reverting to a prior version is the quickest way to resolve the issue.

Select Recover from the z/OS Change Tracker Primary Option Menu to display the following panel.

Recover (1 of 3)						
Command ==>						
Scroll ==> CSR						
	Protected Resource	Volume	Protection Attributes		Check Out Y/N Count	
-	DEV.PRODUCT.INSTALL	-	-	-	I	N -
-	DEV.PRODUCT.LINKLIB	-	-	-	I	N -
-	DEV.PRODUCT.MAC	-	-	-	I	N -
-	DEV.PRODUCT.PANELS	-	-	-	I	N -
-	DEV.PRODUCT.PAMRLIB	-	-	-	I	N -
-	DEV.PRODUCT.SOURCE	-	-	-	I	Y 002

Figure 41. Recover (1 of 3)

This panel lists the protected resources currently defined in the system. The Protection Attributes and other fields are discussed in [Table 4 on page 30](#).

Specifying members for recovery or listing

After selecting the resource containing the member that you would like to recover, a second recovery panel is displayed to allow for specific selection criteria.

On the second panel, enter a member name or pattern and then select the reporting mode. Select **List backups** to select from a list of matching backup versions for recovery, or select **Detailed report of backups** to generate a SYSPRINT report. In the example below, the member CYGARC has been specified.

```
Recover or View Member Versions Online (2 of 3)
Command ==>
Protected Resource
Volume . . . . . : DEV.PRODUCT.INSTALL      ("- " indicates cataloged)
Member . . . . . : CYGARC

Show versions in the following date range
From . . . . . : (YYYY/MM/DD)
To . . . . . : (YYYY/MM/DD)

Select the mode
- 1. List backups
- 2. Detailed report of backups
```

Figure 42. Recover or view member versions online (2 of 3)

```
Backup Versions of Member (CYGARC)
Command ==>
Change Reporting:
Job Name RECOVER
Enter "/" to select option
- Batch execution
Scroll ==> CSR

Name      Ver  Changed          ID      Backup Comment
CYGARC    0   2020/08/30 04:13  USER12  STC: -OTH IKJEFT01  USER12
S CYGARC  -1   2020/08/30 04:11  USER12  STC: -UPD IKJEFT01  USER12
```

Figure 43. Recover - backup versions of member (3 of 3)

Recovering one or more versions

When an exact member name is specified in the Member field, a list of all backup versions for that member is displayed. One or more versions can be selected from this list for recovery.

Primary commands: The following primary commands are available:

- **LOCATE member** locates a particular member within a long list of members, where the member is not immediately visible.
- **SORT NAME** displays the versions sorted alphabetically by their name.
- **SORT CHANGED** sorts the versions by their changed date.
- **SORT ID** displays versions in order, based on their user ID.

Naming conventions for the recovered versions

Suppose there are two backup versions available for the member called CYGARC. When recovery of both versions is requested, the recovered versions in the recovery PDS will appear as the following:

```
CYGARC, ZZZZV001
```

The last three digits represent the version number for the backup. Therefore, ZZZZV001 represents the -1 backup version of CYGARC. Similarly, the ZZZZV002 represents the -2 backup version, and ZZZZV003 represents the -3 backup version and so on.

Recovery PDS naming convention

The Recovery PDS stores the recovered member backup version. This PDS is compatible with the original PDS to which the member belongs.

When the user selects one member for recovery, the Recovery data set is allocated dynamically by z/OS Change Tracker and displayed to the user online. A message appears on the user's screen to indicate the name of the Recovery data set which contains that single recovered backup.

If the user selects multiple versions for recovery, then a batch job is generated, and in that job is a reference to this already pre-allocated Recovery PDS. The Recovery PDS always has a form similar to the following:

```
IBMUSER.RECOVERY.CYG137040.T2218370
```

The first two nodes are self-documenting. The third node is of the form CYGYYDDD. The YY stands for the last two digits of the year (for example, 2013 would be 13). The last three digits DDD represent the number of calendar days into the year, for that particular day (the day that the recovery is performed). The third node (T2218370) is the time node. In this example, the 22 is the military hour, the 18 is the minutes past the hour, and the 370 is the number of milliseconds into that minute. The combination of all these nodes produces a unique recovery PDS name every time.

Comparing pairs of versions online

When all versions for a specific member are listed, you can use the C line command to select a pair of versions for comparison.

```

Backup Versions of Member (CYGARC)
Command ==>
Scroll ==> CSR

Versions for the Date Range: not specified

Change Reporting:
  Job Name RECOVER

  Enter "/" to select option
  Batch execution

  Name      Ver  Changed          ID      Backup Comment
C  CYGARC   0   2020/08/30 04:13  USER12  STC:  -OTH IKJEFT01  USER12
C  CYGARC  -1   2020/08/30 04:11  USER12  STC:  -UPD IKJEFT01  USER12

```

Figure 44. Comparing pairs of versions online

Line commands: The following line commands are available:

V (View)

View the contents of the version without actually recovering the version.

E (Edit)

Edit the member and save it into a library.

N (Notes)

View all of the notes that were entered for a given backup at the time of CSAVE.

C (Changes)

Enter C next to any two backup versions to view the differences between them.

T (Details)

View the change details introduced in all or selected versions in chronological order.

S (Select for recovery)

Select one or more backup versions for recovery.

Reporting on chronological change details

The z/OS Change Tracker STC automatically creates member versions as the changes are introduced. It is highly desirable to be able to recover the versions collected in a specified date range and then compare those versions in the right order to produce the chronological change details report.

This function is automated in z/OS Change Tracker. From the main menu select option 3 Recovery. On the subsequent panel, select a protected library. On the subsequent panel specify the member name for which the Change Detail is required. Also, specify the date range as shown and press Enter:

```

                                Recover or View Member Versions Online (2 of 3)
Command ==>

Protected Resource
Volume . . . . . : DEV.PRODUCT.INSTALL          ("- " indicates cataloged)
Member  . . . . . : CYGARC

Show versions in the following date range
From   . . . . . : 2020/09/08                  (YYYY/MM/DD)
To     . . . . . : 2020/09/09                  (YYYY/MM/DD)

Select the mode
  1. List backups
  2. Detailed report of backups
```

Figure 45. Chronological change details (1 of 3)

On the above panel, press Enter to see the following panel.

```

                                Backup Versions of Member (*)
Command ==>                                Scroll ==> CSR

Versions for the Date Range: not specified

Change Reporting:
  Job Name RECOVER

  Enter "/" to select option
  Batch execution

Name   Ver   Changed          ID      Backup Comment
C  CYGARC   0   2020/09/09 04:13  USER12  STC:  -0TH IKJEFT01  USER12
C  CYGARC  -1   2020/09/09 04:11  USER12  STC:  -UPD IKJEFT01  USER12
C  CYGARC  -2   2020/09/08 22:54  USER12  STC:  -Initial version
```

Figure 46. Chronological change details (2 of 3)

All collected versions in the specified date range are displayed on the previous panel. Once the panel is displayed, use the line command T for all versions and press Enter. As a result, a job is created which recovers all the versions to a recovery PDS and then compares the recovered versions in the right sequence.

In this example, VER -1 (recovered under ZZZZV001) is compared with VER -2 (recovered under ZZZZV002) and constitutes CHANGE 01, and then VER 0 (recovered under CYGARC) is compared with VER -1 (recovered under ZZZZV001) and constitutes the CHANGE 02. The results of all comparisons are reported in a SYSPRINT file (as MOD).

In the final step of the job, the SYSPRINT file is copied into another SYSPRINT file with the last node of change in the name. Refer to [Figure 47 on page 50](#) for example. Browse the file TSOID.SYSPRINT.CHANGE and search for the string CHANGE: to see the changes in chronological order.

```

-----IBM z/OS Change Tracker - Chronological Change Report-----
-----
Look for "CHANGE:" to review the changes:

===== CHANGE: 001 =====
NEW:  WSER12.RECOVERY.CYG13252.T0155260(ZZZZV001)          OLD: WSER12.RECOVER

                LISTING OUTPUT SECTION (LINE COMPARE)

ID      SOURCE LINES
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----
I  -    ADDED BY CICS TEAM
        ADDED BY CICS TEAM
        ADDED BY CICS TEAM
        ADDED BY CICS TEAM
        ADDED 5TH LINE
        ADDED LINE-1
        ADDED LINE-1
        ADDED LINE-1
        ADDED LINE-1
        /*JOB CARD JOB CARD
***** CHANGE SECTION CUTOFF *****

```

Figure 47. Chronological change details (3 of 3)

Disaster recovery

The disaster recovery option is used to perform the DR-related functions of z/OS Change Tracker. Tokenization and comparison of DR software with PROD software can be performed by selecting this option.

```

                                Disaster Recovery
Option ==>

1  Tokenize your Software      Tokenize PROD or DR Software
2  Report Differences          Use Token files to compare environments

```

Figure 48. Disaster recovery functions

Tokenize your Software

The first step is to tokenize the data sets in each environment. The tokens created will be used to perform the comparison between the two environments.

The Production Software Environment option resolves the data sets of your production software to be tokenized, and must be performed on your production system. The DR Software environment option resolves data sets of your disaster recovery site, and must be run from that site. Select each option in turn to perform the tokenization of that environment.

```

                                Prepare Tokens
Command ==>

To prepare tokens for a software environment, make a selection below, then
press Enter to continue.

Environment to tokenize
- 1. Production Software Environment
- 2. DR Software Environment

```

Figure 49. Prepare tokens

For each environment specify the name of an external token file to be allocated.

Tokenize Production	
Command ==>	
To prepare tokens for your PROD software environment, enter the name of an external token file to be allocated.	
Name of External Token File . .	_____

Figure 50. Tokenize production (1 of 3)

Enter the data set names or HLQs for the environment being tokenized.

Tokenize (2 of 3)		Row 1 to 1 of 1
Command ==>		Scroll ==> CSR
To prepare tokens enter one or multiple specific DSNs or dsname levels on each line, then press Enter. DSNs can be PS, DA, PDS, PDSE (source or load).		
Specific DSNs or Dsname Levels		
CYGUSER.CYG.JCL		

Figure 51. Tokenize (2 of 3)

The next panel displays the data set names that have been resolved. Names can be excluded from the list by using the X line command. Enter a name for the batch job being created and provide a comment. The job will be saved in your JCL library under this name. The comment is logged in the repository for audit purposes. When Enter is pressed a batch job will be built for you to submit.

Tokenize (3 of 3)		Row 1 to 1 of 1
Command ==>		Scroll ==> CSR
Specify comment and the job name. You may exclude data sets from the resolved list using the line command X. Press Enter to build the JCL for submission.		
Comment Tokenize the PROD Software Environment		
Job Name PREPARE		
X	Data Set Name	Prompt
	CYGUSER.CYG.JCL	

Figure 52. Tokenize (3 of 3)

Report differences

The Report Differences option uses the previously created external token files to compare the production and DR environments and report the differences.

Specify the external token files created for your production and DR environments.

Enter a name for the batch job being created. The job will be saved in your JCL library under this name. When Enter is pressed a batch job will be built for you to submit.

Report Differences	
Command ==>	
To report differences between your PROD software and your DR software, specify the external token files for both environments and press Enter to create and submit the job.	
External Token File for PROD . .	
External Token File for DR . . .	
Job Name	

Figure 53. Report differences

Utilities

The z/OS Change Tracker Utilities menu.

This menu features utilities for:

- Comparing data sets and data set members using SuperC.
- Comparing and reconciling data sets using z/OS Change Tracker tokenization.
- Comparing z/OS UNIX directories and files using z/OS Change Tracker tokenization.

Utilities Menu		
Option ==>		
1	SuperC Member Compare	SuperC compare of members in a Library
2	Macro Compare	A high-level compare of two PDS(E)s
3	SuperC Compare	A detailed compare using IBM SuperC
4	Group Compare	High-level compare of pairs of PDS(E)s
5	Reconcile Two Libraries	Reconcile one pair of libraries
6	Reconcile Library Groups	Reconcile pairs of libraries
7	Tokenize Directory Files	List the Directories to select from
8	Report Directory Changes	Use Token File for a Difference Report
9	Macro Compare Files	Comparison of two OpenMVS Directories
10	List a Token File	List the Tokenized Regular files

Figure 54. Utilities menu

SuperC member compare

Select this option to use the SuperC program to compare members in a protected resource.

On the first panel, enter a member name or pattern, or * to match all members. Matching members will be displayed for selection on the Member List panel. Also specify whether batch execution of SuperC is required.

SuperC of Protected Members	
Command ==>	
Member	*
Job Name	SUPERC
Enter "/" to select option	
_	Batch execution

Figure 55. SuperC of protected members

Select a data set from the list of Protected Resources.

Protected Resources		Row 1 to 1 of 1	
Command ==>		Scroll ==> CSR	
(S)elect a resource to view its member list. Then, select any two members for a SuperC compare.			
S Protected Resource	Volume	Protection Attributes	Check Out Y/N Count
_ CXPUSER.CNTL2	-	- - L -	N -

Figure 56. Select a resource for SuperC

The Member List panel displays a list of the matching members in the selected resource. You can select a pair of members to be compared.

Member List										Row 1 to 6 of 6	
Command ==>										Scroll ==> CSR	
Select a pair of members for the Superc Compare.											
S Name	VV MM	Created	Changed	Size	Init	Mod	ID				
- BLDDEVO	01.20	20/10/14	22/04/12 07:21	131	447	131	CYGUSR1				
- BLDDEVT	01.05	20/11/10	22/01/18 14:00	138	130	25	CYGUSR1				
- BLDDEVX	01.17	20/10/14	22/01/18 14:01	130	447	130	CYGUSR1				
- BLDSRC	01.04	20/12/11	22/01/18 14:01	86	85	3	CYGUSR1				
- NEWVER	01.00	22/08/12	22/08/12 11:29	1	1	0	CYGUSR1				
- TSTVER	01.01	22/08/11	22/08/12 08:20	1	1	0	CYGUSR1				

Figure 57. Select members for SuperC comparison

When Enter is pressed SuperC will run in the foreground or a SuperC batch job will be built for you to submit.

Macro compare

Use the Macro Compare utility to perform a high-level comparison of two source or load libraries (PDS or PDSE).

z/OS Change Tracker, in its own style of high-level comparison, identifies the following conditions between two libraries:

- SAME matching member names with same contents
- DIFF Matching member names with different contents
- MATCH member names found in both libraries
- MISMATCH member names existing in one, but missing from the other

When comparing two libraries any or all of these conditions may be requested. When the processing option (ALL) is requested, the generated output will report the cases: SAME, DIFF and MISMATCH, all in a single report.

When the processing option MODS is requested, will report the cases: MISMATCH and DIFF all in a single report, providing a convenient way to focus on only the modifications between the two partitioned data sets.

The Macro Compare utility can also invoke SuperC to verify where changes have been introduced in a selected member.

Upon selection of Macro Compare from the Utilities Menu, two main panels are displayed successively, requesting information about the two data sets that are to be Macro Compared. The following figure shows the initial panel that is displayed to the user.

```

Macro Compare: New Data Set (1 OF 2)

Command ==>

Specify "New" Data Set to be compared, then press the ENTER key.

ISPF Library:
Project ==> DEV
Group ==> PRODUCT ==> ==>
Type ==> PANELS

"New" Other Partitioned or Sequential Data Set:
Data Set Name ==>
Volume Serial ==> (If not cataloged)

Process Option: ==> DIFF DIFF, SAME, MATCH, MISMATCH, MODS, ALL
Consider Directory: ==> N Y/N: Y-reports directory differences also

Enter "/" to select Execution Mode:
/ Foreground
_ Batch

Saved JOB ==> COMPARE (JOB is saved under this name)

```

Figure 58. Macro compare new data set

Notice that on this initial panel, the term New is used to refer to the data set that has changed in some fashion. The process option DIFF has been requested, and Consider Directory has been set to N. When Enter is pressed, the second panel is displayed.

```

Macro Compare: Old Data Set (2 OF 2)

Command ==>

Specify "Old" Data Set to be compared, then press the ENTER key.

ISPF Library:
Project ==> DEV
Group ==> OSECURE ==> ==>
Type ==> PANELS

"Old" Other Partitioned or Sequential Data Set:
Data Set Name ==>
Volume Serial ==> (If not cataloged)

```

Figure 59. Macro compare old data set

On the above panel, the user specifies their old (original) data set which is to be compared against. z/OS Change Tracker then quickly tokenizes both libraries, finds the member-level differences, and categorizes the results in a statistical form as shown below.

```

Statistics of Macro Compare OPTION=DIFF --

*****
After review press ENTER to continue
*****

Date: 2021/02/04 Time: 10:56

The Process completed; Statistics follow:

446 MATCH
445 SAME
1 DIFF
0 MISMATCH in Dsn1
0 MISMATCH in Dsn2

```

Figure 60. Macro compare statistics

As shown above, amongst 446 matched member names, only one member is a DIFF. All others have SAME contents. There are zero MISMATCH counts. By pressing Enter the matched member name with the DIFF content is displayed.

In the below example, one member in both libraries has different tokens and thus reported as DIFF.

```

IBM z/OS Change Tracker Result of Macro Compare: OPTION=DIFF
Command =====>                                SCROLL====> CSR

Select a DIFF member for an online SuperC compare. Or, press END to exit.

Member   Result   DSN1-Update-Date DSN2-Update-Date Dir/Alias
S CYGCAPP DIFF      2021/02/04 10:56  2020/08/02 03:47

```

Figure 61. Macro compare result

As shown above, the member CYGCAPP exists in both libraries: it is matched by name, but their content tokens were different and therefore it has been reported as being a DIFF. By selecting any DIFF member, SuperC is invoked to show the details of changes in the two selected members.

```

ISRSUPC - MVS/PDF FILE/LINE/WORD/BYTE/SFOR COMPARE UTILITY- ISPF FOR z/OS
NEW: DEV.PRODUCT.PANELS(CYGCAPP)                                OLD: DEV.OSECURE.PA

LISTING OUTPUT SECTION (LINE COMPARE)

ID      SOURCE LINES
-----1-----2-----3-----4-----5-----6-----7-----
I - Added a line here
  )ATTR
  )ATTR
    % Type(text) color(white)
    + Type(text) color(blue)
    ) Type(output) intens(low) pad('') caps(off) just(left) color(yellow)
    ( Type(text) color(green)

    @ type(input) intens(high) pad('') caps(on) just(left)
    $ type(input) intens(high) pad('') caps(on) just(left)
    type(input) intens(high) pad('') caps(off) just(left)

```

Figure 62. SuperC Compare Report

The line beginning with I shown above is the inserted line in the new member CYGCAPP.

Several points are discussed below when filling in the information on the panels.

Specification of the Volume Serial

On the first macro compare panel, the user also has the option to specify a volume serial. This volume serial input field is only required if the new data set is not cataloged. If the data set is in fact cataloged, then this input field can be left blank.

Processing options

On the first macro compare panel the user has requested the processing option DIFF which requests for comparing the tokens of the matching members and reporting the ones with different content tokens. Refer to [Table 7 on page 56](#) for more descriptions of the processing options, SAME, DIFF, MATCH, MISMATCH, MODS, and ALL.

Directory Considerations

In [Figure 58 on page 54](#), Consider Directory is set to N. This means that members which have identical content but different ISPF statistics will not be reported as different. If Consider Directory had been set to Y, the online member report would indicate whether the member difference was due to a difference in ISPF statistics (directory difference) or an actual content difference. In the following sample output,

the word DIR in the Dir/Alias column indicates that the member CYGCBKP has identical content in both libraries but the directory information is different.

```

Command ==>          Result of Macro Compare: OPTION=DIFF          Row 1 of 2
                                SCROLL==> CSR
Select a DIFF member for an online SuperC compare. Or, press END to exit.

  Member      Result      DSN1-Update-Date  DSN2-Update-Date  Dir/Alias
-  CYGCAPP     DIFF        2020/09/10 16:14  2020/07/24 03:47
-  CYGCBKP     DIFF        2020/09/11 10:26  2020/09/05 09:47  DIR
***** Bottom of data *****

```

Figure 63. Difference report when “Consider Directory” is requested

Table 7. Description of input fields (CYGUCMP1)	
Field	Description
Data set name	This input field is used to specify the name of the new data set (the partitioned data set that was perhaps changed in some way).
Volume serial	The Volume Serial input field is only required if the data set that was specified for the input field called data set name, is not cataloged. If the specified data set is cataloged, then this field may be left blank.
Process option	<p>This field specifies the type of comparison reporting to perform.</p> <ul style="list-style-type: none"> • SAME: Two members are only considered to be the same (SAME) if they match by name and are exactly identical in content. • DIFF: Members are considered to be different (DIFF) when they match by name but have different content. • MATCH: When (MATCH) processing is requested, members are first matched by name, and then a finer level of comparison reporting is performed to further classify each member pair as being the SAME or DIFF. • MISMATCH: a MISMATCH occurs when a member exists in one data set but not the other. • MODS: The processing option (MODS) reports: DIFF and MISMATCH. • ALL: The processing option (ALL) reports the cases: SAME, DIFF and MISMATCH.
Execution mode	This option specifies whether to perform comparison in the foreground, or to edit and submit a job in the background to receive a batch report. Use the forward slash / character to select either foreground or batch.
Consider directory	This parameter specifies whether to consider ISPF statistics in the comparison report. It is conceivable that two members being compared can have identical content, but their ISPF statistics are different. This can occur if someone opens a member for edit and then saves it without making any changes to the content. In this case, the ISPF statistic Changed, will be different. z/OS Change Tracker can be requested to detect such differences. To report directory differences, the user must specify Y (Yes) in this field. Otherwise specify N (No) to omit ISPF statistics in the comparison.

SuperC compare

Use the SuperC compare function to perform a detailed comparison of any two members of two libraries (PDS or PDSEs).

The following panel is the initial panel displayed.

```

                                SuperC Compare: New Data Set
Command ==>

Specify the "New" Data Set and a specific member. Press Enter for next panel.

ISPF Library:
  Project . . . DEV
  Group . . . . PRODUCT      . . .      . . .      . . .
  Type . . . . PANELS
  Member . . . . CYGCBKP
"New" Other Partitioned or Sequential Data Set:
  Data Set Name
  Volume Serial      (If not cataloged)

Job Name . . . . SUPERC

Enter "/" to select option
_ Batch execution

```

Figure 64. SuperC compare new data set

In the above panel, a library and an explicit member of the library has been specified. By pressing Enter the second panel is displayed.

```

                                SuperC Compare: Old Data Set
Command ==>

Specify the "Old" Data Set and press Enter.

ISPF Library:
  Project ==> DEV
  Group   ==> OSECURE      ==>      ==>      ==>
  Type    ==> PANELS
"Old" Other Partitioned or Sequential Data Set:
  Data Set Name ==>
  Volume Serial ==>      (If not cataloged)

```

Figure 65. SuperC compare old data set

SuperC is invoked to compare the two members from the new library versus the old library.

```

***** Top of Data *****
ISRSUPC - MVS/PDF FILE/LINE/WORD/BYTE/SFOR COMPARE UTILITY- ISPF FOR z/OS
NEW: DEV.PRODUCT.PANELS(CYGCBKP) OLD: DEV.OSECURE.PA

                                LINE COMPARE SUMMARY AND STATISTICS

76 NUMBER OF LINE MATCHES          0 TOTAL CHANGES (PAIRED NONPAIRED
0 REFORMATTED LINES                0 PAIRED CHANGES )REFM-PAIRED INS
0 NEW FILE LINE INSERTIONS         0 NON-PAIRED INSERTS
0 OLD FILE LINE DELETIONS          0 NON-PAIRED DELETES
76 NEW FILE LINES PROCESSED
76 OLD FILE LINES PROCESSED

LISTING-TYPE = CHNG      COMPARE-COLUMNS = 1:72      LONGEST-LINE = 80
PROCESS OPTIONS USED: SEQ(DEFAULT)

***** Bottom of Data *****

```

Figure 66. SuperC compare result

Group compare

The z/OS Change Tracker Group Compare facility enables users to prove or disprove whether or not two software environments are identical. When a user needs to perform a high-level comparison of groups of data sets (PDS or PDSE) to determine their similarities and differences, this function may be used.

Similar to the Macro Compare utility discussed earlier, the Group Compare utility is designed to accomplish the same task, but for groups of data sets. The key point to be emphasized here is that data sets are matched in a pair-wise fashion. For the group compare process to work, you must identify their first set of libraries in the comparison, which are referred to as new. These libraries can be thought of as the set containing updates and changes. z/OS Change Tracker also assists the user in defining their second group of data sets for the comparison, which are referred to as old. The old set can be thought of as the reference set, or the unchanged environment.

Note: For the Group Compare process, ensure that the number of new data sets being compared matches the number of old data sets. From these two sets of libraries, z/OS Change Tracker creates pairs for comparison by sequentially picking them one line at a time from both lists.

The following table shows two lists of data sets that are paired for a group compare.

Data Set	Pairing	Comparison Data Set
DEV.PRODUCT.INSTALL	Paired with	TEST.PRODUCT.INSTALL
DEV.PRODUCT.LINKLIB	Paired with	TEST.PRODUCT.LINKLIB
DEV.PRODUCT.MAC	Paired with	TEST.PRODUCT.MAC
DEV.PRODUCT.PANELS	Paired with	TEST.PRODUCT.PANELS
DEV.PRODUCT.PARMLIB	Paired with	TEST.PRODUCT.PARMLIB
DEV.PRODUCT.SOURCE	Paired with	TEST.PRODUCT.SOURCE

Note: The Group Compare function in ISPF uses the COMPARE command for each paired data set and requires equal number of data sets in each group so they can be paired. However, the GROUPCOMPARE command in batch does not have this requirement. Refer to the [Chapter 4, “Using z/OS Change Tracker batch commands,”](#) on page 73 to see examples of using [“GROUPCOMPARE command”](#) on page 92 and [“REMOTEGROUPCOMP command”](#) on page 112.

In this section, we use the Group Compare function. Two sets of panels are needed to define the New and the Old groups of data sets.

Enter a data set pattern to define the New data set group

On the following panel specify a data set pattern and press Enter. Here, we have specified the DEV.PRODUCT.* as the New data set group.

```

Group Compare: New Data Sets
Command ==>
                                SCROLL==> CSR

To perform a Group Compare of PDS pairs, establish the "New" set by specifying
one or multiple DSNs or Dsname levels on each line, and then press ENTER.

    Explicit DSN or Dsname Level
-   DEV.PRODUCT_____

```

Figure 67. Group compare - defining the new set (1 of 2)

Resolved data sets from the catalog, for defining the New data set group

```
Command ==> Group Compare - Options SCROLL==> CSR
Job Name . . . . COMPARE
Compare Option
1 1. DIFF
  2. SAME
  3. MATCH
  4. MISMATCH
  5. MODS
  6. ALL
Enter "/" to select option
Include ISPF Statistics
Batch execution
X Data Set Name Prompt
- DEV.PRODUCT.INSTALL
- DEV.PRODUCT.LINKLIB
- DEV.PRODUCT.MAC
- DEV.PRODUCT.PANELS
- DEV.PRODUCT.PARMLIB
- DEV.PRODUCT.SOURCE
***** Bottom of data *****
```

Figure 68. Group compare - resolved libraries for the new set (2 of 2)

On Figure 68 on page 59, the result of the user entering DEV.PRODUCT is shown. The processing options for the input field called Process Option have already been discussed under the [Macro Compare utility](#). The input field called Consider Directory directs z/OS Change Tracker to consider ISPF statistics in the comparison of members. If two matched members have identical content but different ISPF statistics, they will be reported as DIFF, if this option was requested. Notice that the user may exclude unwanted data sets from the panel. Above, we have not excluded any data sets from the resolved list.

Upon pressing Enter on the above panel, the user is presented with another panel, allowing the user to define their “Old” set. In this case, the user is interested in the pattern TEST.PRODUCT for defining their Old set.

Panel for identifying the Old data sets

```
Command ==> Group Compare: Old Data Sets SCROLL==> CSR
Establish the "Old" set by specifying one or more specific DSNs or dsname
levels. Then press Enter to refine the resolved list.
Explicit DSN or Dsname Level
_ test.product_____
```

Figure 69. Group compare - defining the old set (1 of 2)

Resolved data sets from the catalog, for defining the Old data sets group

```
Command ==> Group Compare: Old Resolved List SCROLL==> CSR
Data Set Name Prompt
- TEST.PRODUCT.INSTALL
- TEST.PRODUCT.LINKLIB
- TEST.PRODUCT.MAC
- TEST.PRODUCT.PANELS
- TEST.PRODUCT.PARMLIB
- TEST.PRODUCT.SOURCE
```

Figure 70. Group compare - resolved libraries for the old set (2 of 2)

The user can tailor the list as shown above, and press Enter to generate the batch job. A return code of RC=4 indicates that there were some differences identified. However, a return code of RC=0 indicates that no DIFF or MISMATCH members were found comparing the two data sets group. This is the confirmation that both environments have the same configuration and all their members have the same contents.

You can see in the generated JCL how the data sets in both groups have been paired for the COMPARE commands.

```

Command ==> -----
000012 //CYGIN      DD *
000013 OPTIONS=DIFF
000014 *
000015 * FOR UNCATALOGED DSNS, ADD THE VOLN=VVVVVV COMMAND FOR THAT DATA SET.
000016 *
000017 DSN1=DEV.PRODUCT.INSTALL
000018 DSN2=TEST.PRODUCT.INSTALL
000019 COMPARE
000020 *
000021 DSN1=DEV.PRODUCT.LINKLIB
000022 DSN2=TEST.PRODUCT.LINKLIB
000023 COMPARE
000024 *
000025 DSN1=DEV.PRODUCT.MAC
000026 DSN2=TEST.PRODUCT.MAC
000027 COMPARE
000028 *
000029 DSN1=DEV.PRODUCT.PANELS
000030 DSN2=TEST.PRODUCT.PANELS
000031 COMPARE
000032 *
000033 DSN1=DEV.PRODUCT.PARMLIB
000034 DSN2=TEST.PRODUCT.PARMLIB
000035 COMPARE
000036 *
000037 DSN1=DEV.PRODUCT.SOURCE
000038 DSN2=TEST.PRODUCT.SOURCE
000039 COMPARE
000040 *

```

Figure 71. Group compare JCL

Verify the generated job to ensure the data sets are properly paired and then submit the job for execution. Segments of the sample reports are shown in [Figure 87 on page 68](#) and [Figure 88 on page 68](#).

Below, we have shown the install libraries in both sets that have some differences. Three members are identified as being different in their contents (DIFF). No mismatch has been identified, 67 members exist in both libraries.

```

NEW: DEV.PRODUCT.INSTALL                                OLD TEST.PRODUCT.INS
Member          DSN1-Update-Date  DSN2-Update-Date  OPTIONS= DIFF
DIFF            SYNCARC           2013/09/09 01:51  2013/07/01 19:57
DIFF            CYGCHECK          2013/09/05 09:26  2009/03/10 18:22
DIFF            CYGCOPY           2013/09/05 09:38  2013/09/05 09:38

CYGMAIN  (857W) COMPARE STATISTICS FOR:
              - DSN1: DEV.SECURE.INSTALL
              - DSN2: TEST.PRODUCT.INSTALL

              67 TOTAL MEMBER NAMES PAIRED
              64 MEMBER(S) HAD NO CHANGES
**           3 MEMBER(S) HAD CHANGES
              0 NEW FILE MEMBER(S) NOT PAIRED
              0 OLD FILE MEMBER(S) NOT PAIRED

```

Figure 72. Example group compare report (1 of 2)

```

NEW: DEV.PRODUCT.LINKLIB                                OLD TEST.PRODUCT.LINKL

CYGMAIN  (856I) COMPARE STATISTICS FOR:
          - DSN1: DEV.SECURE.LINKLIB
          - DSN2: TEST.PRODUCT.LINKLIB

          58 TOTAL MEMBER NAMES PAIRED
          58 MEMBER(S) HAD NO CHANGES
           0 MEMBER(S) HAD CHANGES
           0 NEW FILE MEMBER(S) NOT PAIRED
           0 OLD FILE MEMBER(S) NOT PAIRED

```

Figure 73. Example group compare report (2 of 2)

Above, two other paired data sets are shown which have no differences. The LINKLIB in each environment has 58 load modules in which all of them have identical contents.

Support for uncataloged data sets

If the data sets involved in the compare process are uncataloged, the commands VOL1 and VOL2 must be used respectively for each data set in a given data set pair, shown below:

```

VOL1=RES001
DSN1=DEV.PRODUCT.INSTALL
VOL2=RES002
DSN2=TEST.PRODUCT.INSTALL
COMPARE          <== Now z/OS Change Tracker compares the pair of data sets
*
```

Reconcile two libraries

During a z/OS maintenance cycle, it may be required to reflect the newly introduced changes in a base library into a target library. In general, there are occasions that two libraries need to be reconciled.

z/OS Change Tracker offers the Reconcile Two Libraries utility to meet this need. This utility provides the mechanism to ensure that a pair of partitioned data sets contain identical member configurations. We now examine how the z/OS Change Tracker Reconcile works.

To begin, designate a From library. This library is not altered in any way during the Reconciliation process. z/OS Change Tracker updates the members of the To ("Old") library to make them match the corresponding members in the From library.

Reconciliation Process

z/OS Change Tracker achieves reconciliation by identifying members in the From library which are different (DIFF) from the To library, and then copies them over to achieve synchronized libraries.

Any new members residing in the From library which are absent in the To library are also copied over. You can use the DELETE MISMATCH option to remove members of the To library which are absent in the From library.

Intelligent Reconciliation

Before the copy operation begins, z/OS Change Tracker deletes the members to be replaced, deletes the extra members, and then compresses the To PDS, to avoid out of space conditions.

To simulate how the To library will be affected, specify YES for the TESTRUN option. This option causes z/OS Change Tracker to produce only a report identifying the changes (ADD, REP, DEL) that would take place, but it suppresses the actions of the actual reconciliation.

Specifying the From data set

The following is the initial panel displayed to the user after selecting Reconcile two Libraries. Among other options, this panel allows the user to identify their From library (the changed library DEV.PRODUCT.PANELS). Most of the input fields on this initial panel are self-explanatory.

```
Reconcile Two PDS (1 OF 2)

Command ==>

ISPF Library:
Project ==> DEV
Group ==> PRODUCT
Type ==> PANELS

"From" Other Partitioned Data Set:
Data set name ==>
Volume Serial ==> (If not cataloged)

Job Name ==>

Enter "/" to select option
- Compress the PDS
- Delete Mismatch
- Test Run
```

Figure 74. Reconcile two libraries (specifying the from data set)

The following table summarizes the three parameters.

Table 9. Description of input parameters for reconciliation (CYGUCOP1)	
Field	Description
Compress the PDS	Compresses the To PDS prior to the copy operations of reconciliation, which helps avoid out of space conditions.
Delete Mismatch	Removes members from the To library that are not present in the From library.
Test Run	This option reports the REP, ADD, and DEL members along with statistics, without performing any reconciliation.

Specifying the To ("Old") data set

Pressing Enter on the initial panel shown in [Figure 74 on page 62](#) displays the second panel, in which you can specify the To data set.

```
Reconcile Two PDS (2 OF 2)

Command ==>

ISPF Library:
Project ==> DEV
Group ==> OSECURE
Type ==> PANELS

"Old" Other Partitioned Data Set or Sequential Data Set:
Dsname ==>
Volume Serial ==> (If not cataloged)
```

Figure 75. Reconcile two libraries (specifying the To ("Old") data set)

As shown above, the user has designated their To data set to be DEV.OSECURE.PANELS. This To data set will be Reconciled (altered) to match the configuration and content of the From data set, defined earlier. Upon pressing Enter, z/OS Change Tracker generates the batch job for performing the Reconciliation.

Because the TESTRUN=Y is specified, the following job just identifies the ADD, REP, and DEL members without actually performing the reconcile operation.

```
TESTRUN=YES
COMPRESS=YES
DELETEMEM=YES
*
DSN1=DEV.PRODUCT.PANELS
DSN2=DEV.OSECURE.PANELS
COPYDIFF
*
```

Figure 76. Sample reconcile JCL

Reconcile groups of libraries

This function enables users to reconcile the differences between groups of partitioned data sets (such as PDS, PDSE, source, load). This utility provides the mechanism to ensure that multiple pairs of partitioned data sets contain identical member configurations with identical member contents.

Because libraries in both groups must be paired for the COPYDIFF operation, each group must have the same number of data sets. z/OS Change Tracker creates pairs of data sets by sequentially picking them one line at a time from both lists and matching them.

We now examine how z/OS Change Tracker reconciles groups of data sets. First, the user designates a From set of libraries. These “From” libraries are not altered in any way during the reconciliation process. They are only used as a reference to compare against the To set of libraries. z/OS Change Tracker updates the members of the To libraries to make them identical with the members in the From libraries.

In the Reconcile Groups of Libraries process, it is important to make sure that the number of To data sets being reconciled matches the number of From data sets.

Table 10. Data sets paired for group reconciliation		
DEV.PRODUCT.INSTALL	Paired with	TEST.PRODUCT.INSTALL
DEV.PRODUCT.LINKLIB	Paired with	TEST.PRODUCT.LINKLIB
DEV.PRODUCT.MAC	Paired with	TEST.PRODUCT.MAC
DEV.PRODUCT.PANELS	Paired with	TEST.PRODUCT.PANELS
DEV.PRODUCT.PARMLIB	Paired with	TEST.PRODUCT.PARMLIB
DEV.PRODUCT.SOURCE	Paired with	TEST.PRODUCT.SOURCE

For details on the processing options for reconciliation, refer to [“Intelligent Reconciliation” on page 61](#).

Specifying the From data sets

The following is the initial panel displayed to the user after selecting Reconcile Groups of Libraries. This panel is primarily designed to allow a user to identify their From set of libraries (on the lines shown). Multiple explicit library names or a library pattern may be specified. In the example below, the user has decided to specify the library pattern DEV.PRODUCT

Reconcile Groups: From		SCROLL==> CSR
Command ==>		
To perform a Group Reconcile of PDS pairs, establish the "From" set by specifying one or more specific DSNs or Dsname levels, then press ENTER.		
Explicit DSN or Dsname Level		
_ dev.product_____		

Figure 77. Reconcile groups of libraries - specifying the From data sets

Upon pressing Enter on the above panel, the user is presented with a second panel (shown below) which displays the resolved libraries from the system catalog, and allows the user to specify other processing options for reconciliation.

Resolved list of From data sets, and the processing options

```

                                Group Reconcile - Options
Command =====>                                SCROLL====> CSR

Job Name      ==>

Compress the PDS ==> Y          Y/N Y: Compress the "To" PDS
Delete Mismatch ==> Y          Y/N Y: Delete the MISMATCH from the "To" PDS
Testrun       ==> N          Y/N Y: Only produce the Reconcile Report

Data Set Name
- DEV.PRODUCT.INSTALL
- DEV.PRODUCT.LINKLIB
- DEV.PRODUCT.MAC
- DEV.PRODUCT.PANELS
- DEV.PRODUCT.PARMLIB
- DEV.PRODUCT.SOURCE

```

Figure 78. Reconcile groups of libraries - the resolved From data sets

The panel displays the resolved libraries from the catalog, which can be used as the From set of libraries for the reconciliation. Data sets can be excluded from the list by using the X line command. The input fields Compress the PDS, Delete Mismatch, and Testrun, have already been discussed in the previous section [“Reconcile two libraries”](#) on page 61.

Once the user presses Enter, they are able to designate their To set of libraries. Here the user can specify multiple explicit or pattern libraries to be used as the To set of libraries for the reconciliation.

Specifying the To data sets

The following panel is displayed to the user for designating the To set of libraries during reconciliation. Multiple explicit library names or a library pattern may be specified. In the example below, the user has decided to specify the library pattern TEST.PRODUCT.

```

                                Group Reconcile: To
Command =====>

Establish the "To" set by specifying one or more DSNs or Dsname levels, and
then press ENTER to refine the resolved list.

Explicit DSNs or Dsname Levels
- test.product_____

```

Figure 79. Reconcile groups of libraries - specifying the To data sets

Resolved list of To data sets

Press Enter on the above panel to display the resolved libraries from the catalog. Data sets can be excluded from the list by using the X line command as shown. Once the user presses Enter, z/OS Change Tracker generates a batch job for performing the group reconciliation.

Group Reconcile - Resolved "To" List	
Command ==>	
X	Data Set Name
-	DEV.PRODUCT.INSTALL
-	DEV.PRODUCT.LINKLIB
-	DEV.PRODUCT.MAC
-	DEV.PRODUCT.PANELS
-	DEV.PRODUCT.PARMLIB
-	DEV.PRODUCT.SOURCE
	Prompt

Figure 80. Reconcile groups of libraries - resolved To list

As shown above, the user has designated their To data set to be TEST.PRODUCT group of libraries. This To data set will be reconciled (altered) to match the configuration and content of the From data set, defined earlier. Upon pressing Enter, z/OS Change Tracker generates the batch job for performing the reconciliation.

```
//CYGIN DD *
TESTRUN=YES
COMPRESS=YES
DELETEMEM=YES
*
* FOR UNCATALOGED DSNs, ADD THE VOLN=VVVVVV COMMAND FOR THAT DATA SET.
*
DSN1=DEV.PRODUCT.INSTALL
DSN2=TEST.PRODUCT.INSTALL
COPYDIFF
*
DSN1=DEV.PRODUCT.LINKLIB
DSN2=TEST.PRODUCT.LINKLIB
COPYDIFF
*
DSN1=DEV.PRODUCT.MAC
DSN2=TEST.ZSECURE.MAC
COPYDIFF
*
```

Figure 81. Group reconciliation JCL

In the figure above, a segment of the created job is shown. Because the TESTRUN=Y is specified, the job just identifies the ADD, REP, and DEL members without actually performing the reconcile operation.

Note: Make sure the data sets are paired correctly before submitting the job.

Support for uncataloged data sets

If the data sets involved in the reconciliation process are uncataloged, the commands VOL1 and VOL2 must be used respectively for each data set in a given data set pair (as shown below):

```
VOL1=RES001
DSN1=DEV.PRODUCT.INSTALL
VOL2=RES001
DSN2=TEST.PRODUCT.INSTALL
COPYDIFF <== Reconcile the first pair of data sets
*
VOL1=RES001
DSN1=DEV.PRODUCT.LINKLIB
VOL2=RES001
DSN2=TEST.PRODUCT.LINKLIB
COPYDIFF <== Reconcile the second pair of data sets
*
```

Tokenize directory files

z/OS Change Tracker has a direct interface to the Open Edition MVS (OMVS). Thus, users can view the regular files and directories in an zFS or z/OS UNIX path. In order to generate a token representation of the files in a z/OS UNIX directory, the z/OS Change Tracker Tokenize Files in a Directory option must be

selected. This utility first expands a specified zFS or USS path by listing all regular files and directories in the path. Next, the user must select a particular directory of interest to be tokenized. The point to be emphasized is that using the ISPF interface, z/OS Change Tracker tokenizes the entire directory and not the subdirectories within that directory.

Tokenizing subdirectories within a directory requires a two step process, first resolving all subdirectories and then passing the subdirectories to the z/OS Change Tracker tokenize function. See CYGU3TAF in Chapter 5, “Sample jobs,” on page 147 for more information.

z/OS Change Tracker creates a hash token value for each regular file in the selected OMVS directory. The token file establishes a content reference for each regular file in the selected directory so that any changes made to them over time can be detected and reported. The following is the initial panel for specification of an OMVS path to expand.

Tokenize an OpenMVS Directory		
Command ==>		
Specify a Pathname and a Token file below and then press ENTER to list the Files in the path. Path is case sensitive: Examples: /u OR /u/jones		
OpenMVS Path to expand	==> /u/smith	
New or Existing Token file	==> TOKEN	
(New only) DASD Unit	==> SYSDA	(SYSDA,SYSCALLDA...)
Space Units	==> TRK	(TRK,CYL,BLK)
Primary	==> 1	(1-5)
Secondary	==> 1	(0-5)
Saved JOB	==> ZFSTOKEN (JOB is saved under this name)	

Figure 82. Tokenize files in a path (1 of 2)

In the panel above, the user is tokenizing the directory /u/smith. By pressing the Enter key, all files in the directory are displayed for viewing. Press the Enter key again and a tokenization job is created. The paths and file names are case-sensitive.

Specification of a Token File and a z/OS UNIX path

The token file can be an existing file on your system, or a new file to be allocated by the generated batch job. Note the file allocation specifications for the new token file called TOKEN.U.SMITH in the above example. Alternatively, this user could have specified the z/OS UNIX path /u/smith/*.* to first view and then tokenize all regular files in the specified directory.

List of Directories and Regular Files in the path

Upon pressing the Enter key on the initial panel, a second panel displays the regular files and subdirectories in the path. Notice that all the regular files and directories in the path /u/smith are listed. Regular files are indicated with the keyword File, and directories are indicated with the keyword Dir. At this point, the user must either press Enter to proceed with tokenizing all the regular files in the /u/smith directory, or exit and re-enter a different directory for tokenization.

List of Directories and Files		SCROLL==>
Command ==>		
Files in the selected path are presented. Press ENTER to tokenize all files.		
OpenMVS Path	==> /u/smith	
Type Resolved Files in the Selected Path		
File file1.src		
File file2.src		
File file3.src		
File cnt1.jcl		
File copy.jcl		

Figure 83. Tokenize files in a path (2 of 2)

By pressing Enter, the tokenize JCL is created. Below, a fragment of the job is shown.

```
//CYGIN DD *  
SYSUT1=PATH=' /u/smith/*. *'  
SAVEDD=DD1  
*
```

Figure 84. JCL to tokenize files in a z/OS UNIX path

Report changed files in a directory

Use the Report Changed Files in a Directory option to detect changes made to regular files in a z/OS UNIX directory, over time or across environments.

This utility requests the name of a pre-generated token file which contains the reference tokens. To identify the changed files, the regular files in the specified z/OS UNIX path are tokenized and kept in memory. These tokens are compared with the ones stored in the token file and variances in the hash token values indicate that there has been change in the content of the file.

To generate a difference report, z/OS Change Tracker first matches the paths. When two files are successfully matched, their token values are examined to determine if the contents are the same.

```
Report the changed files in a path  
Command ==>  
  
To identify the changed files, specify a pre-generated token file followed by  
a path name or pattern. Example: /u/jones/*. * or /u/jones/*.src  
  
Pre-generated Token File  TOKEN.U.SMITH  
Path name . . . . . /u/smith  
  
Job name . . . . . ZFSCHG  
  
Compare Option  
1 1. CHANGESONLY  
2 2. SAME  
3 3. DIFF  
4 4. MATCH  
5 5. MISMATCH
```

Figure 85. Report changed files in a directory

Compare Option

The Compare Option field affects the contents of the difference report.

Two files are considered to be the SAME if their names match, and their file content as represented by their tokens are identical. If two files have the same name but their tokens are different, they are reported as DIFF. If a user requests a reporting option of MATCH, z/OS Change Tracker will identify all files with matching names, regardless of whether their content is the SAME or DIFF. A MISMATCH means that the file name exists in one path but not the other. For example, if the pre-generated token file contains a token for a file called test.jcl, but this file has been deleted or renamed in the specified path, this will result in a MISMATCH. Similarly, a file which appears in the path but not in the token file will also result in a MISMATCH. The CHANGESONLY option reports those files that are DIFF or MISMATCH.

In Figure 85 on page 67, the user has specified a pre-generated token file called TOKEN.U.SMITH. This token file will be compared with the z/OS UNIX path /u/smith/*. * This user wishes to compare their pre-generated token file with the specified Explicit pathname, to see if any of the regular files have changed in content over time. Notice that the Compare Option field has been specified as CHANGESONLY, to report both differences in file contents and mismatched file names. The following sample JCL is created for this function. For more explanation of the commands used below, see [Chapter 4, “Using z/OS Change Tracker batch commands,” on page 73](#).

```
//CYGIN      DD *
OPTION=CHANGESONLY
SYSUT1=PATH='/u/smith/*.*'
LOADDD=DD1
PATHCOMPARE
*
```

Figure 86. JCL to report changed files over time

Generated Difference Report

In the example shown on Figure 85 on page 67, the following report displays matching file names with DIFFerent content. The files in the /u/smith path were grouped in two categories based on their suffixes: .src and .jcl. Among the 3 files with the suffix of .src there was only one file with a DIFFerent token. This file was file2.src. The other two files in this category were remained unchanged. The statistics indicate that there were 3 files in the .arc category where two of them had remained unchanged.

```
-----
Change Detection Begins for the Following two PATHS
-----
/u/smith/*.src
/u/smith/*.src

      SYSUT1:NEW-stamp  SYSUT2:OLD-stamp  FILENAME(prefix only)

UPD   2013/09/17 09:45  2013/09/17 09:29  file2

CYGMAIN  (890W)          SYSUT1          SYSUT2
                        3                3 Matched File Names
                        2                2 SAME Content
                        1                1 DIFFerent Content (UPD)
                        0                0 Mismatched Files (ADD.DEL)
```

Figure 87. Report of changed files (1 of 2)

In the Group Compare Summary Report below, it is indicated that there are two matched paths/categories .src and .jcl. Only one of the two paths/categories has some differences, the .src path. The second path/category .jcl does not have any changed files.

Note: If the new environment of /u/smith happened to have a new category of files like .cpy then this would constitute an unmatched path/category in LIST2 (now it shows 0 below).

```
Group Compare Summary Report
=====

CYGMAIN (296I) SUMMARY STATISTICS:
      2 TOTAL MATCHED PATHS
      0 TOTAL MISMATCH PATHS ON LIST1
      0 TOTAL MISMATCH PATHS ON LIST2
      1 TOTAL PATHS WITH DIFFERENCES

*** List of PATHS With Differences ***

      /u/smith/*.src

CYGMAIN (298I) NUMBER OF CHANGES WRITTEN TO TARCMDS FILE:  0

CYGMAIN (803I) WARNINGS: 1
                  ERRORS  : 0
```

Figure 88. Report of changed files (2 of 2)

Other reporting options

If a Compare Option of SAME or MISMATCH is specified the report would look very similar, except that the left hand column would display the word SAME or MISMATCH respectively. When MATCH is selected, all files that satisfy the criteria for SAME or DIFF will be identified in the same output report, and therefore the Difference report will contain more entries.

Table 11. Description of input fields (CYGOCHG)	
Field	Description
Pre-generated Token File	This field is used to specify a pre-generated token file containing the tokens of the regular files in one or more z/OS UNIX paths. Refer to the CYGU* jobs in the samples library for more examples.
Path name	The z/OS UNIX path which is being compared to the pre-generated token file. If the content of a regular file has changed since it was tokenized, it will be reported.
Compare Option	<p>This field is used to specify a reporting option. Valid reporting options include:</p> <ul style="list-style-type: none"> • CHANGESONLY: Reports those files that are DIFF or MISMATCH. • SAME: Two files are reported as SAME if their names match and their file content is the same. • DIFF: If two files have matching names but their content is different, they are reported as DIFF. • MATCH: Identifies all files with matching names, regardless of whether their content is SAME or DIFF. • MISMATCH: Reports cases where the file appears in one location but not the other.

Macro compare files

Use the Macro Compare Files option to quickly report similarities and differences between any two z/OS UNIX paths.

The reporting options are SAME, MATCH, MISMATCH, AND DIFF. See [“Report changed files in a directory”](#) on page 67 for more information.

All regular files in the specified paths are tokenized internally in order to perform a precise file by file comparison. Because z/OS Change Tracker is able to use the tokens to intelligently compare file contents, similarities and differences between the two paths can be quickly identified in a concise report. The following panel depicts the input fields that must be specified for the comparison report.

Compare two Paths

Command ==>

To compare regular files in two existing paths, specify the names of the paths and press ENTER. Paths are case sensitive: Example: /u/jones/*.src

"New" Pathname ==> /u/smith/*.jcl
"Old" Pathname ==> /u/jones/*.jcl

Process option: ==> DIFF SAME, DIFF, MATCH or MISMATCH
Saved JOB ==> ZFSCOMP (for Batch, JOB is saved under this name)

Figure 89. Macro compare files in two Paths

The input fields on this panel are the same as the ones described in [“Report changed files in a directory”](#) on page 67. The generated report for the Macro Compare and Report Directory Changes utilities is also the same, except that two z/OS UNIX paths are being compared rather than a z/OS UNIX path and a token file.

```

OPTION=DIFF
*
USSPATHREPL=(*u/smith', '/u/jones')
SYSUT1=PATH='/u/smith/*.jcl'
SYSUT2=PATH='/u/jones/*.jcl'
PATHCOMPARE
*

```

Figure 90. JCL to compare two paths

Below, two files have been reported as being different, cntl.jcl and copy.jcl. The suffix .jcl is not shown in the actual report, but shown in the beginning of the report indicating the category of files being compared.

Change Detection Begins for the Following two PATHS				

/u/smith/*.jcl				
/u/jones/*.jcl				
	SYSUT1:NEW-stamp	SYSUT2:OLD-stamp	FILENAME(prefix only)	
UPD	2013/09/17 09:30	2013/09/17 10:03	cntl	
UPD	2013/09/17 09:30	2013/09/17 10:03	copy	
CYGM	MAIN (890W)	SYSUT1	SYSUT2	
		2	2	Matches File Names
		0	0	SAME Content
		2	2	DIFFerent Content (UPD)
		0	0	Mismatched Files (ADD,DEL)

Figure 91. Path compare report

List a token file

If a user is to perform the utility Report Change Files in a Directory, a token file needs to be specified along with an z/OS UNIX path of interest. The user needs to list the content of the token file ensuring the token file is up to date, including all the files to compare. This will determine which regular files will be tokenized and stored in the token file.

The following panel depicts how the content of a token file can be listed in a report.

List contents of a Token File

Command ==>>>SCROLL==> CSR

Specify the Token file below and then press ENTER.

Token File: ==> TOKEN.U.SMITH

Saved JOB ==> ZFSLIST (for Batch, JOB is saved under this name)

Figure 92. List a token file

In the above example, the user has specified TOKEN.U.SMITH as the name of their Token File. This results in a JCL shown below.

```
//CYGIN DD *
LOADDD=DD1
LIST
```

Figure 93. JCL to list content of a Token File

Contents of the Token File

The sample report below illustrates a typical token file listing. Files are grouped according to their suffix. In this example, the TOKEN.U.SMITH file contained only two file types as indicated by the suffix types, .jcl and .src. All these files along with their token values have been stored in the specified token file.


```

CYGMAIN  (881I) Tokens for the path is set in memory: SYSUT1
           PATH: /u/smith/*.jcl

CYGMAIN  (881I) Tokens for the path is set in memory: SYSUT1
           PATH: /u/smith/*.src

List of the Tokenized Files:

Dir: /u/smith
file cntl.jcl
file copy.jcl

Dir: /u/smith
file file1.src
file file2.src
file file3.src

```

Figure 94. Example contents of a token file

Table 12. Description of input fields (CYGOLIST)	
Field	Description
Token File	Specify the name of a pre-existing token file generated some time in the past.

Chapter 4. Using z/OS Change Tracker batch commands

This chapter provides a description of all the main batch commands used in z/OS Change Tracker.

z/OS Change Tracker commands are executed by the CYGMAIN batch program and are passed in the CYGIN DD statement:

```
//jobname JOB
//stepname EXEC PGM=CYGMAIN,REGION=0M
//STEPLIB DD DISP=SHR,DSN=product-high-level-qualifier.SCYGLNK
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CYGIN DD *
Command1
Command2
.
.
.
//*
```

JCL statements must be entered in uppercase. z/OS Change Tracker commands can be abbreviated as long as the command is unambiguous. For example, MEMBERRECOVER, MEMBERRECOV, and MEMBERREC are all accepted forms of the command.

Some commands are dependent on the output of other commands. Commands such as TOKEN=LIST would operate on a LIST that is produced by a previous command such as SHOW=PROTECTIONS. Some commands can alter their behavior or output depending on previous OPTION commands. For example, the types of activities displayed by the SHOW=ACTIONS command can be controlled by parameters specified on a previous OPTION command.

The supplied SCYGSAMP data set provides additional and more complete examples of using the commands, and interactions between commands.

How to read the syntax diagrams

To read syntax diagrams, follow one line at a time from the beginning to the end, and code everything you encounter on that line.

Text-based syntax diagrams

The list below details how to read text-based syntax diagrams:

- UPPERCASE LETTERS or WORDS must be coded as they appear in the syntax diagrams.
- Lowercase letters or words represent variables for which you must supply a value.
- Parentheses () must be entered exactly as they appear in the syntax diagram.
- An ellipsis . . . (three consecutive periods) indicates that you can enter the preceding item more than once.
- A single item in brackets [] indicates that the enclosed item is optional. Do not specify the brackets in your command.
- Stacked items in brackets [] indicate that the enclosed items are optional. You can choose one or none. Do not specify the brackets in your command.
- Stacked items in braces { } indicate that the enclosed items are alternatives. You must specify one of the items. Do not specify the braces in your command.
- Items separated by a vertical bar | indicate that you can specify only one of the items. Do not specify the vertical bar in your command.
- An underlined operand indicates the default value when no alternate value is specified.
- Single quotation marks ' ' indicate that information must be enclosed in single quotation marks.

Railroad syntax diagrams

The following list applies to railroad syntax diagrams:

- Read the syntax diagrams from left to right and top to bottom.
- Each syntax diagram begins with a double arrowhead (➤➤) and ends with opposing arrowheads (➤➤).
- An arrowhead (➤) at the end of a line indicates that the syntax continues on the next line. A continuation line begins with an arrowhead (➤).
- You must provide all items enclosed in parentheses, (), and you must include the parentheses.
- Required items appear on the horizontal line (the main path).

➤➤ required_item — required_item ➤➤

- Optional items appear below the main path.

➤➤ required_item — optional_item ➤➤

- If you can choose from two or more items, they appear vertically, in a stack. If you *must* choose one of the items, one item of the stack appears on the main path.

➤➤ required_item — required_choice1
required_choice2 ➤➤

If choosing one of the items is optional, the entire stack appears below the main path.

➤➤ required_item — optional_choice1
optional_choice2 ➤➤

If one of the items is the default, it will appear above the main path and the remaining choices will be shown below.

➤➤ required_item — default_choice
optional_choice
optional_choice ➤➤

- An arrow returning to the left, above the main line, indicates an item that can be repeated.

➤➤ required_item — repeatable_item ➤➤

If the repeat arrow contains a comma, you must separate repeated items with a comma.

➤➤ required_item — repeatable_item ➤➤

A repeat arrow above a stack indicates that you can repeat the items in a stack.

ARCHIVEBACKUP command

The ARCHIVEBACKUP command moves backups from live backup to archive backup.

Syntax

```
ARCHIVEBACKUP=(DSN=data-set-name
                ,MEM=member-name
                ,VER=version-number
                [,VOL=volume-serial])
```

➤ ARCHIVEBACKUP= — (— DSN=*data-set-name* — ,MEM= *member-name* — ,VER= *version-number* — ➤

└──────────────────────────┘) ➤
 ,VOL= *volume-serial*

Parameters

DSN=*data-set-name*

The name of the data set containing the member to be archived.

MEM=*member-name*

The name of the member to be archived.

VER=*version-number*

The version number of the member to be archived. Member versions in the live backup are identified by number, where 0 is the current version, -1 is the previous version and so on. A wildcard character cannot be specified.

VOL=*volume-serial*

For uncataloged data sets, the volume serial number of the volume on which the data set resides. Omit this parameter if the data set is cataloged.

Usage

This command is available for use by the Administrator only.

Archiving is necessary to keep the live backup system smaller in size and provides a long term backup medium that can be kept for future recovery. Any version of a member backup can be selected for archiving.

Regularly running the AUTOARC command will improve archive organization. AUTOARC performs archiving based on DAYS or COPIES criteria set when the data sets are protected or reprotected. See the AUTOARC command for further details.

Example 1: Archive the three latest backup versions for a member in a protected data set

```
ARCHIVEBACKUP=(DSN=SYS1.PARMLIB,
                MEM=IEASSN00,
                VER=-2)
ARCHIVEBACKUP=(DSN=SYS1.PARMLIB,
                MEM=IEASSN00,
                VER=-1)
ARCHIVEBACKUP=(DSN=SYS1.PARMLIB,
                MEM=IEASSN00,
                VER=0)
```

Related commands

[ARCHIVERECOVER](#)

[AUTOARC](#)

ARCHIVERECOVER command

The ARCHIVERECOVER command is used to recover archived backups.

Syntax

```
ARCHIVERECOVER=(DSN=data-set-name
, MEM=member-name
, VER=version-number
, OUTDSN=recovery-data-set
[, VOL=volume-serial])
```

```

➤ ARCHIVERECOVER= ( — DSN= data-set-name — ,MEM= member-name — ,VER= version-number —
➤ ,OUTDSN= recovery-data-set — ,VOL= volume-serial — ) ➤

```

Parameters

DSN=*data-set-name*

The name of the data set containing the member to be recovered.

MEM=*member-name*

The name of the member to be recovered. The member must have been previously archived.

VER=version-number

The version number of the member to be recovered. Member versions in the live backup are identified by number, where 0 is the current version, -1 is the previous version and so on. A wildcard character cannot be specified.

OUTDSN=*recovery-data-set*

A destination PDS or PDSE to contain the recovered members. It should have the same attributes as the data set that the members were originally in.

VOL=volume-serial

For uncataloged data sets, the volume serial number of the volume on which the original data set resides. Omit this parameter if the data set is cataloged.

Usage

This command is available for use by the Administrator only.

Backups of the specified member are assumed to exist and have been archived using the ARCHIVEBACKUP or AUTOARC command. SHOW=ARCHIVES is useful to show archive versions.

Ensure that the OUTDSN exists and is empty before recovery is started.

Example 1: Recovering multiple versions of a specific member from the archive system

```

ARCHIVERECOVER=(DSN=SYS1.PARMLIB,
                  MEM=IEASSN00,
                  VER=0,
                  OUTDSN=IBMU19.RECOVERY.PDS)
*
ARCHIVERECOVER=(DSN=SYS1.PARMLIB,
                  MEM=IEASSN00,
                  VER=-1,
                  OUTDSN=IBMU19.RECOVERY.PDS)
*
ARCHIVERECOVER=(DSN=SYS1.PARMLIB,
                  MEM=IEASSN00,
                  VER=-2,
                  OUTDSN=IBMU19.RECOVERY.PDS)

```

UNPROTECT

AUTOARC command

The AUTOARC command examines all protected data sets and applies the archival policies to each of the protected data sets.

Syntax

```
AUTOARC
```

➤ AUTOARC ➤

Parameters

None.

Usage

This command is available for use by the Administrator only.

The archival policies for a given data set are set when the data set is protected or reprotected. The DAYS and COPIES parameters on the PROTECT and REPROTECT commands can be used to set either the number of days for which a backup is kept or the maximum number of backup copies that are allowed for each member. Member backups that fall outside either of these windows when AUTOARC is run will be archived and deleted from the live backup system. The live and archive backup systems are part of the repository files that are set up when the product is installed and configured.

The started task must be stopped or deactivated before AUTOARC can be run.

When no data set list is provided, the AUTOARC command runs against all protected data sets. If a list is present, AUTOARC operates against the list of the data set established before the AUTOARC command is run.

Example 1: Automatically archiving all versions meeting the archival criteria

```
DEACTIVATESTC
AUTOARC
REACTIVATESTC
```

Example 2: AUTOARC function to run against a specified data set

```
EXPLICIT=IBMUSER.HUGE.PDS
SHOW=LIST
DEACTIVATESTC
AUTOARC
REACTIVATESTC
```

Example 3: AUTOARC function to run against all except two protected data sets

```
SHOW=PROTECTIONS
EXCLUDE=IBMUSER.TEST.PDSA
EXCLUDE=IBMUSER.HUGE.PDS
...
SHOW=LIST
*
DEACTIVATESTC
AUTOARC
REACTIVATESTC
```


Related commands

[DEACTIVATESTC](#)

[EXCLUDE](#)

[PROTECT](#)

[REACTIVATESTC](#)

[REPROTECT](#)

[SHOW=LIST](#)

[SHOW=PROTECTIONS](#)

BEGINDATE, ENDDATE commands

BEGINDATE and ENDDATE control the date and time ranges on z/OS Change Tracker reporting.

Syntax

```
BEGINDATE=begin-date  
ENDDATE=end-date
```

➤ BEGINDATE= — *begin-date* ➤

➤ ENDDATE= — *end-date* ➤

Parameters

BEGINDATE=*begin-date*

A valid date in the format yyyy/mm/dd_hh:mm. If hh:mm is not specified, the value is defaulted to 00:00.

ENDDATE=*end-date*

A valid date in the format yyyy/mm/dd_hh:mm. If hh:mm is not specified, the value is defaulted to 00:00.

Usage

BEGINDATE and ENDDATE are used with reporting functions such as SHOW=LOG or SHOW=ACTIONS=LIST.

These commands are mutually exclusive with the DATE command that can be used to select a particular day to report on.

Example 1: Log activities for a 7 day window

```
BEGINDATE=2021/01/01_00:00  
ENDDATE=2021/01/07_23:59  
SHOW=LOG
```

Example 2: Log rename activities for all protected data sets in a 7 day window

```
BEGINDATE=2021/01/01_00:00  
ENDDATE=2021/01/07_23:59  
OPTION=REN /* OPTION=ADD or UPD, DEL or ALL may be specified */  
SHOW=PROTECTIONS  
SHOW=ACTIONS=LIST
```

Related commands

[DATE](#)

[OPTION](#)

[SHOW=ACTIONS=LIST](#)

[SHOW=LOG](#)

CHECKIN command

The CHECKIN command releases a checked out member.

Syntax

```
CHECKIN=(DSN=data-set-name
          ,MEM=member-name
          [,COMMENT=comment-line]
          [,VOL=volume-serial])
```

➤ CHECKIN= — (— DSN= *data-set-name* — ,MEM= *member-name* ➤

└─ ,COMMENT= *comment-line* ─┘ └─ ,VOL= *volume-serial* ─┘) ➤

Parameters

DSN=*data-set-name*

The name of the data set containing the member to be checked in.

MEM=*member-name*

The name of the member to be checked in. It must be a member that was previously checked out. Wildcard characters are not accepted.

comment=*comment-line*

This parameter is optional. The comment can include the reason for the check in such as a work item number and the status of the work item. A CKI prefix in the comment can be used to identify the type of operation, for example ("CKI:user comment for the checkin goes here"). The comment is limited to 64 characters.

VOL=*volume-serial*

For uncataloged data sets, the volume serial number of the volume on which the data set resides. Omit this parameter if the data set is cataloged.

Usage

This command is available for use by the Administrator only.

An ENDCHECK command must follow the last CHECKOUT or CHECKIN command in a step.

A TRIGGERREFRESH should be the last command in job following CHECKIN and CHECKOUT commands. This will update the started task.

Example 1: Check in two members that were previously checked out

```
CHECKIN=(DSN=SYS1.PARMLIB,
          MEM=IEASSN00,
          COMMENT=("CKI:", "Updated CICS"))
*
CHECKIN=(DSN=SYS1.PARMLIB,
          MEM=IEASSN01,
          COMMENT=("CKI:", "Changed IRLM"))
*
```


Usage

This command is available for use by the Administrator only.

An ENDCHECK command must follow the last CHECKOUT or CHECKIN command in a step.

The CHECKOUT command is used to check out a member of a locked data set, which allows you to modify the member even though the data set is locked.

The CHECKOUT command can only be used with data sets protected with the LOCK attribute.

Example 1: Check out a member of a protected data set to a user ID

```
CHECKOUT=(DSN=SYS1.PARMLIB,  
          VOL=RES001,  
          MEMBER=IEASSN00,  
          COMMENT=("CKO: Update for CICS"),  
          USERID=IBMU19)  
ENDCHECK  
TRIGGERREFRESH
```

Example 2: Check out members of a protected data set to a RACF group

```
CHECKOUT=(DSN=SYS1.PROCLIB,  
          MEMBER=CICSA,  
          COMMENT=("CKO: Available to CICS group"),  
          RACFGROUP=CICSGRP)  
CHECKOUT=(DSN=SYS1.PROCLIB,  
          MEMBER=DB2IRLM,  
          COMMENT=("CKO: Available to DB2 group"),  
          RACFGROUP=BD2GRP)  
ENDCHECK  
TRIGGERREFRESH
```

Example 3: Multi line comment in the CHECKOUT command

```
CHECKOUT=(MEM=CYGCOPY,  
          DSN=DEV.PRODUCT.INSTALL,  
          USERID=IBMUSER,  
          COMMENT=("CKO: this is the first 32 bytes",  
                  "this is the 2nd part - 32 bytes."))
```

Related commands

[CHECKIN](#)

[ENDCHECK](#)

[TRIGGERREFRESH](#)

COMPARE command

The COMPARE command compares members of two data sets.

Syntax

```
COMPARE
```

➤ COMPARE ➤

Parameters

None.

Usage

The DSN1 and DSN2 commands are required to specify the data sets to be compared. The VOL1 and VOL2 commands are required to specify the volumes when comparing uncataloged data sets.

The output is influenced by the setting of the OPTION command. The possible reporting options are SAME, DIFF, MISMATCH, MATCH, MODS, and ALL.

Example 1: COMPARE multiple pairs of libraries – Detailed MODS comparison

```
OPTION=MODS
*
DSN1=SYS1.PROCLIB.NEW
DSN2=SYS1.PROCLIB.OLD
COMPARE
```

Example 2: COMPARE multiple pairs of libraries identifying differences only

```
OPTIONS=DIFF
*
NEWDSN=SYS1.PARMLIB
OLDDSN=SYS2.PARMLIB
COMPARE
*
NEWDSN=SYS1.LINKLIB
OLDDSN=SYS2.LINKLIB
COMPARE
```

Example 3: COMPARE multiple pairs of libraries residing on a primary volume and an alternate volume

```
OPTIONS=MODS
*
VOL1=RES001
DSN1=SYS1.PARMLIB
VOL2=RES002
DSN2=SYS1.PARMLIB
COMPARE
*
DSN1=SYS1.PROCLIB
DSN2=SYS1.PROCLIB
COMPARE
```

Example 4: Using the REJECT command during the COMPARE operation

```
OPTIONS=CHANGEONLY
REJECT=(DSN=IBMUSER.PRS3.LINKLIB, MEM=CYGDEFLT, VOL=*)
REJECT=(DSN=IBMUSER.PRS2.LINKLIB, MEM=CYGDEFLT, VOL=*)
REJECT=(DSN=IBMUSER.PRS3.SRL, VOL=*, MEM=COMP*)
REJECT=(DSN=IBMUSER.PRS2.SRL, VOL=*, MEM=COMP*)
. . .
DSN1=IBMUSER.PRS3.LINKLIB
DSN2=IBMUSER.PRS2.LINKLIB
COMPARE
DSN1=IBMUSER.PRS3.SRL
DSN2=IBMUSER.PRS2.SRL
COMPARE
```

Related commands

DSN1, DSN2

OPTION

VOL1, VOL2

COPYDIFF command

The COPYDIFF command reconciles the differences between two partitioned data sets.

Syntax

```
COPYDIFF
```

➡ COPYDIFF ➡

Parameters

None.

Usage

The DSN1 and DSN2 commands are required to specify the data sets to be reconciled. The source data set is specified by the DSN1 command. The target data set is specified by the DSN2 command. The source data set is not changed during reconciliation, only the target data set is modified.

The VOL1 and/or VOL2 commands are required to specify the volume when uncataloged data sets are involved.

The reconciled data sets become identical in their member configuration and content.

The data sets to be reconciled can be PDS, PDSE, SOURCE, or LOAD.

The TESTRUN command can be used to request a simulation run.

The DSN1, DSN2 and COPYDIFF commands can be repeated to reconcile multiple data sets.

When DELETOMEM=YES precedes this command, target members missing from the source are deleted.

Example 1: Reconcile two pairs of data sets

```
DELETOMEM=YES
DSN1=SYS1.PARMLIB
VOL1=WORK01
DSN2=SYS2.PARMLIB
VOL2=WORK02
COPYDIFF
*
DELETOMEM=NO
DSN1=SYS1.PROCLIB
DSN2=SYS2.PROCLIB
COPYDIFF
```

Related commands

[DSN1, DSN2](#)

[VOL1, VOL2](#)

[TESTRUN](#)

DATE command

The DATE (or SINCE) command is used to refer to a previous relative date.

Syntax

```
DATE=Today-number-of-days
```

➤ DATE= — *TODAY-number-of-days* ➤

Parameters

DATE=TODAY-*number-of-days*

The *number-of-days* is the number of days before the current date.

Usage

The command precedes other commands such as SHOW=ACTIONS. The following examples show actions for a specific relative day.

Example 1: Referring to several days in the past

```
DATE=TODAY-1 <= TODAY-1 refers to yesterday
DATE=TODAY-3 <= TODAY-3 refers to three days ago
SINCE=TODAY-7 <= since 7 days ago (weekly report)
```

Example 2: Show all delete actions recorded by the STC 7 days ago

```
DATE=TODAY-7
OPTION=DEL
*
SHOW=ACTIONS=(DSN=SYS1.PARMLIB, MEM=*)
SHOW=ACTIONS=(DSN=SYS1.PROCLIB, MEM=*)
```

Related commands

[SHOW=ACTIONS](#)

[WRITETOFILE](#)

DD1SHOW, DD2SHOW commands

The DD1SHOW and DD2SHOW commands report the contents of external token files.

Syntax

```
DD1SHOW=(DSN=pattern, VOL=pattern) OR DD1SHOW
DD2SHOW=(DSN=pattern, VOL=pattern) OR DD2SHOW
```

Or

```
DD1SHOW
DD2SHOW
```

➤ DD1SHOW= — (— DSN=*pattern* — ,VOL= *pattern* —) ➤

➤ DD2SHOW= — (— DSN=*pattern* — ,VOL= *pattern* —) ➤

Parameters

DSN=*pattern*

A pattern of data sets that will be reported. A wildcard * character can be used in the pattern. The default behavior is to report all data sets.

VOL=*pattern*

A pattern of volumes containing data sets that will be reported. A wildcard * character can be used in the pattern. The default behavior is to report all cataloged data sets.

Usage

The DD1SHOW and DD2SHOW commands display the contents of the token files associated with the DD1 and DD2 DD statements respectively. They can be used on their own or with the LIST command to create input for the REMOTEGROUPCOMP command.

Example 1: Show the contents of a token file for data sets matching SYS1.* pattern

```
//DD1 DD DISP=SHR,DSN=IBMUSER.TOKENS1
//CYGIN DD *
DD1SHOW=(DSN=SYS1.*)
```

Example 2: Show the contents of two token files

```
//DD1 DD DISP=SHR,DSN=IBMUSER.TOKENS1
//DD2 DD DISP=SHR,DSN=IBMUSER.TOKENS2
//CYGIN DD *
DD1SHOW
DD2SHOW
```

Example 3: Populate LISTs of tokenized data sets in memory to COMPARE

```
//DD1 DD DISP=SHR,DSN=IBMUSER.TOKENS1
//DD2 DD DISP=SHR,DSN=IBMUSER.TOKENS2
//CYGIN DD *
OPTIONS=DIFF
*
DD1SHOW
LIST
*
DD2SHOW
LIST
*
REMOTEGROUPCOMP
```

Related commands

[LIST](#)

[REMOTEGROUPCOMP](#)

DEACTIVATESTC command

The DEACTIVATESTC command deactivates the started task.

Syntax

```
DEACTIVATESTC
```

OR

```
DECATIVATESTC=time-period-seconds
```

➤ DEACTIVATESTC ➤

➤ DEACTIVATESTC= — *time-period-seconds* ➤

Parameters

DEACTIVATESTC=*time-period-seconds*

The number of seconds for which the started task should deactivate. The started task will reactivate when this period expires.

Usage

This command is available for use by the Administrator only.

Deactivating the started task is an alternative to stopping it. It may be used to perform specific functions such as auto archiving or surveillance which require a quiescent environment during the run.

Reactivate the started task following the task that required deactivating it. If a command following the DEACTIVATESTC cancels for any reason, the subsequent REACTIVATESTC command is not executed. Ensure that REACTIVATESTC runs in a separate job to get the started task activated again. Using the DEACTIVATESTC=*time-period-seconds* form of the command is a safer way to deactivate and ensure that normal operation of the started task is resumed.

Example 1: DEACTIVATESTC during AUTOARC

```
DEACTIVATESTC
AUTOARC
REACTIVATESTC
```

Example 2: Use of the DEACTIVATESTC during the SURV function

```
SHOW=PROTECTIONS
*
DEACTIVATESTC
SURV=(LIST,CHG=YES)
REACTIVATESTC
```

Example 3: Set a two minute time limit on the started task remaining idle

```
SHOW=PROTECTIONS
DEACTIVATESTC=120
SURV=(LIST,CHG=YES)
```

Related commands

[AUTOARC](#)

[OPTION=HALTOBJ](#)

[REACTIVATESTC](#)

[SURVEILLANCE](#)

DELETEBACKUP command

The DELETEBACKUP command deletes backups from the live system.

Syntax

```
DELETEBACKUP=(DSN=data-set-name
               ,MEM=member-name
               ,VER=backup-version
               [,VOL=volume-serial])
```

►► DELETEBACKUP= — (— DSN= — *data-set-name* — ,MEM= *member-name* —
 ► ,VER= *version-number* —) —
 └─ ,VOL= *volume-serial* ─┘

Parameters

DSN=*data-set-name*

The name of the data set containing members whose backups are to be deleted. This must be a protected data set.

MEM=*member-name*

The name of the member whose backups are to be deleted. Wildcards are not allowed.

VER=*version-number*

Version number of the backup to be deleted. Specify an asterisk (*) to request a delete of all backed-up versions of the given member.

VOL=*volume-serial*

For uncataloged data sets, the volume serial number of the volume on which the data set resides. Omit this parameter if the data set is cataloged.

Usage

This command is available for use by the Administrator only.

Deleting unnecessary backups can keep the live backup system smaller.

Example 1: Delete the last three backup versions of a given member

```
DELETEBACKUP=(DSN=SYS1.PARMLIB,  
               MEM=IEASSN00,  
               VER=-2)  
DELETEBACKUP=(DSN=SYS1.PARMLIB,  
               MEM=IEASSN00,  
               VER=-1)  
DELETEBACKUP=(DSN=SYS1.PARMLIB,  
               MEM=IEASSN00,  
               VER=0)
```

Related commands

MEMBERBACKUP

MEMBERRECOVER

DSN1, DSN2 commands

The DSN1 and DSN2 commands specify source and target data sets for other commands such as COPYDIFF or COMPARE.

Syntax

```
DSN1=(source-data-set-name)  
DSN2=(target-data-set-name)
```

➤ DSN1= — (— *source-data-set-name* —) ➤

➤ DSN2= — (— *target-data-set-name* —) ➤

Parameters

DSN1=*source-data-set-name*

Specify the name of the source data set to be operated on by subsequent commands.

DSN2=*target-data-set-name*

Specify the name of the target data set to be operated on by subsequent commands.

Usage

The commands are followed by other commands that need to operate on the source and target data sets such as COMPARE or COPYDIFF.

Related commands

[COMPARE](#)

[COPYDIFF](#)

ENDCHECK command

The ENDCHECK command is used to signify the end of a CHECKOUT or CHECKIN function.

Syntax

```
ENDCHECK
```

➤ ENDCHECK ➤

Parameters

None.

Usage

This command must be specified at the end of one or multiple CHECKIN, CHECKOUT as well as REMOVECHECKOUT commands. Repository counts are updated when this command is issued.

To avoid conflicts, the ENDCHECK command does not refresh the protection list. If multiple steps contain ENDCHECK commands, run the TRIGGERREFRESH command as a last step to refresh the started task.

Example 1: ENDCHECK following CHECKOUT

```
*
CHECKOUT=(MEM=PDSA001,
          DSN=IBMUSER.QA.PDSA,
          USERID=IBMUSER,
          COMMENT=('CKO: ','Ticket #100'))
*
CHECKOUT=(MEM=PDSA002,
          DSN=IBMUSER.QA.PDSA,
          USERID=IBMUSER,
          COMMENT=('CKO: ','Ticket #101'))
*
ENDCHECK
TRIGGERREFRESH
```

Example 2: ENDCHECK following REMOVECHECKOUT

```
OBJECT=(DSN=IBMUSER.TEST.PDSA)
REMOVECHECKOUT
ENDCHECK
TRIGGERREFRESH
```

Example 3: Multiple steps with ENDCHECK followed by a TRIGGERREFRESH step

```
//STEP1
//CYGIN DD *
CHECKOUT=(DSN=SYS1.PARMLIB, MEM=MEM1, USERID=IBMUSER)
CHECKOUT=(DSN=SYS1.PARMLIB, MEM=MEM2, USERID=IBMUSER)
...
```

```

ENDCHECK
//*
//STEP2
//CYGIN DD *
CHECKOUT=(DSN=SYS1.PROCLIB, MEM=MEM1, USERID=IBMUSER)
CHECKOUT=(DSN=SYS1.PROCLIB, MEM=MEM2, USERID=IBMUSER)
...
ENDCHECK
//*
//STEPn
//CYGIN DD *
TRIGGERREFRESH
//*

```

Related commands

[CHECKIN](#)

[CHECKOUT](#)

[REMOVECHECKOUT](#)

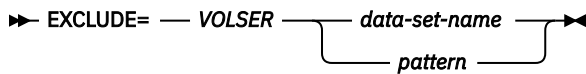
[TRIGGERREFRESH](#)

EXCLUDE command

The EXCLUDE command is used to exclude specific data sets or data set patterns from a resolved list.

Syntax

```
EXCLUDE=VOLSER, data-set-name | pattern
```



Parameters

EXCLUDE=*data-set-name***|***pattern*

A cataloged data set name or a pattern of data sets to be excluded from the list. Wildcards can be used to specify the pattern.

The name of a data set or a pattern of data sets to be tokenized or compared.

Usage

EXCLUDE commands can be specified either before or after a list of PATTERN commands.

Example 1: Show a list of data sets selected by PATTERN except for excluded data sets

```

EXCLUDE=SYS1.LOGREC
EXCLUDE=SYS1.BROADCAST
EXCLUDE=SYS1.DUMP*
*
PATTERN=SYS1.*
SHOW=LIST

```

Related commands

[EXPLICIT](#)

[INCLUDE](#)

[PATTERN](#)

SHOW=LIST

EXPLICIT command

The EXPLICIT command selects a LIST of cataloged data sets that will be used by a subsequent command.

Syntax

```
EXPLICIT=data-set-name
```

➡ EXPLICIT= — *data-set-name* →

Parameters

EXPLICIT=*data-set-name*

The name of a specific cataloged data set. Wildcards are not allowed.

Usage

Any command that operates on a LIST (such as the TOKEN, PROTECT, COMPARE and the GROUPCOMPARE command) can work with the LIST built by the EXPLICIT commands.

Example 1: Tokenize a list of data sets

The SHOW=LIST command shows the populated list. The TOKEN command tokenizes each data set in the LIST.

```
EXPLICIT=SYS1.PARMLIB
EXPLICIT=SYS1.PROCLIB
EXPLICIT=SYS1.VTAMLIB
...
SHOW=LIST
TOKEN=LIST
```

Example 2: Compare two groups of cataloged data sets

Use of the SHOW=LIST command is required to build the two lists.

```
OPTION=DIFF
*
EXPLICIT=SYS1.PARMLIB
EXPLICIT=SYS1.PROCLIB
EXPLICIT=SYS1.VTAMLIB
SHOW=LIST
*
INCLUDE=(VOL=RES001,DSN=SYS1.PARMLIB)
INCLUDE=(VOL=RES001,DSN=SYS1.PROCLIB)
INCLUDE=(VOL=RES001,DSN=SYS1.VTAMLIB)
SHOW=LIST
*
GROUPCOMPARE
```

Related commands

GROUPCOMPARE

TOKEN

GROUPCOMPARE command

The GROUPCOMPARE command compares two groups of data sets.

Syntax

```
GROUPCOMPARE
```

➡ GROUPCOMPARE ➡

Parameters

None.

Usage

INCLUDE, EXPLICIT or PATTERN commands need to be used to build the groups for comparison. The SHOW=LIST command builds up the lists of data sets in each group to be compared.

Data set names that are not matched between the two groups are reported as unmatched data sets in the final report.

The report contents can be adjusted using the OPTION command. For example, OPTION=DIFF requests only members with different contents are reported.

STRREPL command can be used to map data set names for comparison.

Example 1: Compare identically named data sets that reside on different volumes

```
OPTION=DIFF
*
INCLUDE=(VOL=RES001,DSN=SYS1.*)
SHOW=LIST
*
INCLUDE=(VOL=RES002,DSN=SYS1.*)
SHOW=LIST
*
GROUPCOMPARE
```

Example 2: Compare multiple data sets that reside on different volumes

```
OPTION=DIFF
*
INCLUDE=(VOL=RES001,DSN=SYS1.*)
INCLUDE=(VOL=RES001,DSN=SYS2.*)
SHOW=LIST
*
INCLUDE=(VOL=RES002,DSN=SYS1.*)
INCLUDE=(VOL=RES002,DSN=SYS2.*)
SHOW=LIST
*
GROUPCOMPARE
```

Example 3: Compare cataloged with uncataloged data sets

```
OPTION=DIFF
*
PATTERN=SYS1.PROC*
PATTERN=SYS2.PROC*
SHOW=LIST
*
INCLUDE=(VOL=RES001, DSN=SYS1.PROC*)
INCLUDE=(VOL=RES002, DSN=SYS2.PROC*)
SHOW=LIST
GROUPCOMPARE
```

```
OPTION=DIFF
*
STRREPL=('SYS1','SYS7')
*
EXPLICIT=SYS1.PARMLIB
EXPLICIT=SYS1.PROCLIB
EXPLICIT=SYS1.VTAMLIB
SHOW=LIST
*
EXPLICIT=SYS7.PARMLIB
EXPLICIT=SYS7.PROCLIB
EXPLICIT=SYS7.VTAMLIB
SHOW=LIST
*
GROUPCOMPARE
```

INCLUDE
PATTERN
SHOW=LIST
STRINGREPLACE

The **INCLUDE** command selects a pattern of uncataloged data sets that can be tokenized or compared.

```
INCLUDE=(DSN=pattern
           ,VOL=volume-serial
           [,TYPE=ALL])
```

```
INCLUDE=(VOL=RES001,DSN=SYS1.*)
INCLUDE=(VOL=RES001,DSN=SYS2.*)
*
```

```
EXCLUDE=SYS1.PROFILE.*
EXCLUDE=SYS1.FONTLIB
*
SHOW=LIST
TOKEN=(LIST)
```

Related commands

[EXPLICIT](#)

[PATTERN](#)

[TOKEN](#)

LIST command

The LIST command lists the contents of a pre-generated token file representing an zFS path.

Syntax

```
LIST
```

➤ LIST ➤

Parameters

None.

Usage

The tokenized file must be loaded using a LOADDD command before its contents can be displayed using the LIST command.

Example 1: Use the LIST command to display the contents of a token file

The LOADDD command is used to point to the token file of interest specified in DD1.

```
LOADDD=DD1
LIST
```

Related commands

[DD1SHOW](#), [DD2SHOW](#)

[LOADDD](#)

LOADDD command

The LOADDD command loads the contents of a z/OS UNIX Token file into memory for subsequent use by other commands such as LIST or PATHCOMPARE.

Syntax

```
LOADDD=ddname
```

➤ LOADDD= — *ddname* ➤

Parameters

ddname

The label of a DD JCL statement in the job which references a z/OS UNIX Token file.

Usage

This command is used to display or compare the contents of z/OS UNIX Token files.

Example 1: Load contents of token file

```
//TOKFILE DD DISP=SHR,DSN=CYGUSS.TOKENS.WEB.ORIG
//CYGIN DD *
LOADDD=TOKFILE
LIST
```

Related commands

[LIST](#)

[PATHCOMPARE](#)

[USSREPORT](#)

LOGCOMMENT command

The LOGCOMMENT command associates a comment with a following command.

Syntax

```
LOGCOMMENT
```

➡ LOGCOMMENT ➡

Parameters

None.

Usage

This command can be used with the PROTECT, UNPROTECT, and REPROTECT commands to associate a comment with the action.

Related commands

[PROTECT](#)

[REPROTECT](#)

[UNPROTECT](#)

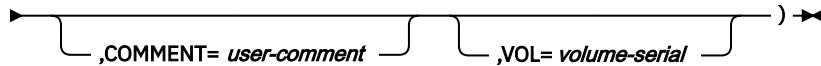
MEMBERBACKUP command

The **MEMBERBACKUP** command backs up a data set member to the repository.

Syntax

```
MEMBERBACKUP=(DSN=data-set-name
               ,MEM=member-name
               [,COMMENT=user-comment]
               [,VOL=volume-serial])
```

►► MEMBERBACKUP= — (— DSN= — *data-set-name* — ,MEM= *member-name* →



Parameters

DSN=*data-set-name*

The name of the data set containing the member to be backed up. This must be a protected data set.

MEM=*member-name*

The name of the member to be backed up. Wildcards are not allowed.

COMMENT=*user-comment*

A description of the reason for the backup. A suggested format is 'USR: ', 'Comment Text'. This helps users to identify their own user-initiated backups during recovery.

VOL=volume-serial

For uncataloged data sets, the volume serial number of the volume on which the data set resides. Omit this parameter if the data set is cataloged.

Usage

This command is available for use by the Administrator only.

Members of protected data sets are also automatically backed up when they are modified.

Note: z/OS Change Tracker will not back up a PDS or PDSE member that has more than 500,000 records.

Example 1: Back up two members in a data set

```
MEMBERBACKUP=(DSN=SYS1.PARMLIB,
               VOL=SYSRES, MEM=IEASSN00,
               COMMENT=('USR:', 'CICS changes worked o.k.'))
*
MEMBERBACKUP=(DSN=SYS1.PARMLIB,
               MEM=IEASSN01,
               COMMENT=('USR:', 'Changed IRLM'))
```

Related commands

DELETEBACKUP

MEMBERRECOVER

MEMBERCOMMENT command

The MEMBERCOMMENT command retrieves the comment associated with a member backup version.

Syntax

```
MEMBERCOMMENT=(DSN=data-set-name
                ,MEM=member-name
                ,VER=version-number
                [,VOL=volume-serial])
```

►► MEMBERCOMMENT= — (— DSN=*data-set-name* — ,MEM= *member-name* —>

 ► ,VER= *version-number* —————) ◄◄

 , VOL= *volume-serial* —

Parameters

DSN=*data-set-name*

The name of the data set containing the member. This must be a protected data set.

MEM=*member-name*

The name of the member.

VER=*version-number*

The version number of the member.

VOL=*volume-serial*

The volume of the data set containing the member. This is only specified for data sets that are protected as uncataloged data sets.

Usage

When a data set member is backed up using the MEMBERBACKUP command, or an automatic backup is taken as a result of the CSAVE command, the supplied comment is saved in the repository along with the backup. This command is used to retrieve this comment.

Example

```
MEMBERCOMMENT=(DSN=USR1.QA.PDSC,
                ,MEM=PDSC001
                ,VER=0)
```

Related commands

[MEMBERBACKUP](#)

MEMBERRECOVER command

The MEMBERRECOVER command recovers a member of a protected resource from the live backup.

Syntax

```
MEMBERRECOVER=(DSN=data-set-name
                ,MEM=member-name
                ,OUTDD=DD1
                ,VER=version-number
                [,VOL=volume-serial])
```



```
VER=0,  
OUTDD=DD1)
```

Related commands

[MEMBERBACKUP](#)

Related information

CYGA* sample jobs.

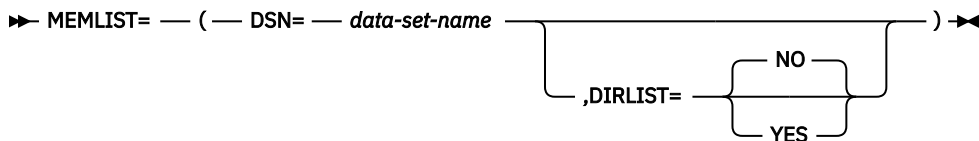
CYGH* sample jobs.

MEMLIST command

The MEMLIST command builds a list of members in a data set.

Syntax

```
MEMLIST=(DSN=data-set-name  
[,DIRLIST=YES|NO])
```



Parameters

DSN=*data-set-name*

The name of a cataloged data set whose members are to be listed.

DIRLIST=YES|NO

This parameter is optional.

When DIRLIST=YES, the entries are displayed, otherwise the list is built internally only.

DIRLIST=NO is the default.

Usage

This command can be combined with the OPTION=ORPHAN command to identify and report member aliases including any orphaned aliases. In this case, DIRLIST=YES must be specified.

Example 1: List members of two data sets

```
MEMLIST=(DSN=IBMUSER.TEST.PDS,  
          DIRLIST=YES)  
*  
MEMLIST=(DSN=IBMUSER.TEST.PDSE,  
          DIRLIST=YES)
```

Example 2: Report member aliases including any orphaned aliases

```
OPTION=ORPHAN  
MEMLIST=(DSN=SYS1.LINKLIB,  
          DIRLIST=YES)
```

Example 3: Report aliases for all data sets on a specific DASD volume

```
OPTION=ORPHAN
INCLUDE=(VOL=RES001,DSN=*)
SHOW=LIST
*
MEMLIST=(LIST,
          DIRLIST=YES)
```

Related commands

OPTION=ORPHAN

OBJECT command

The OBJECT command limits action to a selected object.

Syntax

```
Object=(DSN=data-set-name
        [,VOL=volume-serial])
```

►► OBJECT= — (**— DSN=** — *data-set-name* —) **►►**

Parameters

DSN=*data-set-name*

The name of the data set to which further commands apply.

VOL=volume-serial

For uncataloged data sets, the volume serial number of the volume on which the data set resides. Omit this parameter if the data set is cataloged.

Usage

This command can only be used in combination with other commands to limit the action of those commands to the specified data set. For example, if it is combined with REMOVECHECKOUT command, then only records for the specified data set are removed.

Can be used with the HALTOBJ and UNHALTOBJ to suspend and resume the tracking of a particular data set. This can be useful to allow for a temporary modification to a locked data set to take place without having to shut down or deactivate the started task.

The **OBJECT** command can only be used to specify a single data set. If multiple **OBJECT** commands are issued in one job, only the latest **OBJECT** command takes effect.

Example 1: Suspend the tracking of an object

```
OPTION=HALTOBJECT
OBJECT=(DSN=IBMUSER.TEST.PDS)
```

Example 2: Generate a diagnostic report about an object (only if requested by IBM)

```
OPTION=DIAGNOSE
OBJECT=(DSN=SYS1.PARMLIB)
```

Related commands

DEACTIVATESTC

OPTION=DIAGNOSE

OPTION=HALTOBJ

OPTION=UNHALTOBJ

REMOVECHECKOUT

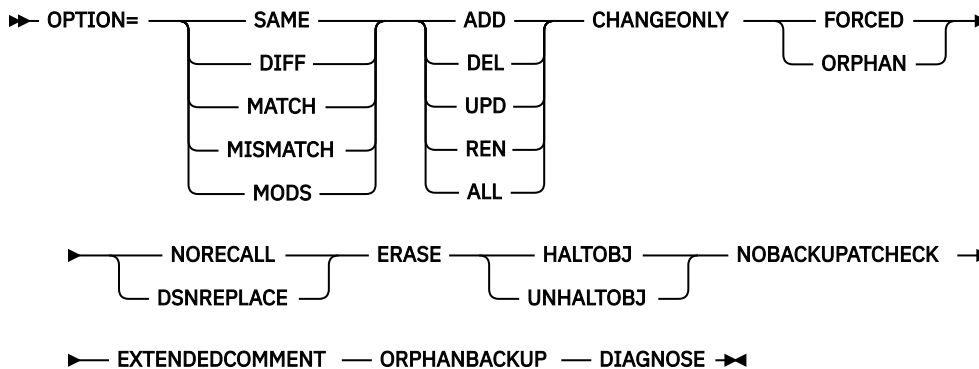
OPTION command

The OPTION command provides various processing conditions.

Syntax

```
OPTION=SAME | DIFF | MATCH | MISMATCH | MODS |  
        ADD | DEL | UPD | REN | ALL |  
        CHANGEONLY |  
        FORCED | ORPHAN |  
        NORECALL | DSNREPLACE |  
        ERASE |  
        HALTOBJ | UNHALTOBJ |  
        NOBACKUPCHECK |  
        EXTENDEDCOMMENT |  
        ORPHANBACKUP |  
        DIAGNOSE
```

Note: Only one parameter per command can be specified.



Parameters

OPTION=ADD

List the actions that relate to adding a new member in a protected data set.

OPTION=ALL

List all the actions that relate to a member in a protected data set. The actions are ADD, DEL, UPD, and REN.

OPTION=CHANGEONLY

Only changed data sets and changed, new, or deleted members of the data sets are reported. Identical data sets or data sets that have not changed are not included in the report.

OPTION=DEL

List the actions that relate to deleting a member in a protected data set.

OPTION=DIAGNOSE

This parameter is only needed for problem diagnosis when requested by IBM.

OPTION=DIFF

Reports the members that have different content only.

OPTION=DSNREPLACE

This parameter directly associates a source data set name with a target data set name without using string search. This is used to provide a particular mapping of source to target without a general pattern mapping operation that may have unwanted side effects.

OPTION=ERASE

Delete selected records.

OPTION=EXTENDEDCOMMENT

Display extended comments. By default, only the first 32 bytes of the comment is displayed on the line.

OPTION=FORCED

This parameter allows certain conditions to be ignored. For example, use FORCED to remove token records when the original data set no longer exists.

OPTION=HALTOBJECT

Suspend the tracking of an object. The tracking can be resumed by specifying the OPTION=UNHALTOBJ command.

OPTION=MATCH

Report only the members that have matching member names in the data sets being compared.

OPTION=MISMATCH

Report only the members that do not have matching member names in the data set being compared.

OPTION=MODS

Report all of the comparison conditions. This is equivalent to combining OPTION=SAME, OPTION=DIFF, OPTION=MATCH, and OPTION=MISMATCH in a single consolidated report.

OPTION=NOBACKUPCHECK

Suppresses the creation of backup at CHECKOUT and CHECKIN.

OPTION=NORECALL

This parameter does not recall archived data sets. The default behavior is to recall archived data sets that will be protected, tokenized, or compared.

OPTION=ORPHAN

Report and identify the aliases.

OPTION=ORPHANBACKUP

Report the orphaned backups. Orphaned backups do not have token records in the z/OS Change Tracker Data File.

OPTION=REN

List the actions that relate to renaming a member in a protected data set.

OPTION=SAME

Report only the members that have the same content.

OPTION=UNHALTOBJ

Resume the tracking of a halted object.

OPTION=UPD

List the actions that relate to updating a member in a protected data set.

Usage

OPTION=ADD | DEL | UPD | REN | ALL and OPTION=EXTENDEDCOMMENT are used with the SHOW=ACTIONS command.

OPTION=ERASE and OPTION=ORPHANBACKUP are used with SHOW=BACKUP command.
OPTION=ERASE is available to the Administrator only.

OPTION=HALTOBJ and UNHALTOBJ are used with the OBJECT command.

OPTION=NORECALL is used with the PATTERN command.

OPTION=ORPHAN is used with the MEMLIST command.

OPTION=SAME | DIFF | MATCH | MISMATCH | MODS are used with comparison reports.

OPTION=DSNREPLACE is used with the STRREPLACE command in data set comparison or change identification reports.

OPTION=CHANGEONLY is used with commands that compare data sets or zFS paths such as COMPARE, PATHCOMPARE or TOKEN(..., MODS).

Related commands

[COMPARE](#)

[OBJECT](#)

[MEMLIST](#)

[PATHCOMPARE](#)

[PATTERN](#)

[SHOW=ACTIONS](#)

[SHOW=BACKUP](#)

[STRINGREPLACE](#)

[TOKEN\(..., MODS\)](#)

[UNPROTECT](#)

PATHCOMPARE command

The PATHCOMPARE command compares similarities and differences between any two zFS or z/OS UNIX paths.

Syntax

```
PATHCOMPARE
```

➤➤ PATHCOMPARE ➤➤

Parameters

None.

Usage

The two paths to be compared are specified prior to issuing the PATHCOMPARE command using one of the following methods:

- SYSUT1 and SYSUT2 commands specifying two different paths.
- A SYSUT1 command specifying a path and a LOADDD command specifying a DD statement for a previously created token file. This token file is created using the USSTOKEN command to contain the tokens for a specific path.
- Two LOADDD commands specifying DD statements for two previously created token files.

The USSPATHREPL command can be used to map two path names that are different so that they are compared against each other. The path names are case sensitive.

The OPTION command is used to specify the type of comparison to be performed.

Example 1: Comparing two paths mapped using the USSPATHREPL command

```
OPTION=MODS
USSPATHREPL=(' /u/jones', '/u/smith')
*
SYSUT1=PATH=' /u/jones/*.*'
SYSUT2=PATH=' /u/smith/*.*'
*
PATHCOMPARE
```

Example 2: Detect changes introduced over time, to pre-tokenized paths

Files in the path '/u/jones/*.*' are internally tokenized and compared against the same files that were tokenized sometime in the past (stored in the pre-generated token file pointed to by DD1).

```
//DD1 DD DISP=SHR,DSN=IBMU19.TOKEN.JONES
OPTION=DIFF
SYSUT1=PATH=' /u/jones/*.*'
LOADDD=DD1
```

Example 3: Comparing two token files

Two token files nominated by the DD1 and DD2 cards were previously created from the /RAC003 and /RAC004 paths respectively. Using the LOADDD commands, the contents of the token files pointed to by DD1 and DD2 are loaded. The LIST command then builds a list of the path. The USSPATHREPL command is used to translate all paths starting with /RAC003 to the path /RAC004. Once two paths are matched, the files in those paths are compared.

```
//DD1 DD DISP=SHR,DSN=IBMUSER.USS.TOKEN1
//DD2 DD DISP=SHR,DSN=IBMUSER.USS.TOKEN2
//CYGIN DD *
OPTION=MODS
USSPATHREPL=(' /RAC003', '/RAC004')
LOADDD=DD1
LIST
*
LOADDD=DD2
LIST
*
PATHCOMPARE
```

DIFF, MISMATCH, and SAME are other options that you can specify.

Example 4: Reporting the z/OS UNIX file changes using CHANGEONLY option

```
//STEP1CMP EXEC PGM=CYGMAIN,REGION=0M
//DD1 DD DISP=SHR,DSN=IBMUSER.CYGUSS.NEWTOKEN
//DD2 DD DISP=SHR,DSN=IBMUSER.CYGUSS.OLDTOKEN
//CYGIN DD *
OPTION=CHANGEONLY
LOADDD=DD1 new token file loaded in memory LOADDD=DD2 old token file loaded in memory
*
PATHCOMPARE
```

Related commands

[LOADDD](#)

[OPTION](#)

[SAVEDD](#)

[SYSUT1 and SYSUT2](#)

[USSPATHREPL](#)

PATTERN command

The PATTERN command resolves a group of data sets from the catalog using a pattern.

Syntax

```
PATTERN=data-set-pattern
```

➤ PATTERN= — *data-set-pattern* ➤

Parameters

PATTERN=*data-set-pattern*

A data set pattern as used in ISPF 3.4. Quotes should not be used to surround the data set name and a wildcard * character can be used to indicate a requested match to any sequence of characters.

Usage

The PATTERN command can be used to resolve a group of data sets from the catalog. Its output is an input to following commands that operate on the resolved data sets.

Any archived data sets are recalled from the archive unless OPTION=NORECALL is specified.

Example 1: Resolve and display some data sets from the catalog

```
PATTERN=SYS1.PARMLIB  
PATTERN=SYS1.LINKLIB PATTERN=PAYROLL.*  
PATTERN=ONLINE.*  
PATTERN=*.*.LINKLIB  
SHOW=LIST
```

Related commands

[OPTION=NORECALL](#)

[SHOW=LIST](#)

[TOKEN=LIST](#)

[PROTECT=LIST](#)

Related information

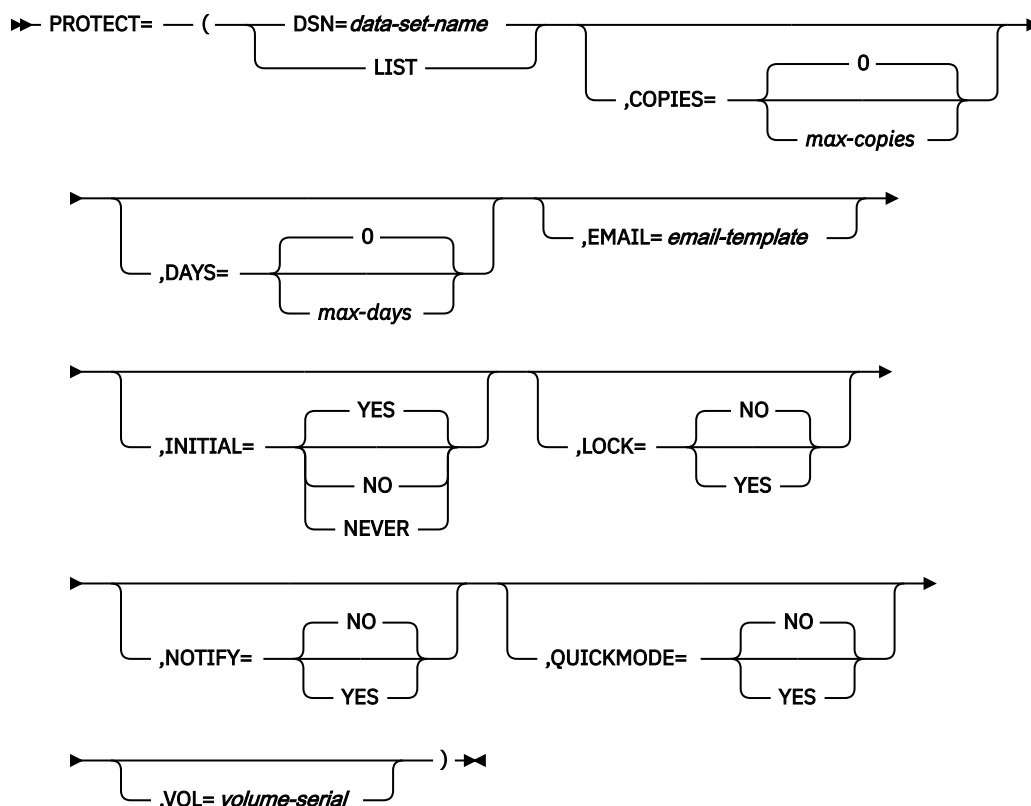
CYGD*TOK sample jobs.

PROTECT command

The PROTECT command is used to protect a resource, or a list of resources.

Syntax

```
PROTECT=(DSN=data-set-name | LIST  
  [, COPIES=0 | max-copies]  
  [, DAYS=0 | max-days]  
  [, EMAIL=email-template]  
  [, INITIAL=YES | NO | NEVER]  
  [, LOCK=YES | NO]  
  [, NOTIFY=YES | NO]  
  [, QUICKMODE=YES | NO]  
  [, VOL=volume-serial])
```



Parameters

DSN=*data-set-name*|LIST

The name of the data set to be protected.

LIST specifies a list of data set names generated by a previous command such as EXPLICIT or PATTERN.

This parameter must be specified.

COPIES=*max-copies*

The maximum number of backup copies that AUTOARC allows to remain in the live backup system.

The default value is 0 (zero) and means the members are immediately eligible to be auto archived unless this is overwritten by a specification of the DAYS parameter. Optional parameter.

DAYS=*max-days*

The maximum number of days that AUTOARC allows backup copies to remain in the live backup system.

The default value is 0 (zero) and means the members are immediately eligible to be auto archived unless this is overwritten by a specification of the COPIES parameter. Optional parameter.

EMAIL=*email-template*

The name of the email text template member to be used for email alerts. This member must reside in the data set named in the EMAILDSN parameter in CYGPARMs. If this member is not found in the EMAILDSN data set, the default member, CYGEMDAT, will be used.

If this parameter is not specified, email alert is not enabled for the specified data set. Optional parameter.

See Chapter 2, “Configuration and implementation,” on page 19 for the optional step to set up the email data set.

INITIAL=YES|NO|NEVER

Controls initial and subsequent backups of data set members.

YES causes an initial backup of all members to be taken when the PROTECT command runs. Automatic backup of members will occur on any subsequent updates, and manual backups can be taken at any time.

NO causes no backups to be taken when the PROTECT command runs. Automatic backup of members will still occur on any subsequent updates, and manual backups can be taken at any time.

NEVER disables backups for the data set. No automatic backup of members will occur and manual backups cannot be performed.

The default is YES. Optional parameter.

LOCK=YES|NO

Determines how a data set will be protected.

When YES, the data set is locked and no member can be modified unless checked out by the Administrator to a specified user or group.

When NO, a data set member can be modified by any user that is authorized to edit the member.

The default is NO. Optional parameter.

NOTIFY=YES|NO

Controls user notification regarding the CSAVE command.

YES causes the user to be notified that they are editing a protected data set and to use the CSAVE command to document changes.

NO disables notification.

The default is NO. Optional parameter.

QUICKMODE=YES|NO

Use with conjunction with INITIAL=NO to avoid performance and space issues related to protecting data sets that have a very large number of members.

When QUICKMODE=YES, the member list of the PDS or PDSE is not stored in the repository, avoiding the system overhead of handling the member list, which becomes significant when it gets very large. QUICKMODE=YES must be preceded by INITIAL=NO so that no initial member backups are generated in the repository. YES is the recommended setting when the number of members exceeds 200,000.

With QUICKMODE=YES, the started task still records change activity and takes a backup each time a member is updated. However, since the member list has not been initially stored in the repository, both ADD and UPDATE actions will be recorded as UPDATE. Also, RENAME operations will not be detected. The Change Report will only show UPDATE and DELETE activities for data sets protected with QUICKMODE=YES.

The default is NO. Optional parameter.

VOL=volume-serial

For uncataloged data sets, the volume serial number of the volume on which the data set resides. Omit this parameter if the data set is cataloged. Optional parameter.

Usage

This command is available for use by the Administrator only.

Within one minute from when the PROTECT command is issued, the protection list is refreshed automatically and the started task starts to monitor the resource.

The COPIES and DAYS parameters are mutually exclusive. They affect the results of the AUTOARC command which archives any member backups that fall outside the backup retention criteria. If the number of backups for a member exceeds the number specified on the COPIES parameter, the oldest ones in excess of that number will be archived from the live system, or if backups for a member exist

which are older than the number of days specified on the DAYS parameter, they will be archived from the live system.

Note: For email support, additional email text files (CYGEM001, CYGEM002, and so on) may be created using the model CYGEMDAT text file.

Example 1: Monitor but do not lock data set, keep backups for 20 days, no initial backup

```
PROTECT=(DSN=SYS1.PROCLIB,  
          LOCK=NO,  
          DAYS=20,  
          INITIAL=NO)
```

Example 2: Protect a data set on primary and alternate residence volumes

Although the data sets on the primary residence volume are cataloged, by specifying the VOL parameter, z/OS Change Tracker defines the entity as a DSN+VOL entity uniquely. This technique prevents future conflicts when the IPL volume is switched to the alternate residence volume.

```
PROTECT=(DSN=SYS1.PARMLIB,  
          VOL=RES001,  
          LOCK=NO)  
  
*  
PROTECT=(DSN=SYS1.PARMLIB,  
          VOL=RES002,  
          LOCK=NO)
```

Example 3: Protect a list of data sets

The EXPLICIT command adds data sets to the list. The SHOW=LIST command shows the populated list. The PROTECT command operates on each data set in the LIST to define them as protected resources.

```
EXPLICIT=SYS1.PARMLIB  
EXPLICIT=SYS1.PROCLIB  
EXPLICIT=SYS1.VTAMLIB  
...  
SHOW=LIST  
PROTECT=(LIST,  
          LOCK=NO,  
          INITIAL=NO,  
          COPIES=30)
```

Example 4: Protect a large PDS

```
PROTECT=(DSN=IBMUSER.HUGE.PDS,  
          INITIAL=NO,  
          QUICKMODE=YES,  
          COPIES=20,  
          DAYS=0)
```

Note: QUICKMODE=YES must be preceded by INITIAL=NO.

Example 5: Upfront notification of editing a member of a protected data set using NOTIFY=YES

```
PROTECT=(DSN=SYS1.PARMLIB,  
          INITIAL=YES,  
          NOTIFY=YES,  
          COPIES=20,  
          DAYS=0)
```

Example 6: Protect a data set requesting to never generate backups

```
PROTECT=(DSN=IBMUSER.PRS.CNTL,  
          INITIAL=NEVER)
```

Example 7: Specific EMAIL to be issued when the library is updated

```
PROTECT=(EMAIL=CYGEM001,  
          LOCK=NO,  
          INITIAL=YES,  
          DSN=SYS1.PARMLIB,  
          COPIES=30,  
          DAYS=0)
```

Related commands

[AUTOARC](#)

[CHECKOUT](#)

[LOGCOMMENT](#)

[REPROTECT](#)

[UNPROTECT](#)

REACTIVATESTC command

The REACTIVATESTC command re-activates the STC after a previous deactivation.

Syntax

```
REACTIVATESTC
```

➡ REACTIVATESTC ➡

Parameters

None.

Usage

This command is available for use by the Administrator only.

This command is used to reactivate the STC after it has been deactivated by the DEACTIVATESTC command.

Example 1: Application of the DEACTIVATE/REACTIVATE during AUTOARC

```
DEACTIVATESTC  
AUTOARC  
REACTIVATESTC
```

Example 2: Application of the DEACTIVATE/REACTIVATE during the SURV function

```
SHOW=PROTECTIONS  
*  
DEACTIVATESTC  
SURV=(LIST,CHG=YES)  
REACTIVATESTC
```

Example 3: Use of DEACTIVATE/REACTIVATE to control the update of a LOCK=YES library

```
//STEP1 EXEC PGM=CYGMAIN,REGION=0M  
//CYGIN DD *  
DEACTIVATESTC  
//*  
//STEP2 EXEC PGM=IEBCOPY,REGION=0M
```

```
//SYSPRINT DD SYSOUT=*
//INPUT DD DISP=SHR,DSN=WSER15.TEST.PDSA
//OUTPUT DD DISP=OLD,DSN=DEV.PRODUCT.INSTALL
//SYSIN DD *
COPY INDD=((INPUT,R)),OUTDD=OUTPUT SELECT MEMBER=((CYGIEB,,R))
//STEP3 EXEC PGM=CYGMAIN,REGION=0M
//CYGIN DD *
//REACTIVATESTC
//*
DEACTIVATESTC
//*
//IEBCOPY1 EXEC PGM=IEBCOPY,REGION=0M
//SYSPRINT DD SYSOUT=*
//INPUT DD DISP=SHR,DSN=WSER15.TEST.PDSA
//OUTPUT DD DISP=OLD,DSN=DEV.PRODUCT.INSTALL
//SYSIN DD *
COPY INDD=((INPUT,R)),OUTDD=OUTPUT
SELECT MEMBER=((CYGIEB,,R))
//STEPLAST EXEC PGM=CYGMAIN,REGION=0M
REACTIVATESTC
//*
```

Related commands

AUTOARC

DEACTIVATESTC

HALTOBJ

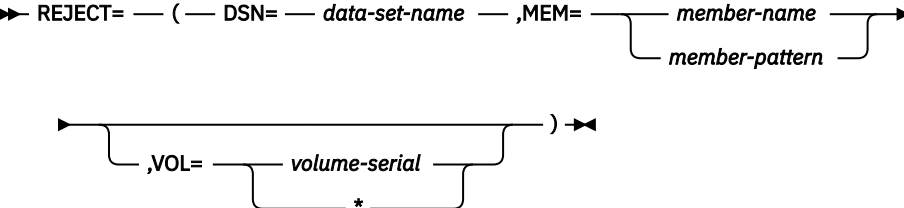
SURVEILLANCE

REJECT command

The REJECT command excludes members from the Tokens MODS report.

Syntax

```
REJECT=(DSN=data-set-name
        ,MEM=member-name|member-pattern
        [,VOL=volume-serial|*])
```



Parameters

DSN=*data-set-name*

The name of the data set which was previously tokenized.

MEM=*member-name|member-pattern*

The name of the member to be excluded from the comparison. MEM= * excludes all members of the specified data set.

VOL=volume-serial

For uncataloged data sets, the volume serial number of the volume on which the data set resides. Omit this parameter if the data set is cataloged.

Usage

It may be desirable to exclude changes for certain non-critical data set members from the MODS report. This can be done using one or more REJECT commands.

Example 1: Exclude data set from MODS report

```
OPTION=CHANGEONLY
REJECT=(DSN=SYS4.CYG.LOADLIB,VOL=*,MEM=CYGQESR)
REJECT=(DSN=SYS4.CYG.LOADLIB,VOL=*,MEM=CYGDEFLT)
...
*
SHOW=TOKENS show tokenized data sets
SHOW=LIST generate a list in memory
TOKEN=(LIST,MODS) re-tokenize the list to report changes
```

Related commands

[TOKEN](#)

REMOTECOMPARE command

The REMOTECOMPARE command compares data sets in two remote environments represented by the unloaded token files.

Syntax

```
REMOTECOMPARE
```

➡ REMOTECOMPARE ➡

Parameters

None.

Usage

The 2 remote environments are represented by 2 token files which must be specified using DD1 and DD2 DD JCL statements. REMOTEOBJECT1 and REMOTEOBJECT2 commands must precede the REMOTECOMPARE command to specify the data set in each environment to be compared.

The data sets to be compared can have different names.

The output of the command is directed to SYSPRINT.

The output is influenced by the OPTION command settings.

This command is similar to REMOTEGROUPCOMPARE but operates on specific data sets.

Example 1: Compare changes between two pairs of uncataloged data sets

Uncataloged data set SYS1.PARMLIB on volume RES001 in hlq.TOKENS.SYSA token file is compared with SYS7.PARMLIB on RES002 in hlq.TOKENS.SYSB token file. Uncataloged data set SYS1.PARMLIB on volume RES001 in hlq.TOKENS.SYSA token file is compared with SYS1.PARMLIB on RES002 in hlq.TOKENS.SYSB token file

```
//DD1 DD DISP=SHR,DSN=hlq.TOKENS.SYSA
//DD2 DD DISP=SHR,DSN=hlq.TOKENS.SYSB
//CYGIN DD *
OPTION=DIFF
*
REMOTEOBJECT1=(DSN=SYS1.PARMLIB,VOL=RES001)
REMOTEOBJECT2=(DSN=SYS7.PARMLIB,VOL=RES002)
```

```
REMOTECOMPARE
*
REMOTEOBJECT1=(DSN=SYS1.PROCLIB,VOL=RES001)
REMOTEOBJECT2=(DSN=SYS1.PROCLIB,VOL=RES002)
REMOTECOMPARE
```

Related commands

[REMOTEOBJECT1, REMOTEOBJECT2](#)

[REMOTEGROUPCOMP](#)

[OPTION](#)

Related information

CYGR* sample jobs.

REMOTEGROUPCOMP command

The REMOTEGROUPCOMP is used to compare two token files each representing a remote environment.

Syntax

```
REMOTEGROUPCOMP
```

➤ REMOTEGROUPCOMP ➤

Parameters

None.

Usage

This command is similar to REMOTECOMPARE but operates on lists of data sets.

The 2 remote environments are represented by 2 token files which must be specified using DD1 and DD2 DD JCL statements. DD1SHOW, DD2SHOW and SHOW=LIST commands must be precede the REMOTEGROUPCOMP command to specify the data sets in the each environment to be compared.

The output of the command is directed to SYSPRINT.

The output is influenced by the OPTIONS command settings.

The STRREP command can be used to map data sets that may have different names but are meant to reflect similar content.

Example 1: Compare two environments represented by two external token files

```
//DD1 DD DISP=SHR,DSN=h1q.TOKENS.SYSA
//DD2 DD DISP=SHR,DSN=h1q.TOKENS.SYSB
//SYSPRINT DD SYSOUT=*
//CYGIN DD *
OPTION=MODS
*
DD1SHOW=(DSN=*,VOL=*)
SHOW=LIST
*
DD2SHOW=(DSN=*,VOL=*)
SHOW=LIST
*
REMOTEGROUPCOMP
```

Related commands

DD1SHOW, DD2SHOW

REMOTECOMPARE

SHOW=LIST

OPTION

REMOTEOBJECT1, REMOTEOBJECT2 commands

The REMOTEOBJECT1 and REMOTEOBJECT2 commands are used to establish data sets in token files to be compared.

Syntax

```
REMOTEOBJECT1=(DSN=data-set-name  
                [,VOL=volume-serial])
```

```
REMOTEOBJECT2=(DSN=data-set-name  
                [,VOL=volume-serial])
```

[illegible][illegible]

Parameters

DSN=*data-set-name*

The name of the data set to be compared.

VOL=volume-serial

For uncataloged data sets, the volume serial number of the volume on which the data set resides. Omit this parameter if the data set is cataloged.

Usage

These commands must precede the REMOTECOMPARE command.

REMOTEOBJECT1 and REMOTEOBJECT2 specify the names of the data sets to be compared. REMOTEOBJECT1 specifies the name of the data set in the environment represented by the token file specified by the DD1 DD JCL statement. REMOTEOBJECT2 specifies the name of the data set in the environment represented by the token file specified by the DD2 DD JCL statement.

The commands specify data sets. Wildcard expansion is not supported.

The data sets specified by REMOTEOBJECT1 and REMOTEOBJECT2 can have different names and can reside on different volumes.

Examples

See “REMOTECOMPARE command” on page 111.

Related commands

REMOTECOMPARE

Related information

CYGR* sample jobs.

REMOVECHECKOUT command

The REMOVECHECKOUT command removes all checkout records from the repository.

Syntax

```
REMOVECHECKOUT
```

➤ REMOVECHECKOUT ➤

Parameters

None.

Usage

This command is available for use by the Administrator only.

This command is functionally equivalent to a check in of all checked out members. This is a quick way of cleaning the repository of previous CHECKOUT actions before restarting with new CHECKOUT commands.

If the command is preceded by an OBJECT command, then only CHECKOUT records for the data set specified in the OBJECT command are removed.

The command should be followed by ENDCHECK.

A TRIGGERREFRESH following ENDCHECK updates the STC with the removal of checked out records.

Example 1: Remove all checkout records from the repository

```
REMOVECHECKOUT  
ENDCHECK  
TRIGGERREFRESH
```

Example 2: Remove all checkout records for a given data set from the repository

```
OBJECT=h1q.DEV.SOURCE  
REMOVECHECKOUT  
ENDCHECK  
TRIGGERREFRESH
```

Related commands

[CHECKOUT](#)

REPROTECT command

The REPROTECT command is used to modify the attributes of a protected resource.

Syntax

```
REPROTECT=(DSN=data-set-name  
            [,COPIES=max-copies]  
            [,DAYS=max-days]  
            [,EMAIL=email-template]  
            [,LOCK=YES|NO])
```

```
[,NOTIFY=YES|NO]
[,VOL=volume-serial])
```

UNPROTECT= — (**DSN=** *data-set-name* **,VOL=** *volume-serial*) ▶▶

Parameters

DSN=*data-set-name*

The name of the data set currently protected.

COPIES=*max-copies*

The maximum number of backup copies that AUTOARC allows to remain in the live backup system. 0 means the members are immediately eligible to be auto archived unless this is overwritten by a specification of the DAYS parameter.

DAYS=*max-days*

The maximum number of days that AUTOARC allows backup copies to remain in the live backup system. 0 means the members are immediately eligible to be auto archived unless this is overwritten by a specification of the COPIES parameter.

EMAIL=*email-template*

Email text template that resides in a data set that is configured in CYGPARMs. Refer to the [“PROTECT command”](#) on page 105 for more details.

LOCK=YES|NO

Determines how a data set will be protected.

Refer to the “PROTECT command” on page 105 for more details.

NOTIFY=YES|NO

Controls user notification regarding the CSAVE command.

Refer to the “PROTECT command” on page 105 for more details.

VOL=volume-serial

For uncataloged data sets, the volume serial number of the volume on which the data set resides. Omit this parameter if the data set is cataloged.

Usage

The REPROTECT command can be used to conduct emergency changes on a locked data set, allowing members of a locked data set to be temporarily unlocked for modification.

Within one minute from when the REPROTECT command is issued, the protection list is refreshed automatically and the new settings take effect.

The resource must already be protected as it cannot be protected using this command. At least one data set attribute must be specified that is different from an existing protection attribute.

Modifying the values of COPIES and DAYS parameters affects future archiving as performed by the AUTOARC command.

Note: For email support, additional email text files (CYGEM001, CYGEM002, and so on) may be created using the model CYGEMDAT text file.

Example 1: Unlocking a data set

The lock status of the SYS1.PARMLIB data set is changed from LOCK=YES to LOCK=NO (allowing any authorized user update access). The SHOW=UNIVERSE command reports all the data sets that are protected by z/OS Change Tracker.

```
REPROTECT=(LOCK=NO,
            DSN=SYS1.PARMLIB,
```

VOL=RES001)
SHOW=UNIVERSE

Example 2: Modifying protection attributes of a protected library

```
LOGCOMMENT='Set up enforced documentation'
*
REPROTECT=(DSN=SYS1.PARMLIB,
            LOCK=YES,
```

Example 3: Define an email text file name for an already protected resource

```
REPROTECT=(DSN=SYS1.PARMLIB,
            EMAIL=CYGEM001)
```

Related commands

AUTOARC

CHECKOUT

PROTECT

RESYNC command

The RESYNC command is used to synchronize information in the Control File after a protected PDS is compressed.

Syntax

```
RESYNC=(DSN=data-set-name
        [,VOL=volume-serial])
```

►► RESYNC= — (— DSN= *data-set-name* — *,VOL= volume-serial*) ►◄

Parameters

DSN=*data-set-name*

The name of a protected data set that is being compressed.

VOL=volume-serial

For uncataloged data sets, the volume serial number of the volume on which the data set resides. Omit this parameter if the data set is cataloged.

Usage

When a protected PDS is compressed, the directory information stored in the z/OS Change Tracker Control File no longer represents the directory of the compressed PDS. If a subsequent batch IEBCOPY job runs, the changes will not be recorded. To resynchronize information stored in the Control File with the current directory information, use the RESYNC command after a COMPRESS step is run.

Example 1: Resync repository directory after a compress

STEP1 uses IEBCOPY to compress a protected PDS. STEP2 uses z/OS Change Tracker to resynchronize the stored directory information in the Control File with the current directory of the PDS after it is compressed.

```
//STEP1 EXEC PGM=IEBCOPY <= compress step
//SYSPRINT DD SYSOUT=*
```

```
//F1 DD DISP=OLD,DSN=IBMUSER.TEST.PDSA
//SYSIN DD *
COPY INDD=F1,OUTDD=F1,LIST=NO
//*
//STEP2 EXEC PGM=CYGMAIN,REGION=0M
//STEPLIB DD DISP=SHR,DSN=YOUR.APF.LINKLIB
//CYGPARM DD DISP=SHR,DSN=YOUR.PARMLIB(CYGPARM)
//SYSPRINT DD SYSOUT=*
//CYGIN DD *
RESYNC=(DSN=IBMUSER.TEST.PDSA)
/*
```

Related commands

[SURVEILLANCE](#)

[TRIGGERREFRESH](#)

SAVEDD command

The SAVEDD command is used to save the tokenization of an zFS path into an MVS token file.

Syntax

```
SAVEDD=dd_name
```

➤ SAVEDD= — *dd_name* ➤

Parameters

SAVEDD=dd_name

The label of the DD JCL statement for the token file. The attributes of the token file are:
PS,F,4096,4096

Usage

The SAVEDD command must follow the SYSUT1 command which performs tokenization.

Related commands

[SYSUT1](#), [SYSUT2](#)

SHOW=ACTIONS command

The SHOW=ACTIONS command lists all the member-level actions for a specific member, or member patterns.

Syntax

```
SHOW=ACTIONS=(DSN=data-set-name | LIST
               ,MEM=member | member pattern
               [,VOL=volume-serial])
```

➤ SHOW=ACTIONS= — (— DSN= data-set-name — ,MEM= — member — member pattern —) — ,VOL= volume-serial — ➤

Parameters

DSN=*data-set-name*

The name of the data set for which actions are shown.

LIST

LIST operates on a list of data sets.

MEM=*member*|*member pattern*

The member name whose actions are to be shown. A wildcard character is allowed.

VOL=*volume-serial*

For uncataloged data sets, the volume serial number of the volume on which the data set resides. Omit this parameter if the data set is cataloged.

SHOW=ACTIONS=LIST

Show actions on a list of data sets.

Usage

This command is available for use by the Auditor only.

The SHOW=ACTIONS=LIST variation of the command displays selected members of the data sets in the list and does not allow for the specification of members. The LIST is built by other commands such as SHOW=PROTECTIONS. The OPTION command controls the types of actions to be shown. One of the following values may be specified: ADD, UPD, DEL, REN or ALL(default is to report all activities).

Example 1: Show update activities related to a given member

```
OPTIONS=UPD
*
SHOW=ACTIONS=(DSN=SYS1.PARMLIB,
               VOL=RES001,
               MEM=IEASSN00)
```

Example 2: Report all activities for all protected libraries

The SHOW=PROTECTIONS command creates a list of all protected libraries. The SHOW=ACTIONS=LIST operates on that list to report important change activity.

```
SHOW=PROTECTIONS
SHOW=ACTIONS=LIST
```

Example 3: Report selected delete activities for protected libraries

```
OPTION=DEL
*
SHOW=PROTECTIONS
SHOW=ACTIONS=LIST
```

Related commands

[SHOW=PROTECTIONS](#)

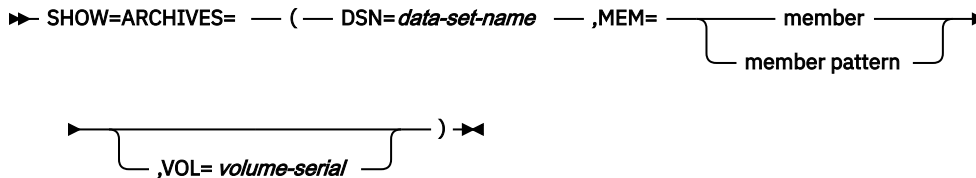
[OPTION](#)

SHOW=ARCHIVES command

The SHOW=ARCHIVES command lists the archived backups for an explicit member or member pattern.

Syntax

```
SHOW=ARCHIVES=(DSN=data-set-name  
                ,MEM=member|member pattern  
                [,VOL=volume-serial])
```



Parameters

DSN=*data-set-name*

The name of the data set containing the member that has been archived.

MEM=*member|member pattern*

The member name whose archives are to be shown. A wildcard character is allowed.

VOL=*volume-serial*

For uncataloged data sets, the volume serial number of the volume on which the data set resides. Omit this parameter if the data set is cataloged.

Usage

The SHOW=ARCHIVES command is available to the Administrator only.

If volume is specified, it must match the volume used when the data set was protected.

Example 1: View a list of archived backups for a given member

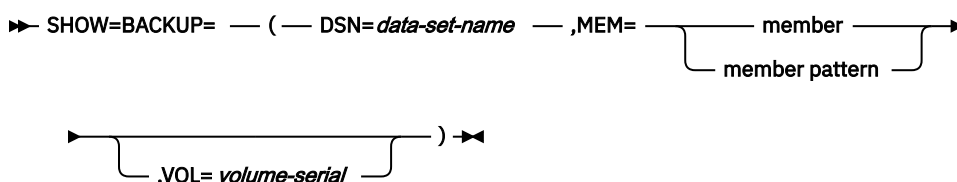
```
SHOW=ARCHIVES=(DSN=SYS1.PARMLIB,  
                VOL=RES001,  
                MEM=IEASSN00)
```

SHOW=BACKUP command

The SHOW=BACKUP command lists all existing backups for a given member or member pattern within the live backup system.

Syntax

```
SHOW=BACKUP=(DSN=data-set-name  
              ,MEM=member|member pattern  
              [,VOL=volume-serial])
```



Parameters

DSN=*data-set-name*

The name of the data set containing the member that has been backed up.

MEM=*member|member pattern*

The member name whose backups are to be shown. A wildcard character is allowed.

VOL=*volume-serial*

For uncataloged data sets, the volume serial number of the volume on which the data set resides. Omit this parameter if the data set is cataloged.

Usage

Users can only view backups of data sets that they are authorized to access.

If volume is specified, it must match the volume used when the data set was protected.

Example 1: View all backups for a given member

```
SHOW=BACKUP=(DSN=SYS1.PARMLIB,  
              VOL=RES001,  
              MEM=IEASSN00)
```

SHOW=CHECKOUT command

The SHOW=CHECKOUT command lists all checked out members in all protected libraries.

Syntax

```
SHOW=CHECKOUT
```

➤ SHOW= — CHECKOUT ➤

Parameters

None.

Usage

This command is available for use by the Administrator only.

Example 1: Report all checked out members in all protected libraries

```
SHOW=CHECKOUT
```

SHOW=LIST command

The SHOW=LIST command displays a list of data sets targeted for processing.

Syntax

```
SHOW=LIST
```

➤ SHOW= — LIST ➤

Parameters

None.

Usage

The list of data sets displayed is built by INCLUDE, EXCLUDE or PATTERN commands.

Example 1: Display a list by EXCLUDE and PATTERN commands

```
EXCLUDE=SYS1.BROADCAST
EXCLUDE=SYS1.ICEDGTM
EXCLUDE=SYS1.DUMP* <-- Pattern exclusion
EXCLUDE=SYS1.HASPACE
EXCLUDE=SYS1.HASPKPT
EXCLUDE=SYS1.LOGREC
EXCLUDE=SYS1.RACF
EXCLUDE=SYS1.VTOCIX.* <-- Pattern exclusion
*
PATTERN=SYS1.*
SHOW=LIST
```

Related commands

[INCLUDE](#)

[EXCLUDE](#)

[PATTERN](#)

SHOW=LOG command

The SHOW=LOG command displays the Administrator log.

Syntax

```
SHOW=LOG
```

➡ SHOW= — LOG ➡

Parameters

None.

Usage

This command is available for use by the Administrator only.

It can be combined with BEGINDATE, ENDDATE, or DATE command to display log entries for a particular date range only.

It can be combined with the OPTION=ERASE command to erase log records.

Example 1: Display Administrator log records for a given date range

Report on all log records from the first day of 2017, up to and including the last day of the year 2022.

```
BEGINDATE=2017/01/01_00:01
ENDDATE=2022/12/31_23:59
SHOW=LOG
```

Example 2: Erase Administrator log records in the given date range from the repository

Report on all log records from the first day of 2017, up to and including the last day of the year 2022. In a subsequent run, by uncommenting the OPTION=ERASE command line all such records will be erased.

```
BEGINDATE=2017/01/01_00:01  
ENDDATE=2022/12/31_23:59  
OPTION=ERASE  
SHOW=LOG
```

SHOW=PROTECTIONS command

The SHOW=PROTECTIONS command lists all protected data sets in the repository.

Syntax

```
SHOW=PROTECTIONS
```

➤ SHOW= — PROTECTIONS ➤

Parameters

None.

Usage

This command is available for use by the Administrator only.

The command output is a list of protected data sets that can be operated on by other commands that accept such a list. This includes the SHOW=ACTIONS command.

Example 1: Display protected data sets

```
SHOW=PROTECTIONS
```

Example 2: Display actions on protected data sets

```
SHOW=PROTECTIONS  
SHOW=ACTIONS=LIST
```

Related commands

[SHOW=ACTIONS](#)

SHOW=TOKENS command

The SHOW=TOKENS command generates a list of all tokenized data sets in the repository.

Syntax

```
SHOW=TOKENS
```

➤ SHOW= — TOKENS ➤

Parameters

None.

Usage

This command is available for use by the Administrator only.

The output of the command is a list of data sets that have been tokenized into the repository and the date this was completed.

The output list of data sets produced by this command can be the input for a tokenize command.

Example 1: Show data sets that have been tokenized into the repository

```
SHOW=TOKENS
```

Example 2: Tokenize all data sets that have been previously tokenized into the repository

The SHOW=TOKENS command creates a list of data sets that currently exist in the repository, then the TOKEN=LIST command proceeds with retokenization of those data sets.

```
SHOW=TOKENS  
TOKEN=LIST
```

SHOW=UNIVERSE command

The SHOW=UNIVERSE command lists all protected resources along with their attributes and protection details.

Syntax

```
SHOW=UNIVERSE
```

►► SHOW= — UNIVERSE ◀◀

Parameters

None.

Usage

This command is available for use by the Administrator only.

The protection attributes of the protected data sets are displayed in the ATTRIBUTES column. The possible values are A (Alert), N (Notify/CSAVE), L (Lock), I (Initial), V (Never), and Q (Quick).

The Cat column indicates whether the resource has been Protected as a cataloged data set or as an uncataloged data set.

Example 1: Display all protected resources

Protected-Resources		Checked-Out-Components		2021/12/09 07:28:45							
CPY	DAY	ATTRIBUTES	Cat	Member	Checker	R	CRNR	Volser	DSN	Admins	EmailTxt
30	0	- - - - -	C	-	-	-	-	B\$US01	USR1.CNTL2		NONE
30	0	- - L - -	C	BLDDEVX	USR12	-	0000	B\$US01	USR1.CNTL3		NONE
30	0	- - - - I -	C	-	-	-	-	B\$US01	USR1.DEV.PRODUCT.CNTL		NONE
30	0	- - L - I -	C	CYGARC	USR1	-	0000	B\$US02	USR1.DEV.PRODUCT.INSTALL		NONE
				CYGESR	USR1	-	0000				
30	0	- - L - I -	C	-	-	-	-	B\$US02	USR1.DEV.PRODUCT.JCL		NONE
30	0	- - - - I -	C	-	-	-	-	B\$US02	USR1.DEV.PRODUCT.LINKLIB		NONE
30	0	- - - - I -	C	-	-	-	-	B\$US02	USR1.DEV.PRODUCT.LINKLIB.DIST		NONE
30	0	- - - - I -	C	-	-	-	-	B\$US02	USR1.DEV.PRODUCT.MAC		NONE
30	0	- - - - I -	C	-	-	-	-	B\$US02	USR1.DEV.PRODUCT.PANELS		NONE
30	0	- N L - I -	C	-	-	-	-	B\$US01	USR1.DEV.PRODUCT.PARMLIB		NONE
30	0	- - - - I -	C	-	-	-	-	B\$US02	USR1.DEV.PRODUCT.SOURCE		NONE

Related commands

[SHOW=PROTECTION](#)

SHOW=USSENV command

The SHOW=USSENV command lists the z/OS UNIX environment nodes defined in the repository.

Syntax

```
SHOW=USSENV
```

➤ SHOW= — USSENV ➤

Parameters

None.

Usage

This command is available for use by the Administrator only.

The environment nodes are created using the USSENV command. The output includes USSENV-RECORD-KEY, Date-Record-Added, Date-Env-Tokenized, and Comment.

Example 1: Show z/OS UNIX nodes

```
SHOW=USSENV
```

Related commands

[USSENV](#)

STRINGREPLACE command

The STRREP command maps data set names when comparing two environments, represented by their tokens.

Syntax

```
STRREP=({old-string, new-string})
```

➤ STRREP= (*old-string* ,*new-string*) ➤

Parameters

old-string

The fragment in the source data set to be replaced.

new-string

A fragment that will replace the old-string in the source data set.

Usage

Use STRREP to apply mapping patterns to data sets to be compared. This mapping is required to pair any data sets that have been renamed or installed under a different prefix.

Preceding this command with `OPTION=DSNREPLACE` prevents mappings of fragment patterns unless they match the entire data set name. This provides for pairing to be done at a specific data set level rather than using global string-matching rules which might result in unintended pairings.

`STRREP` rules applies to data set names only. Use the `VOLREP` command to map volume names.

Example 1: Map a data set that was renamed after it was tokenized

Data sets with prefix.PROTECT1.suffix were tokenized into TOKFILE1. They were renamed to prefix.PROTECTX.suffix and tokenized into TOKFILE2. The `STRREP` command is used in Example 1 to reflect the renaming.

```
OPTION=MODS
STRREP=(PROTECT1,PROTECTX)
*
DD1SHOW
SHOW=LIST
*
DD2SHOW
SHOW=LIST
*
REMOTEGROUPCOMP
//*
```

Example 2: Use of multiple pairs

```
OPTION=MODS
STRREPL=('USER.DATASET1','USER.MOD.DATASET1',
         'USER.DATASET2','USER.MOD.DATASET2')
*
DD1SHOW
SHOW=LIST
*
DD2SHOW
SHOW=LIST
*
REMOTEGROUPCOMP
//*
```

Related commands

[OPTION=DSNREPLACE](#)

[VOLREP](#)

Related information

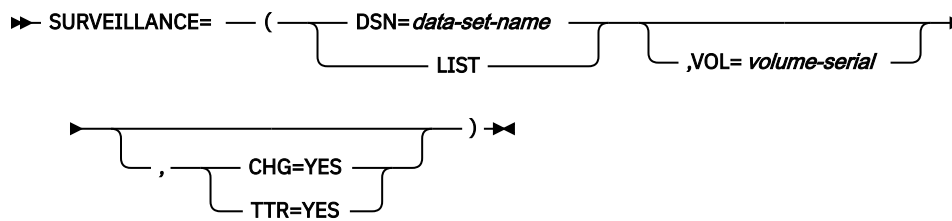
CYG* tokenization sample jobs

SURVEILLANCE command

The `SURVEILLANCE` or `SURV` command examines and backs up changes introduced to protected resources while the STC is inactive.

Syntax

```
SURVEILLANCE=(DSN=data-set-name | LIST
               [,VOL=volume-serial]
               [,CHG=YES | TTR=YES])
```



Parameters

DSN=*data-set-name*

The name of the data set that needs to be examined.

LIST

LIST is a list of data sets to be examined, that is built by another command.

VOL=*volume-serial*

For uncataloged data sets, the volume serial number of the volume on which the data set resides. Omit this parameter if the data set is cataloged.

CHG=YES|TTR=YES

CHG=YES examines the data sets for changes based on their content.

TTR=YES examines the data sets for new or deleted members.

Usage

CHG=YES requires more processing time compared to the TTR=YES mode of surveillance.

The STCs on all LPARs must be stopped or deactivated (using DEACTIVATESTC) before the surveillance command is run.

The SURVEILLANCE command requires READ access to the specified data sets. If READ access is not available, a warning message is issued and surveillance is not run for that data set.

A surveillance run is part of the supplied started task procedure. It identifies and backs up any members in a protected data set that were modified or added since the STC was stopped or deactivated.

Recommendation: Set up your security so that the SURVEILLANCE command and the STC have READ access to all data sets that are protected by z/OS Change Tracker.

Changes and backups discovered by surveillance are attributed to CYGMAIN.

Example 1: Examine changes in protected data sets

In this example, the SHOW=PROTECTIONS command generates a list of currently protected data sets. The SURV command operates against this list to identify and backup any changes introduced while the STC is down.

```

SHOW=PROTECTIONS
SURV=(LIST,TTR=YES)

```

Example 2: Examine any changes in the specified data sets

```

SURV=(DSN=SYS1.PARMLIB,CHG=YES)
SURV=(DSN=SYS1.LINKLIB,CHG=YES)

```


SYSUT1, SYSUT2 command

The SYSUT1 command is used to tokenize an zFS path. When the PATHCOMPARE command is used and two paths are needed, both SYSUT1 and SYSUT2 must be used.

Syntax

```
SYSUT1=PATH='path-name'  
SYSUT2=PATH='path-name'
```

➤ SYSUT1= — PATH= — *path-name* ➤

➤ SYSUT2= — PATH= — *path-name* ➤

Parameters

PATH=*path-name*

Nominates an zFS path to tokenize or compare. Wildcards are accepted.

Usage

This command is usually followed by other commands such as SAVEDD to save the tokens to a token file, PATHCOMPARE to compare this tokenization against another tokenization, or LOADD to compare this tokenization against an existing tokenization file.

This command does not tokenize subdirectories. Use recursive methods of tokenization to tokenize subdirectories.

When performing PATHCOMPARE of two current environments (different zFS paths), SYSUT1 and SYSUT2 are needed.

Example 1: Tokenize files in the home directory

```
//DD1 DD DISP=SHR,DSN=IBMU19.TOKEN.JONES <== PS,F,4096,4096  
SYSUT1=PATH='/u/jones/*.*'  
SAVEDD=DD1
```

Related commands

[LOADDD](#)

[PATHCOMPARE](#)

[SAVEDD](#)

[USSENV](#)

Related information

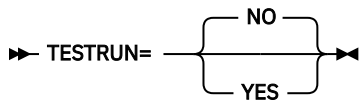
See sample jobs [CYGU2TBF](#) and [CYGU3TAF](#) for examples of recursive tokenization.

TESTRUN command

The TESTRUN command performs a simulation run, where commands are checked but not executed.

Syntax

```
TESTRUN=YES|NO
```



Parameters

TESTRUN=YES|NO

TESTRUN=YES indicates a simulation run.

The default is NO.

Usage

TESTRUN is used in a COPYDIFF operation.

Example 1: Simulation run during a COPYDIFF operation

```

TESTRUN=YES
DELETEMEM=YES
DSN1=SYS1.PARMLIB
VOL1=WORK01
DSN2=SYS2.PARMLIB
VOL2=WORK02
COPYDIFF
  
```

Related commands

[COPYDIFF](#)

TOKEN command

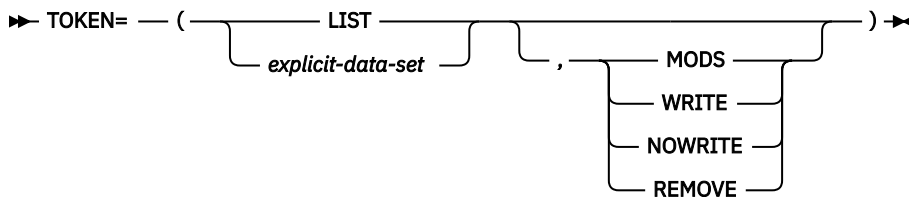
The TOKEN command has three syntax variations:

- Tokenize an explicit data set or a list of data sets.
- Compare against a previous tokenization in the repository.
- Delete tokens from the repository.

Syntax

```

TOKEN=(LIST | explicit-data-set
      [,MODS | WRITE | NOWRITE | REMOVE])
  
```



Parameters

explicit-data-set

This can specify a ddname (DD=*ddname*), a cataloged data set (DSN=*data-set-name*), or an uncataloged data set (DSN=*data-set-name*, VOL=*volume-serial*).

LIST

Specify a list of data set names generated by a previous command such as EXPLICIT or PATTERN.

MODS

Compare with previous tokenization.

NOWRITE

Write tokens to the token file.

REMOVE

Remove tokens from the repository.

WRITE

Write tokens to the repository.

Usage

There are different formats for using the TOKEN=LIST command for processing data sets:

- Only the Administrator can tokenize into the repository. The LIST is generated by a prior command such as SHOW=TOKENS.
- The external tokenization file is nominated using a SYSUT2 DD which must have attributes of: PS, FB, 4096, 4096.
- A CLOSE command must follow a TOKEN=(LIST, NOWRITE) to indicate that tokenization has concluded.
- The report in TOKEN=(LIST,MODS) can be directed to an OUTFILE DD which has attributes of: PS,FB,80,8000. The report content is influenced by previous options such as OPTIONS=MODS or OPTIONS=CHANGEONLY.
- Only the Administrator can remove a previous tokenization from the repository.
- TOKEN=(DSN=, NOWRITE | MODS | REMOVE) operates on a specific data set rather than a prepared list to tokenize a data set, compare a data set, or to remove a specific data set token from the repository.
- Only the NOWRITE option produces a token file that can be later compared using REMOTEGROUPCOMP. All other variations of the TOKEN command to tokenize, compare and delete tokens operate against the repository.
- The repository contains one token per data set member, and different data sets may be tokenized at different times, so a comparison report reflects the change in each data set since it was last tokenized. External token files provide a more flexible way to maintain multiple snap shots of possibly multiple environments.
- The LIST in the TOKEN=LIST command is built by previous PATTERN, INCLUDE, EXCLUDE and SHOW=LIST commands or a SHOW=TOKENS command.
- When the TOKEN command is provided with no options, the default behavior is to tokenize into the repository. This is equivalent to TOKEN=(LIST,WRITE) for lists and TOKEN=(DSN,WRITE) for a particular data set.
- When tokenizing an entire volume, warning errors like message CYG303W can occur for members in any data sets on the volume which are in use at the time. If a message like this occurs, the specified member will not be tokenized correctly. This can be avoided by ensuring that none of the data sets are in use when the tokenization is performed. If it is not possible or practical, the affected data sets can be subsequently tokenized individually.
- Full volume tokenization can be used, for example, by system programmers when they have the task of building a new SYSRES. The SYSRES volume can be tokenized before the new SYSRES is built and then again afterwards to identify member-level changes and new data sets. These changes can then be made to the current live SYSRES in a controlled fashion.
- The TOKEN command with the NOWRITE parameter enables tokenizing directly to an external file. NOWRITE provides a better mechanism for taking a snapshot of an environment at one time, whereas tokens saved in the control file for different data sets could have been created at different times.

Note: Tokenization is not supported for encrypted data sets.

Note: When a group of files is tokenized, by default the tokens are stored in the control file. This is called *internal tokenization*. Later, if the MODS option is requested and a data set in the group no longer exists, the report does not show the deleted data set.

To avoid this problem, use the `TOKEN=(LIST,NOWRITE)` command to store the tokens in a user-specified data set. This is called *external tokenization*. The `REMOPEGROUPCOMP` command can then be used to compare these files and report on deleted or newly allocated data sets.

Example 1: Tokenizing explicit data sets

Generate the tokens for two data sets and store them in the z/OS Change Tracker VSAM Control File.

```
TOKEN=(DSN=SYS1.PARMLIB,VOL=RES001)
TOKEN=(DSN=SYS1.PROCLIB,VOL=RES001)
```

Example 2: Tokenizing and storing the tokens into an external token file

Tokenization of a data set referenced by a `SYSUT1 DD`, and write the tokens to a file referenced by a `SYSUT2 DD`.

```
//SYSUT1 DD DISP=OLD,DSN=SYS1.PROCLIB
//SYSUT2 DD DISP=SHR,DSN=IBMUSER.TOKENS.PROCLIB (PS,FB,4096,4096)
TOKEN=(DD=SYSUT1,NOWRITE)
CLOSE
```

Example 3: Tokenizing a set of data sets whose name matches a pattern

```
PATTERN=SYS1.LINK*
PATTERN=SYS1.PROCLIB
SHOW=LIST
*
TOKEN=LIST
```

Example 4: Tokenizing data sets on two DASD volumes

To tokenize all data sets on a specific DASD volume, the `INCLUDE` command must be used as shown in Example 4. Multiple include commands can be specified for tokenizing multiple DASD volumes.

```
INCLUDE=(DSN=*,VOL=RES001)
INCLUDE=(DSN=*,VOL=RES002)
SHOW=LIST
*
TOKEN=LIST
```

Example 5: Tokenizing a DASD volume and writing tokens to an external file

Tokenize a list of data sets and write the tokens to an external file referenced by a `SYSUT2 DD`. The list is established by multiple `PATTERN` and `INCLUDE` commands.

```
//SYSUT2 DD DISP=SHR,DSN=IBMUSER.TOKENS (PS,FB,4096,4096)
INCLUDE=(DSN=*,VOL=RES001)
EXCLUDE=SYS1.FONTLIB
...
SHOW=LIST
*
TOKEN=(LIST,NOWRITE)
CLOSE
```

Example 6: MODS reporting for explicit data sets

The `MODS` parameter is used to identify all the changes since the tokens were prepared.

```
TOKEN=(DSN=SYS1.PARMLIB,VOL=RES001,MODS)
TOKEN=(DSN=SYS1.PROCLIB,VOL=RES001,MODS)
```

Example 7: MODS reporting for a set of data sets matching a pattern

Example 7 shows how to identify the MODS (any changes) made to a group of pre-tokenized data sets over a period. Tokens reflecting the current content of each data set member are compared against the stored tokens in the control file. These newly generated tokens are used for comparison and are not stored in the control file.

```
OPTION=MODS
*
PATTERN=SYS1.LINK*
PATTERN=SYS1.PROCLIB
SHOW=LIST
TOKEN=(LIST,MODS)
```

Example 8: MODS identification for a set of data sets matching a pattern

Example 8 illustrates how the OPTION=CHANGEONLY command is used to focus on reporting only the changed data sets and the corresponding changed members within those data sets.

```
//OUTFILE DD DISP=SHR,DSN=HLQ.CYG.MODS <-- Required DD
OPTION=CHANGEONLY
*
EXCLUDE=SYS1.BROADCAST
EXCLUDE=SYS1.ICEDGTM
EXCLUDE=SYS1.DUMP* <-- Pattern exclusion EXCLUDE=SYS1.LOGREC
EXCLUDE=SYS1.RACF
EXCLUDE=SYS1.VTOCIX.* <-- Pattern exclusion
*
PATTERN=SYS1.*
SHOW=LIST
*
TOKEN=(LIST,MODS)
```

Note: The HLQ.CYG.MODS data set must have the following attributes: PS,FB,80,8000.

Example 9: Refreshing the tokens in the control file

When there are existing pre-tokenized data sets in the Control File, to refresh those tokens to reflect the current content of data set members, the following commands may be used.

```
SHOW=TOKENS
TOKEN=(LIST)
```

Example 10: Identifying all changes for all previously tokenized data sets

The SHOW=TOKENS command generates a list of tokenized data sets which currently exist in the z/OS Change Tracker Control File. Then, the TOKEN=(LIST,MODS) command works against this generated list to determine the MODS.

```
SHOW=TOKENS
TOKEN=(LIST,MODS)
```

Example 11: REMOVE selected tokens from the repository

To judiciously remove stored tokens for an explicit data set, the following command may be used.

```
TOKEN=(DSN=SYS1.PARMLIB,VOL=RES001,REMOVE)
```

Example 12: Force the removal of a token for data set that no longer exists

```
OPTION=FORCED
TOKEN=(DSN=SYS1.USERLIB.ABC,REMOVE)
```

Example 13: Tokenization multiple DASD volumes, storing tokens externally

Multiple INCLUDE commands build a LIST of data sets on one or multiple DASD volumes.

The NOWRITE parameter indicates tokenization to an external token file.

```
//SYSUT2 DD DISP=SHR,DSN=IBMUSER.TOKENS (PS,FB,4096,4096)
INCLUDE=(DSN=*,VOL=RES001)
INCLUDE=(DSN=*,VOL=RES002)
INCLUDE=(DSN=*,VOL=RES002)
EXCLUDE=SYS1.FONTLIB
...
SHOW=LIST
*
TOKEN=(LIST,NOWRITE)
CLOSE
```

Example 14: MODS identification of changes for an established LIST

```
PATTERN=SYS1.PARMLIB.*
PATTERN=SYS1.PROCLIB.*
EXCLUDE=SYS1.PARMLIB.TEST
TOKEN=(LIST,MODS)
```

Related commands

[INCLUDE](#)

[PATTERN](#)

[EXCLUDE](#)

[SHOW=LIST](#)

[REMOTEGROUPCOMP](#)

[SHOW=TOKENS](#)

[DD1SHOW, DD2SHOW](#)

Related information

CYGC* jobs in the samples library.

TRIGGERREFRESH command

The TRIGGERREFRESH command is used to update the started task running on a given LPAR with changes that were done to the repository.

Syntax

```
TRIGGERREFRESH
```

➤ TRIGGERREFRESH ➤

Parameters

None.

Usage

The changes that need updating could be new protected data sets, new checked-out or checked-in files, or the VOLSER information of a protected data set that has been moved between volumes.

Issuing TRIGGERREFRESH after a series of commands such as ENDCHECK ensures that the started task is up to date with changes in the repository.

TRIGGERREFRESH can also be used to update the started task with the new VOLSER of a protected data set that was reallocated to a different volume while the started task is running. When the refresh is complete, the VOLSER column in the protection list reflects the new VOLSER. This job can be scheduled to run periodically such as every 4 hours.

TRIGGERREFRESH must be the last command issued in the job step.

Example 1: TRIGGERREFRESH

```
TRIGGERREFRESH
```

Related commands

[CHECKIN](#)

[CHECKOUT](#)

[ENDCHECK](#)

[REMOVECHECKOUT](#)

[RESYNC](#)

UNLOADTOKENS command

The UNLOADTOKENS command is used to unload all tokenized data set information stored in the repository into a dynamically allocated flat file.

Syntax

```
UNLOADTOKENS
```

➤ UNLOADTOKENS ➤

Parameters

None.

Usage

This command is available for use by the Administrator only.

The flat file created by this command can be transferred to a remote environment and compared against another unloaded flat file for an environment comparison. The name of the flat file contains the CPU information (CPU number and model number) based on the system where it was created. This information identifies the system where the tokens are coming from as well as the date the token file was created. An example of the dynamically allocated data set follows:

```
root.U0017.M7490.D06293.T1903495
```

The attribute of the flat file are as follows:

```
RECFM=4096, BLKSIZE=4096, DSORG=PS
```

The root node is the TOKENHLQ parameter value specified in the CYGPARMs member. The letter U identifies the file as an unload file and is followed by the CPU number of the machine it was created on. The letter M identifies this node as representing the machine ID and is followed by the machine ID where the file was created.

The date and time nodes follow the machine ID.

Example 1: Unloading tokens

UNLOADTOKENS

Related commands

TOKEN

Related information

CYGR* jobs in the samples library.

UNPROTECT command

The UNPROTECT command removes protection from a resource and delete its backups from the live backup system.

Syntax

```
UNPROTECT=(DSN=data-set-name
            [,VOL=volume-serial])
```

UNPROTECT= — (**— DSN=** *data-set-name* _____) **▶▶**

_____ **.VOL=** *volume-serial* _____

Parameters

DSN=*data-set-name*

The name of the data set of the protected resource.

VOL=volume-serial

For uncataloged data sets, the volume serial number of the volume on which the data set resides. Omit this parameter if the data set is cataloged.

Usage

This command is available for use by the Administrator only.

Within one minute from when the UNPROTECT command is issued, the protection list is refreshed automatically and the resource will no longer be monitored by the started task.

Example 1: Unprotect an uncataloged data set

```
UNPROTECT=(DSN=SYS1.PARMLIB,
            VOL=RES001)
```

Related commands

PROTECT

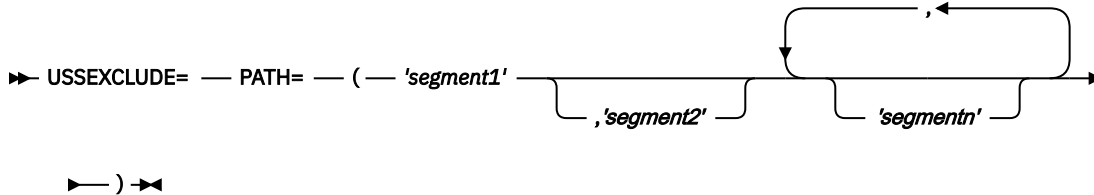
REPROTECT

USSEXCLUDE command

The USSEXCLUDE command excludes unwanted path names from the path compare process.

Syntax

```
USSEXCLUDE=PATH=('segment1',  
                'segment2',  
                .  
                'segmentn')
```



Parameters

PATH='segmentn'

The path name segments.

Usage

Several path names may be excluded by using multiple USSEXCLUDE commands.

Long path names may be excluded by concatenating their string segments on separate lines.

Example 1: Exclude a long z/OS UNIX path name from the compare process

```
USSEXCLUDE=PATH=(' /u/z/OS Change Tracker/*.mytest01.mytest02.',  
                'mytest03.mytest04.mytest05.mytest06.',  
                'mytest07.mytest08.mytest09.mytest10.',  
                'mytest11.abtest12.abtest13.abtest14.',  
                'mytest15.abtest16.abtest17.abtest18.',  
                'abtest19.abtest20.abtest21.abtest22.',  
                'abtest23.abtest24')
```

Example 2: Compare the contents of two z/OS UNIX token files representing an environment at different times

```
//CYGPARM DD DISP=SHR,DSN=YOUR.PARMLIB(CYGPARMS)  
//DD1 DD DISP=SHR,DSN=IBMUSER.CYGUSS.NEWTOKEN  
//DD2 DD DISP=SHR,DSN=IBMUSER.CYGUSS.OLDTOKEN  
//SYSPRINT DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*  
//CYGIN DD *  
OPTION=CHANGEONLY  
LOADDD=DD1 this command sets the sysut1 structure in memory  
LOADDD=DD2 this command sets the sysut2 structure in memory  
*  
USSEXCLUDE=PATH=(' /u/z/OS Change Tracker/*.mytest01.mytest02.',  
                'mytest03.mytest04.mytest05.mytest06.',  
                'mytest07.mytest08.mytest09.mytest10.',  
                'mytest11.abtest12.abtest13.abtest14.',  
                'mytest15.abtest16.abtest17.abtest18.',  
                'abtest19.abtest20.abtest21.abtest22.',  
                'abtest23.abtest24')  
*  
PATHCOMPARE  
//*
```

Related commands

[USSREJECT](#)

USSPATHREPL command

The USSPATHREPL command maps paths used in a compare operation.

Syntax

```
USSPATHREPL=(source-string,target-string)
```

➤ USSPATHREPL= — (— *source-string* — ,*target-string* —) ➤

Parameters

source-string

Prefix of the SYSUT1 path which indicates the file currently being tokenized.

target-string

Indicates the replacement text for the prefix which is intended to result in a name which matches the old name of the path when it was last tokenized.

Usage

When a compare operation begins, all paths in the SYSUT1 environment are examined and the transformation rules are applied to establish matching paths.

Example 1: Map path names for comparison using the USSPATHREPL command

Job 1 for tokenizing the first z/OS UNIX environment:

Tokenize all paths specified in the SYSUT1 commands and saves the tokens into a token file referred to as OLDTOKEN.

```
/* (creates an OLD token file)
  SYSUT1=PATH='/u/home/base/*.txt'
  SYSUT1=PATH='/u/home/base/*.jcl'
  ...
  SAVEDD=DD1
  LIST
```

Job 2 for tokenizing the second z/OS UNIX environment:

Tokenize all paths specified in the SYSUT1 commands and saves tokens into a second token file referred to as NEWTOKEN.

```
/* (creates a NEW token file)
  SYSUT1=PATH='/u/home/target/*.txt'
  SYSUT1=PATH='/u/home/target/*.jcl'
  ...
  SAVEDD=DD1
  LIST
/*
```

Job 3 to compare the two token files:

Referring to the two token files generated in job 1 and job 2, use the USSPATHREPL command to associate the paths and compare the files in the matching paths.

```
//DD1 DD DISP=SHR,DSN=OLDTOKEN <== Must refer to OLD
//DD2 DD DISP=SHR,DSN=NEWTOKEN <== Must refer to NEW
//CYGIN DD *
  USSPATHREPL=(' /u/home/base', ' /u/home/target')
```

```

LOADDD=DD1
LIST
*
LOADDD=DD2
LIST
*
PATHCOMPARE
//*
```

Related commands

[SYSUT2](#)

Related information

CYGU* sample jobs

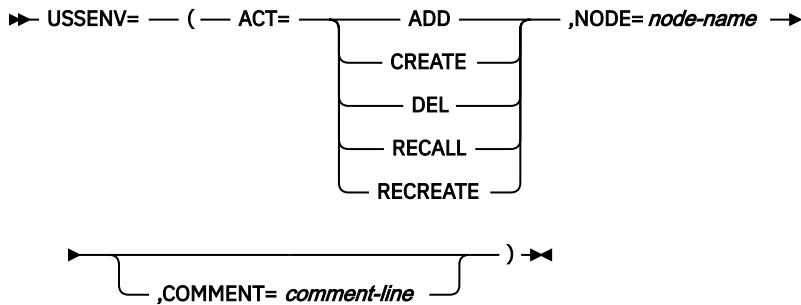
USSENV command

The USSENV command is used to define, add, delete, or recall z/OS UNIX environment nodes that are used to manage token files. These nodes are stored in the repository.

Syntax

```

USSENV=(ACT=ADD | CREATE | DEL | RECALL | RECREATE
        ,NODE=node-name
        [,COMMENT=comment-line])
```



Parameters

ACT=ADD

Define a node. A node must be defined before a token file can be added. As part of defining the node, intermediate and tokenization files are allocated.

ACT=CREATE

Create the ORIG token files related to the node. The processed PATHs are stored in the PATHCMD file.

ACT=DEL

Delete a node that was previously defined using ACT=ADD. Intermediate and tokenization files associated with this node are also deleted.

ACT=RECALL

Recalls the contents of a token file that represents a previous snapshot of the environment. This is usually done to compare against a current snapshot.

ACT=RECREATE

Create the NEW token file for the paths using the established nodes at the time of ACT=CREATE. The processed PATHs are stored in the CMDS file. The tokens are written into the NEW file.

NODE=node-name

Specifies the name of the node that defines a group of token files representing a given z/OS UNIX environment that needs to be monitored for change. The maximum length of the node-name is 8 characters.

COMMENT=comment-line

A brief comment describing the node. The comment must fit one JCL input line. Multi line comments are not supported. The comment is specified when adding a node.

Usage

This command is used by the Administrator who determines the environments of interest and assigns NODE names (an identifier) to each related group of paths as part of a managed approach to z/OS UNIX Tokenization. An example of a few nodes are: RACF, WEB, or CONNECT.

The following files are used to manage z/OS UNIX tokenization:

tokhlq.TOKENS.WEB.ORIG

tokhlq.TOKENS.WEB.NEW

tokhlq.TOKENS.WEB.PATHCMD

tokhlq.TOKENS.WEB.CMDS

Example 1: Define environment nodes

The following example defines 3 environment nodes:

```
USSENV=(NODE=WEB,ACT=ADD,
COMMENT='Websphere Application')
USSENV=(NODE=CICS,ACT=ADD,
COMMENT='CICS related USS Paths')
USSENV=(NODE=RACF,ACT=ADD,
COMMENT='RACF related USS Paths')
```

Example 2: Delete environment node

```
USSENV=(NODE=WEB,ACT=DEL)
```

Example 3: Create base tokenization node

Create a managed token file for a z/OS UNIX environment. Multiple PATHs can be specified. Once all z/OS UNIX files in the specified paths are tokenized, the USSTOKENIZE command writes the paths to the token file. In Example 3, the tokens are stored in a file called: token_hlq.TOKENS.WEB.

```
USSENV=(NODE=WEB,ACT=CREATE) token file is allocated dynamically
*
SYSUT2=PATH='/u/ibmuser/*.*'
SYSUT2=PATH='/u/jones/*.*'
SYSUT2=PATH='/u/smith/*.*'
*
USSTOKENIZE
```

Example 4: Recreate a token file after a change cycle

In STEP1, the PATH statements recorded in the file ".PATHCMD" are read and copied to the intermediate file, token_hlq.TOKENS.WEB.**CMDS**. The CYGIN statement in STEP2, points to this file.

```
//STEP1 EXEC PGM=CYGMAIN ,REGION=0M
//SYSPRINT DD SYSOUT=*
//CYGIN DD *
USSENV=(NODE=WEB,ACT=RECREATE) puts cmds in .CMDS file for next step
//*
//*
//* Execute the commands in the ".CMDS" file as CYGIN
```

```

// * -----
//STEP2      EXEC  PGM=CYGMAIN,REGION=0M
//SYSPRINT    DD   SYSOUT=*
//DD1         DD   DISP=SHR,DSN=CYGUSS.TOKENS.WEB.NEW
//CYGIN       DD   DISP=SHR,DSN=CYGUSS.TOKENS.WEB.CMDS

```

Example 5: Recreate base and current token files to report changes

Assuming a base and current node environment have been created, ACT=RECALL can be run to compare the two token files. The DD1 and DD2 referenced in the commands are dynamically allocated when the parameter ACT=RECALL is specified. They refer to the new and old token files respectively.

```

USSENV=(NODE=WEB,ACT=RECALL)      recalls .ORIG and .NEW token files
*
OPTION=CHANGEONLY
*
LOADDD=DD1           DD1 is allocated by the RECALL function
LOADDD=DD2           DD2 is allocated by the RECALL function
PATHCOMPARE

```

Example 6: Show the existing USSENV nodes in the repository

```
SHOW=USSENV
```

Related commands

[USSTOKENIZE](#)

[SHOW=USSENV](#)

[SYSUT1, SYSUT2](#)

Related information

CYGK* jobs in the samples library.

USSREJECT command

The USSREJECT command is used to exclude one or more unwanted paths from a compare operation.

Syntax

```
USSREJECT=PATH='path-name'
```

➤ USSREJECT= — PATH= — 'path-name' ➤

Parameters

PATH='path-name'

z/OS UNIX path name. Wildcards in suffix path names are allowed.

Usage

One or multiple USSREJECT commands specify USS Paths that are excluded from a compare operation. The rejected paths do not impact on the job return code (RC), regardless of whether they had updates or not.

Example 1: Exclude some z/OS UNIX path from an environment comparison report

```
OPTION=CHANGEONLY
USSENV=(NODE=WEB,ACT=RECALL)
*
SYSUT1=PATH='/u/ibmuser/*.*'
SYSUT1=PATH='/u/jones/*.*'
SYSUT1=PATH='/u/smith/*.*'
*
USSREJECT=PATH='/u/ibmuser/*.o'
USSREJECT=PATH='/u/ibmuser/*.profile'
USSREJECT=PATH='/u/ibmuser/*.trc.16842781'
*
USSREPORT
//*
```

Related commands

[USSEXCLUDE](#)

[USSREPORT](#)

USSREPORT command

The USSREPORT command is used to report changes from a previous tokenization.

Syntax

```
USSREPORT
```

➡ USSREPORT ➡

Parameters

None.

Usage

The SYSUT1 command is used to indicate the z/OS UNIX paths to be compared.

A USSENV command is used to indicate the previous tokenization that is the basis of the compare report.

An OPTION command is used to indicate the type of comparison to be reported.

Example 1: Report modifications in a USS environment

The token file CYGUSS.TOKENS.WEB is read and compared with the current environment tokens generated by SYSUT1 commands. Once two paths are matched by their names, the tokens of the files in those paths are compared, to report the matched file names or the mismatched file names. For the matched file names, the tokens are examined to report SAME or DIFF conditions. If there is a path in one environment but not the other, that path is reported as a mismatched path. The OPTION=CHANGEONLY reports changed data set members only.

```
OPTION=CHANGEONLY
USSENV=(NODE=WEB,ACT=RECALL)
*
SYSUT1=PATH='/u/ibmuser/*.*'
SYSUT1=PATH='/u/jones/*.*'
SYSUT1=PATH='/u/smith/*.*'
*
LOADDD=DD2
USSREPORT
```

Note: In Example 1, the SYSUT1 commands are used to represent the current environment.

Related commands

[OPTION](#)

[SYSUT1](#)

[USSENV](#)

USSTOKENIZE command

The USSTOKENIZE command is used to tokenize one or multiple paths.

Syntax

```
USSTOKENIZE
```

➤ USSTOKENIZE ➤

Parameters

None.

Usage

The SYSUT2 command is used to specify the paths that need to be tokenized.

The SYSUT2 command does not provide for recursive tokenization. However, a shell command can be used to resolve subdirectories recursively and the output of this command can be passed to CYGUDIR to generate the USSTOKENIZE commands. Example 2 shows how this is done.

A USSENV record must be created to indicate the token file destination.

To refresh a token file, the same commands may be used to re-tokenize the environment and write the tokens to the same token file.

Example 1: Tokenize z/OS UNIX paths

Three SYSUT2 commands have been specified followed by the USSTOKENIZE command. When this command is executed the file CYGUSS.TOKENS.WEB is created.

```
//JOB
USSENV=(NODE=WEB,ACT=CREATE) token file allocated dynamically
USSENV=(NODE=WEB,ACT=ADD,COMMENT='WebSphere Application')
*
SYSUT2=PATH='/u/ibmuser/*.*'
SYSUT2=PATH='/u/jones/*.*'
SYSUT2=PATH='/u/smith/*.*'
*
USSTOKENIZE
```

Example 2: Recursive resolve of z/OS UNIX paths

```
//*-----
//* recursive resolve of all directories under multiple paths
//*-----
//*
//STEP1 EXEC PGM=BPXBATCH,REGION=0M
//STDIN DD DUMMY
//STDOUT DD DISP=SHR,DSN=&TOKHLQ..CYGUSAMP.RESOLVED
//STDERR DD SYSOUT=*
//STDPARM DD *,SYMBOLS=JCLONLY
SH
ls -LRD &TOKPATH. | grep -e '/';
/*
//*-----
//* 1) reads the output file created by sh command (resolved paths)
```

```

/* 2) create the z/OS Change Tracker commands to tokenize the resolved
/* list
/*-----
//STEP2   EXEC PGM=CYGUDIR
//STEPLIB DD DISP=SHR,DSN=&PRODHLQ..SCYGLNK
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
/*
//INFILE  DD DISP=SHR,DSN=&RESOLVED Resolved paths
//OUTFILE DD DISP=SHR,DSN=&DIRLIST Tokenize paths
/*
/*-----
/* Tokenize the list of resolved directories
/*
/*-----
//STEP3   EXEC PGM=CYGMAIN,REGION=4096K
//STEPLIB DD DISP=SHR,DSN=&PRODHLQ..SCYGLNK
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSUT2  DD DISP=SHR,DSN=&TOKFILE
//CYGIN   DD DISP=SHR,DSN=&DIRLIST

```

Related commands

[USSENV](#)

[SYSUT2](#)

Related information

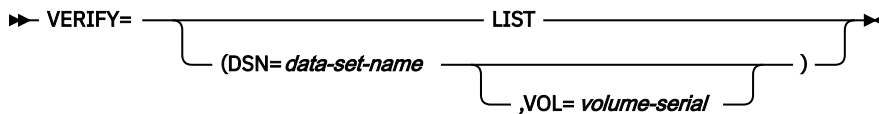
CYGU* jobs in the samples library.

VERIFY command

The VERIFY command verifies the structure of the records in the repository control file.

Syntax

```
VERIFY=LIST|(DSN=data-set-name [,VOL=volume-serial])
```



Parameters

LIST

LIST operates on a list of data sets.

DSN=*data-set-name*

Data set name must match an existing protected data set.

VOL=*volume-serial*

For uncataloged data sets, the volume serial number of the volume on which the data set resides. Omit this parameter if the data set is cataloged.

Usage

This command is available for use by the Administrator only.

This command may be used at any time prior to starting the STC.

VERIFY ensures all required records for a protected resource exist in the repository.

VERIFY=LIST operates on a list that is generated by a previous command, such as SHOW=PROTECTIONS command.

Example 1: Analyze the integrity of all protected data sets

The SHOW command generates a list of all protected resources, and then the VERIFY command analyzes them for integrity.

```
SHOW=PROTECTION  
VERIFY=LIST
```

Example 2: Verify the integrity of a single data set

The VERIFY command verifies the structural integrity of a single data set.

```
VERIFY=(DSN=SYS1.PARMLIB)
```

Related commands

[SHOW=PROTECTIONS](#)

VOL1, VOL2 commands

The VOL1 and VOL2 commands are used to specify volumes for source and target data sets to be operated on by other commands such as COPYDIFF or COMPARE.

Syntax

```
VOL1=source-volume-serial  
VOL2=target-volume-serial
```

➤ VOL1= — *source-volume-serial* — VOL2= — *target-volume-serial* ➤

Parameters

VOL1=*source-volume-serial*

Specify the name of the source volume serial to be operated on by subsequent commands.

VOL2=*target-volume-serial*

Specify the name of the target volume serial to be operated on by subsequent commands.

Usage

The VOL1 and VOL2 commands only need to be used to identify volumes of uncataloged data sets.

The commands must be accompanied by other DSN1 and DSN2 commands that specify the data sets. They must be followed by other commands that need to operate on the source and target data sets.

Related commands

[COMPARE](#)

[COPYDIFF](#)

VOLREP command

The VOLREP command allows data set volume mapping for data sets that are moved following their tokenization.

Syntax

```
VOLREP=(old-volume, new-volume)
```

►► VOLREP= — (— old-volume — ,new-volume —) ►►

Parameters

old-volume

The volume name that needs to be mapped or replaced.

new-volume

The volume name that needs to replace the old-volume.

Usage

Use VOLREP to map source data sets to compare against the same data sets on different volumes, or to reflect the data sets that are reallocated to different volumes following their tokenization.

Use STRREP rules to apply similar mappings to data set names only.

Example 1: Map a data set that has moved to a different volume after it was tokenized

Data sets are tokenized into TOKFILE1 and moved from VOL work01 to VOL work02 and tokenized into TOKFILE2. The VOLREP command is used to reflect this move.

```
OPTION=MODS
VOLREP=(WORK01,WORK02)
*
DD1SHOW
SHOW=LIST
*
DD2SHOW
SHOW=LIST
*
REMOTEGROUPCOMP
/*
```

Related commands

[OPTION=DSNREPLACE](#)

[STRREP](#)

WRITETOFILE command

The WRITETOFILE command is used with the SHOW=ACTIONS command to write the action records to a physical sequential output data set.

Syntax

```
WRITETOFILE
```

►► WRITETOFILE ►►

Parameters

None.

Usage

This command is available for use by the Auditor only.

The OUTFILE DD must be specified. Its attributes are PS,FB,80,8000.

The WRITETOFILE command must be specified before the SHOW=ACTIONS command.

Example 1: Report activities for a particular day

All change activities that were recorded on the previous day (signified by TODAY-1) for the two specified data sets are reported in SYSPRINT and are also written to the external file.

```
DATE=TODAY-1
OPTION=ALL
WRITETOFILE
*
SHOW=ACTIONS=(DSN=SYS1.PARMLIB, MEM=*)
SHOW=ACTIONS=(DSN=SYS1.PROCLIB, MEM=*)
```

Example 2: Show a particular activity for a particular day

All delete activities (signified by command OPTION=DEL) that were recorded within the past two days (signified by TODAY-2) for the specified data set are reported in SYSPRINT and are also written to the external file.

TODAY-2 only considers the delete activities for that day only. That is, any intermediate days since then are not considered. In general, TODAY-nnn (where nnn is some number of days), only considers that specific day.

```
DATE=TODAY-2
OPTION=DEL
WRITETOFILE
*
SHOW=ACTIONS=(DSN=SYS1.PARMLIB, MEM=*)
```

Related commands

DATE

OPTION

SHOW=ACTIONS

Chapter 5. Sample jobs

This chapter presents a summary of jobs in the samples library, SCYGSAMP.

Table 13. Summary of jobs in the samples library	
Job name	Description
CYGA*	Commands for protecting resources and related functions
CYGB*	Commands for tokenizing and reconciling differences
CYGC*	Tokenization and MODS reporting using internal tokens
CYGD*	Verification and synchronization using external tokens
CYGH*	Batch recover and compare of two versions of multiple members
CYGK*	Create and compare USS environment tokens
CYGR*	Remote compare operation using internal tokens
CYGS*	Show the entities in z/OS Change Tracker
CYGT*	Verification and synchronization using external tokens
CYGU*	z/OS UNIX environment change detection
CYGV*	Volume-level tokenization and group compare functions
CYGX*	Example of other helpful commands in z/OS Change Tracker

CYGA* jobs: Commands related to protection of resources

Table 14. CYGA* jobs	
Job name	Description
CYGA1PRO	Protects data sets with LOCK=NO specified. The list of data sets is built using EXPLICIT or PATTERN commands.
CYGA2BKP	Backs up two members of two protected resources along with the user's comments to identify each backup.
CYGA3REC	Recovers a previous version of a specific member, or all versions of that same member.
CYGA4DEL	Backs up a member of a protected resource. It then deletes a backup version of your choice, and finally it reports all available backups currently stored in the data file.
CYGA5REM	Removes two protected resources from the protection list.
CYGA6CKO	Performs an explicit checkout of a member to a designated user. The member remains under exclusive control of the user until it is checked in by the same user
CYGA7CKI	Checks in a member after the work on that member has been completed and releases ownership.
CYGA8REP	Enables the user to set or change the LOCK status of a protected resource by specifying LOCK=YES or LOCK=NO as required. By setting the LOCK status to NO, all members of that library will be made available for update.
CYGA9SRV	Runs the SURVEILLANCE command against all protected resources. In case the STC has been stopped for some period of times, the function identifies, records and backs up any introduced changes while the task has been inactive.

CYGB* jobs: Commands for tokenizing, packaging, and reconciling

The following jobs relate to tokenization. The reconciliation function is presented to synchronize multiple groups of libraries in a pair-wise fashion.

Table 15. CYGB* jobs	
Job name	Description
CYGB1TOK	Tokenizes a group of data sets for future reporting of any MODS introduced during a change cycle.
CYGB2MOD	Identifies any MODS (modifications) introduced to a group of data sets since their time of tokenization.
CYGB5CPY	Reconciles pairs of libraries. This example uses cataloged data sets.
CYGB6CPY	Reconciles pairs of libraries. This example uses uncataloged data sets.

CYGC* jobs: Tokenization and MODS reporting using internal tokens

Use the following jobs to tokenize a list of data sets for MODS reporting over a period of time.

Table 16. CYGC* jobs	
Job name	Description
CYGC1ALC	Allocates a PS file required for the MODS Identification Process when OPTION=CHANGEONLY has been specified. z/OS Change Tracker uses this intermediate file to record the members that have changed since the files were tokenized. This method of MODS identification and packaging uses the reference tokens stored in the z/OS Change Tracker control file. If the tokens are to be stored into an external token file, refer to the examples in CYGD* jobs instead.
CYGC2TOK	Tokenizes a group of data sets (PDS, PDSE, PS, and DA) and stores the tokens in the control file. Either source or load data sets can be tokenized. Use the EXPLICIT, PATTERN, or INCLUDE commands to build a list of data sets for tokenization.
CYGC3MOD	Identifies and reports the MODS in a list of pre-tokenized data sets.

CYGD* jobs: Verify synchrony between production and DR system using external tokens

To verify that the software on your DR system is identical with your production software, perform the following steps:

- Tokenize your software environment on the DR system and then transfer the token file to your production system.
- Tokenize your production software on your production system.
- Using FTP, NDM, or another file transfer utility, make both of these token files available on the same system.
- Run the REMOTEGROUPCOMP command to report the differences between the two environments.

Execute the following jobs in the order listed:

Table 17. CYGD* jobs	
Job name	Description
CYGD1TOK	Tokenizes the software on your DR system and stores the tokens into an external file referred to as the OLDTOKEN. This file must then be transferred to the production site using NDM or FTP.

<i>Table 17. CYGD* jobs (continued)</i>	
Job name	Description
CYGD2TOK	Tokenizes the DR related software at your production site, and stores the tokens into an external file referred to as the NEWTOKEN.
CYGD3CMP	Uses the REMOTEGROUPCOMP command to compare the two token files. This job reports MODS (modifications): member ADD, UPD, and DELETE.

CYGH* jobs: Recover and compare two versions of multiple members

<i>Table 18. CYGH* jobs</i>	
Job name	Description
CYGH1CMP	Recovers two versions of multiple members of a given data set and compares them using SuperC.
CYGH2PRC	Allocates two recovery data sets.
CYGH3PRC	Compares pairs of recovered versions.
CYGH4PRC	Deletes the recovery data sets for repeatability.
CYGH5DAY	Reports change details for selected members of a data set.

CYGK* jobs: Create and compare USS environment tokens

<i>Table 19. CYGK* jobs</i>	
Job name	Description
CYGK1DEL	Deletes any previously created USSENV modes and node data sets.
CYGK2ADD	Defines USSENV nodes in the z/OS Change Tracker Control File.
CYGK3CRE	Creates a token file from a group of USS paths.
CYGK4REC	Retokenizes a USS environment.
CYGK5RCL	Compares the token files created in previous jobs and reports the changes.
CYGK6SHO	Shows the USSENV nodes defined in the z/OS Change Tracker Control File.

CYGR* jobs: Unload internal tokens for use with remote compare

Note: To compare two remote environments, you may prefer to use the sample jobs CYGT* for a direct approach. These sample jobs create the tokens and store the tokens directly into an external token file for comparison by the REMOTEGROUPCOMP command.

<i>Table 20. CYGR* jobs</i>	
Job name	Description
CYGR1UNL	Unloads all of the token records in the control file to a dynamically allocated flat file. This file can then be transferred to other remote LPARs for comparison and reporting of the differences between pairs of data sets.
CYGR2SHO	Shows the content of one or two unload files pointed to by DD1 and DD2. If no parameter is specified for the command, all data sets will be reported. However, a pattern can be specified to display only selected data sets.

<i>Table 20. CYGR* jobs (continued)</i>	
Job name	Description
CYGR3CMP	<p>An unload file may represent an environment at a specified time in the past. Within the same environment, you may compare two unload files (unloaded at different times) and report the changes introduced over time.</p> <p>The Unload files may also represent two remote environments. Comparing these unload files will identify the differences between two remote software environments.</p>
CYGR4TOK	<p>Tokenize a data set referenced by a SYSUT1 DD statement. Users can either write the tokens to the control file or avoid writing them when the NOWRITE parameter is specified.</p> <p>If the SYSUT2 DD is present, the tokens will be written to this physical sequential external file. The token file created as such is internally identical to the one created by the unload tokens command. Therefore, two token files created using this technique can be compared using the REMOTECOMPARE commands to verify the synchrony of the two remote libraries.</p>
CYGR5SHO	Report the contents of two token files each referenced by DD1 and DD2.
CYGR6CMP	Compare two token files to verify whether the two libraries represented by their tokens have identical contents or not.

CYGS* jobs: Using SHOW to report on entities in z/OS Change Tracker

<i>Table 21. CYGS* jobs</i>	
Job name	Description
CYGS1ACT	Reports the member-level update activities recorded by the started task. Activities of the protected resources (source or load PDSE) are reported.
CYGS2BKP	Reports the existing backups and their associated change comments for all members (MEM=*), an explicit member (ABC) or a member pattern (CYG*) for one or multiple protected resources.
CYGS4RES	Reports all protected resources in the system.
CYGS5TOK	Reports all tokenized data sets for which the tokens are stored in the z/OS Change Tracker Control File.
CYGS6UNV	Reports all information known to the started task including the checked-out members.

CYGT* jobs: Verification and synchronization using external tokens

Sample jobs to verify the synchrony between two software environments and to synchronize them when necessary.

- To verify whether two software environments are identical, perform the following steps:
 1. Tokenize your first software environment (on the main LPAR) and store the tokens into an external token file.
 2. Tokenize your second software environment and store the tokens into a second external token file. Then, transfer this second token file to your main system for comparison by the REMOTEGROUPCOMP command.
 3. When both token files coexist on the same system, use the REMOTEGROUPCOMP command to identify and report any differences.
- To synchronize software on both systems, perform the following steps:

1. Tokenize your main software environment and store the tokens into an external token file. Refer to this token file as OLD.
2. After a period of time (such as daily, weekly, monthly) re-tokenize the same software environment (on the main LPAR) and store the tokens into another external token file. Refer to this token file as NEW.
3. Use the REMOTEGROUPCOMP command to report the MODS and then use this information to update the software on the remote system.

The following jobs or manual procedures must be executed in the order listed below.

<i>Table 22. CYGT* jobs</i>	
Job name	Description
CYGT1ALC	Allocates the required temporary PS files.
CYGT2TOK	Tokenizes a group of libraries on your main system.
CYGT3TOK	Tokenizes the same group of libraries after a change cycle.
CYGT4MOD	Uses the REMOTEGROUPCOMP command to compare the two token files. This job reports MODS (modifications): member ADD, UPD, and DELETE.

CYGU* jobs: Change detection in z/OS UNIX files

<i>Table 23. CYGU* jobs</i>	
Job name	Description
CYGU1ALC	Allocates the files required for the subsequent jobs.
CYGU2TBF	Prior to a change cycle, this job tokenizes multiple z/OS UNIX paths recursively using the SH command.
CYGU3TAF	After the changes have been introduced, this job re-tokenizes the same z/OS UNIX paths and saves the tokens to a new token file (NEW).
CYGU4CMP	Compares the two token files NEW and OLD to identify the changed z/OS UNIX files.
CYGU5TOK	Tokenizes one or multiple z/OS UNIX paths and saves the tokens in the dynamically allocated token file for the specified z/OS UNIX environment.
CYGU6REP	Reports the changes introduced in the previously tokenized z/OS UNIX environments by comparing the current tokens with the ones stored in the token file.

CYGV* jobs: Volume-level tokenization and group compare functions

The following jobs are prepared to assist users with volume-level tokenization and the comparison.

<i>Table 24. CYGV* jobs</i>	
Job name	Description
CYGV1GRP	Compares a group of data sets with their equivalent uncataloged ones. In this example we have used the EXPLICIT command to build a list of data sets and then the INCLUDE commands to build a similar list of the uncataloged data sets.
CYGV2DAS	Compares data sets on two DASD volumes. To build a LIST, all PDS and PDSE data sets are resolved from the specified DASD volume.
CYGV3UNC	Compares a group of cataloged data sets with the same uncataloged ones. All PDS and PDSE data sets are resolved either from the catalog or from the specified DASD volume when the INCLUDE command is used.

<i>Table 24. CYGV* jobs (continued)</i>	
Job name	Description
CYGV4EXT	Tokenizes all data sets on a specific DASD volume and saves the tokens to an external file pointed to by a SYSUT2 DD statement. The NOWRITE parameter in the TOKEN command indicates to z/OS Change Tracker that the tokens should not be written to the z/OS Change Tracker Control File. Instead, they are stored into an external token file.
CYGV5EXT	Tokenizes all data sets on a second DASD volume and saves the tokens to another external file pointed to by a SYSUT2 DD statement.
CYGV6REM	Compares the two externally created token files to identify and report the member-level differences.

CYGX* jobs: Examples of other helpful commands in z/OS Change Tracker

<i>Table 25. CYGX* jobs</i>	
Job name	Description
CYGX1CMP	Compare multiple pairs of data sets. In this example the data sets are uncataloged.
CYGX2LOG	Reports the log records.
CYGX3REF	Triggers a protection list refresh. It is used when a protected resource has been re-allocated on a different DASD volume while the z/OS Change Tracker started task was active. Once a protected resource is moved to a volume other than the one known to the started task, either the task must be recycled or this job must be run.

Chapter 6. z/OS Change Tracker messages

This section documents the z/OS Change Tracker messages.

Message types are either informational, warning, or error. Message identifiers include a prefix, a number, and either I, W, or E to indicate the severity:

I

Informational message. Describes information or the status for normal conditions and operations. These messages can also provide supporting information accompanying warning or error messages.

W

Warning message. Alerts to a condition that might cause problems in the future. When a warning message is displayed, processing usually continues, but might not complete in a way that is expected. Examine for possible future errors and possible action.

E

Error message. Alerts to a problem that has occurred. Processing cannot continue. Error messages might be accompanied by additional supporting information for diagnosis, problem determination, and action.

CYG100I	Option STCPRINTOFF is in effect. STC does not generate detailed log information.
----------------	---

Explanation:

The STCPRINTOFF option in the CYGPARMs configuration file has been set. The z/OS Change Tracker started task will not display detailed log information.

System action:

The system continues.

User response:

None. Informational message.

CYG101I	CYGPARM DSN: <i>dsn</i> Current User : <i>user</i> LINKLIB (APF) : <i>lib</i> Enable SAF Exit : YES NO Print Utility : <i>name</i> SMS STORCLAS : <i>class</i> USS Token File HLQ : <i>hlq</i> USS target dir : <i>dir</i> Options Settings : <i>options</i> Panel National Lang : ENU JPN Email file data set : <i>dsname</i> Protection List : <i>string</i>
----------------	---

Explanation:

This message displays the currently active CYGPARMs parameters.

System action:

The system continues.

User response:

None. Informational message.

CYG102I	Control File: <i>dsn</i> Data File : <i>dsn</i> Archive Control File: <i>dsn</i> Archive Data File : <i>dsn</i> Files Access Level : <i>string</i>
----------------	---

Explanation:

This message displays the repository files currently in use.

System action:

The system continues.

User response:

None. Informational message.

CYG103E **REQUIRED parameter is missing: *parm***

Explanation:

The required parameter has not been specified in the CYGPARM member.

System action:

The job or task terminates.

User response:

Contact the installer to check the validity of the specified parameter in the CYGPARM member.

CYG104E **CYGPARM DD not specified and CYGPARM member not found in the PARMLIB concatenation.**

Explanation:

The job needs to access the CYGPARM configuration member but a CYGPARM DD statement has not been specified and a CYGPARM member has not been found in the PARMLIB concatenation.

System action:

The job step terminates.

User response:

Provide the DD or contact the SYSPROG.

CYG105E **Member must be specified in //CYGPARM DD.**

Explanation:

A member name has not been specified in the CYGPARM DD statement.

System action:

The job step terminates.

User response:

Correct the DD statement and resubmit the job.

CYG106E **//CYGPARM DD open failure *parm***

Explanation:

The member specified in the CYGPARM DD statement does not exist.

System action:

The job step terminates.

User response:

Specify a valid member name in the DD statement.

CYG107E **CYGPARM DSN DSORG must be FB: *dsname***

Explanation:

The member specified in the CYGPARM DD statement does not exist.

System action:

The job step terminates.

User response:

Place the CYGPARM member in an 80 byte FB data set.

CYG108E **CYGPARM DSN must be a cataloged data set.**

Explanation:

The data set specified on the CYGPARM DD statement must be a cataloged data set.

System action:

The job step terminates.

User response:

Specify the name of a cataloged data set which contains the CYGPARMs member.

CYG109E**EMAILDSN must be a cataloged data set.****Explanation:**

The data set specified on the EMAILDSN parameter in the CYGPARMs member must be a cataloged data set.

System action:

The job step terminates.

User response:

Specify the name of a cataloged data set on the EMAILDSN parameter in the CYGPARMs member.

CYG110E**IBM z/OS Change Tracker is not enabled.****Product Registration rc=code****Explanation:**

IBM z/OS Change Tracker is not enabled to run on this system.

System action:

The job step terminates.

User response:

Contact the installer to determine why the product is not enabled.

CYG111I**Trial in progress. *num* days remaining.****Explanation:**

z/OS Change Tracker is running using the 90-day trial license feature. *num* indicates the number of trial days remaining.

System action:

The system continues.

User response:

None. Informational message.

CYG112E**LRECL of CYGPARM data set must be 80: *parmlib* data set****Explanation:**

The data set containing the CYGPARMs member must have RECFM=FB and LRECL=80.

System action:

The job step terminates.

User response:

Place the CYGPARMs member in an 80 byte FB data set.

CYG113E**Invalid parameter: *parm*****Explanation:**

The CYGPARMs member contains an invalid parameter.

System action:

The job step terminates

User response:

Correct or remove the invalid parameter from the CYGPARMs member.

CYG114W**Parameter not unique.****Explanation:**

A parameter in the CYGPARMs member has been specified more than once.

System action:

The duplicate statement is ignored.

User response:

Check the parameters in the CYGPARMs member for duplicates.

CYG115E**DD CYGIN required.****Explanation:**

The job does not have the required CYGIN DD statement.

System action:

The job step terminates.

User response:

Add a CYGIN DD statement to the job step.

CYG116I**Opened successfully: *dsname*****Explanation:**

This repository control file or archive control file has been opened successfully.

System action:

The system continues.

User response:

None. Informational message.

CYG117I**90-day trial period expired.****Explanation:**

z/OS Change Tracker is running using the 90-day trial license feature and the 90-day trial period has expired.

System action:

z/OS Change Tracker is not enabled for execution. Message [CYG110E](#) follows.

User response:

None. Informational message.

CYG120E**Parameter parsing failed.**

Routine= *code* Field in error: *parm*

Explanation:

The parsing of a CYGPARMs parameter by the CYGSETPA routine has failed. This is an internal error.

System action:

The job step terminates.

User response:

Contact IBM Support.

CYG130E**Required DD not supplied. DD: *ddname*****Explanation:**

The REMOTECOMPARE command requires two DD statements, DD1 and DD2, specifying the 2 files containing the tokens for the data sets specified by the corresponding REMOTEOBJECT1 and REMOTEOBJECT2 commands.

System action:

The job step terminates.

User response:

Change the job to add the required DD1 or DD2 DD statement.

CYG131E**Commands REMOTEOBJECT1 and REMOTEOBJECT2 are both required.****Explanation:**

The REMOTECOMPARE command requires the prior execution of the REMOTEOBJECT1 and REMOTEOBJECT2 commands, which specify the names of the data sets to be compared.

System action:

The job step terminates.

User response:

CYG132E	Tokens not found in <i>ddname</i> VOL= <i>volser</i> DSN= <i>dsname</i>
---------	--

Explanation:

An invalid CYGMAIN parameter has been specified. Valid parameters are HALT and UNHALT.

System action:

The job step terminates.

User response:

Correct the parameter in error.

CYG204W

Path exceeds 504 characters; not processed.

Explanation:

For the PATHCOMPARE or USSREPORT command a specified path name exceeds the maximum length allowed.

System action:

The specified path is ignored.

User response:

Check that the paths are correctly specified.

CYG205E

**Command is in error. The correct format is:
SYSUT1=PATH=pathname**

Explanation:

The pathname must be enclosed in quotation marks.

System action:

The job step terminates.

User response:

Enclose the pathname in quotation marks.

CYG205I

Paths are to be inserted in a LIST after exclusions:

Explanation:

The following path names will be excluded from the list of paths to be compared.

System action:

The system continues.

User response:

None. Informational message.

CYG205W

The following duplicate path was NOT placed in the LIST:

Explanation:

The following duplicate path names will be excluded from the list of paths to be compared.

System action:

The specified path is ignored.

User response:

Check that paths are correctly specified.

CYG206W

The specified RACF group was not found.

Explanation:

The RACF group name specified on the CHECKOUT command is not defined to RACF.

System action:

The CHECKOUT command is not executed.

User response:

Specify a valid RACF group name on the CHECKOUT command.

CYG208E

**Command is in error. For COMPARE the format is:
OPTIONS=SAME DIFF MATCH MISMATCH MODS ALL
For ACTIONS reporting the format is: OPTIONS=ADD DEL UPD REN ALL
For LPAR reporting the format is: OPTIONS=SUMMARY**

Explanation:

An error occurred in processing the command.

System action:

The job step terminates.

User response:

Correct the command as per the message text.

CYG208W	No data set was tokenized. Check the data set names.
----------------	---

Explanation:

None of the specified data sets were tokenized.

System action:

The job step terminates.

User response:

Check that the specified data set names are valid.

CYG209I	Processed: VOL=vol DSN=dsn
----------------	-----------------------------------

Explanation:

The indicated data set has been tokenized.

System action:

The system continues.

User response:

None. Informational message.

CYG209W	READ access required for list of backups/recovery. DSN of recovery: dsn VOLSER: vol
----------------	--

Explanation:

The list of backups for the data set could not be displayed because the user does not have SAF READ access to the data set.

System action:

The SHOW=BACKUPS command is not executed.

User response:

Obtain SAF READ access to the data set.

CYG210E	The CYGPARM DD cannot be specified for the CHECKOUT, CHECKIN or REMOVECHECKOUT commands.
----------------	---

Explanation:

A CYGPARM DD statement cannot be used in a job step with the CHECKOUT, CHECKIN, or REMOVECHECKOUT command.

System action:

The job step terminates.

User response:

Remove the CYGPARM DD statement from the job step.

CYG210W	READ access required for surveillance. DSN: dsn VOL: vol
----------------	---

Explanation:

The SURVEILLANCE command could not be run for the specified data set because the user does not have SAF READ access to it.

System action:

Surveillance for the data set is not performed.

User response:

Obtain SAF READ access to the data set or select another data set that has SAF READ access.

CYG211E

Command is in error. The correct format is:
SAVEDD=ddname; DDNAME contains: ddname

Explanation:

The format of the SAVEDD command is incorrect. The correct format is SAVEDD=ddname.

System action:

The job step terminates.

User response:

Correct the format of the SAVEDD command.

CYG211W

READ access is required for member backup.
DSN: dsnname
VOLSER: vol

Explanation:

The backup of the member could not be performed because the user does not have SAF READ access to the data set.

System action:

The MEMBERBACKUP command is not executed.

User response:

Obtain SAF READ access to the data set.

CYG212E

Command is in error. The correct format is:
LOADDD=ddname

Explanation:

The format of the LOADDD command is incorrect. The correct format is LOADDD=ddname

System action:

The job step terminates.

User response:

Correct the format of the LOADDD command.

CYG212W

Member does not exist in the library.
MEM: mem

Explanation:

The backup of the member could not be performed because the member does not exist in the data set.

System action:

The MEMBERBACKUP command is not executed.

User response:

Specify the name of a member in the data set.

CYG213E

Unexpected error. Contact support.

Explanation:

Command parsing has failed. This is an internal error.

System action:

The job step terminates.

User response:

Contact IBM support.

CYG213I

Processed: VOL=vol DSN=dsn

Explanation:

The indicated data set has been processed by the COPYDIFF or GROUPNAME command.

System action:

The system continues.

User response:

None. Informational message.

CYG214E

**Command is in error. The correct format is:
CHECKDIR=Y|N**

Explanation:

The format of the CHECKDIR command is incorrect. The correct format is CHECKDIR=Y|N

System action:

The job step terminates.

User response:

Correct the format of the CHECKDIR command.

CYG215E

**Command is in error. The correct format is:
PATHCOMPARE**

Explanation:

The format of the PATHCOMPARE command is incorrect. The correct format is PATHCOMPARE.

System action:

The job step terminates.

User response:

Correct the format of the PATHCOMPARE command.

CYG218E

**The USSTOKENIZE command must be preceded by:
SYSUT2=PATH=" " commands**

Explanation:

The USSTOKENIZE command is not preceded by a SYSUT2=PATH command.

System action:

The job step terminates.

User response:

Add a SYSUT2=PATH command to the job step.

CYG223E

**Command is in error. The correct format is:
VOLSER=vvvvvv**

Explanation:

The format of the VOLSER command is incorrect. The correct format is VOLSER=vvvvvv.

System action:

The job step terminates.

User response:

Correct the format of the VOLSER command.

CYG224E

The supplied VOLSER is not online.

Explanation:

The indicated volume is not online or the VOLSER is not found.

System action:

The job step terminates.

User response:

If the VOLSER is incorrect, enter the correct VOLSER and try again. If the correct VOLSER was specified, find out when the volume will be brought back online.

CYG225E

**Command is in error. The correct format is:
TESTRUN=YES|NO**

Explanation:

The format of the TESTRUN command is incorrect. The correct format is TESTRUN=YES | NO

System action:

The job step terminates.

User response:

Correct the format of the TESTRUN command.

CYG226E

**Command is in error. The correct format is:
DEVICE=3380 or 3390**

Explanation:

The format of the DEVICE command is incorrect. The correct format is DEVICE=3380 | 3390.

System action:

The job step terminates.

User response:

Correct the format of the DEVICE command.

CYG227E

**Command is in error. The correct format is:
MVSCMD=*command* or MVSCMD=(*part-1*, *part-2*)**

Explanation:

The format of the MVSCMD command is incorrect. The correct format is MVSCMD='*command*' or MVSCMD='*part-1*', '*part-2*'.

System action:

The job step terminates.

User response:

Correct the format of the MVSCMD command.

CYG227I

Waiting for the ENDCHECK command to complete.

Explanation:

The ENDCHECK command has a 30 second delay to allow for the started task to refresh.

System action:

The system continues.

User response:

None. Informational message.

CYG228I

Waiting approximately 30 seconds for the STC protection list to be refreshed.

Explanation:

The REPROTECT and TRIGGERREFRESH commands have a 30 second delay to allow for the started task to refresh.

System action:

The system continues.

User response:

None. Informational message.

CYG230E

BASE and TARGET dsn+vol are the same.

Explanation:

The base and target data sets are the same. COPYDIFF requires that the data set names be different.

System action:

The job step terminates.

User response:

Ensure that the base and target data set names are different.

CYG230I **Above path excluded by request.****Explanation:**

The above path name has been excluded from the list of paths to be compared.

System action:

The system continues.

User response:

None. Informational message.

CYG231E **Recovery PDS must be empty. Job is aborted.**
DSN: *dsname***Explanation:**

The recovery data set specified for the MEMBERRECOVER command contains members.

System action:

The job step terminates.

User response:

Ensure that the specified recovery data set contains no members.

CYG231I **Rejected/Included Path: *path*****Explanation:**

The indicated path name has been rejected from the list of paths to be compared or saved.

System action:

The system continues.

User response:

None. Informational message.

CYG232E **The OUTFILE DD must be specified.****Explanation:**

OPTION CHANGEONLY, IDENTIFY, or USSIDENTIFY has been specified but no OUTFILE DD statement has been supplied.

System action:

The job step terminates.

User response:

Add an OUTFILE DD statement to the job step. The OUTFILE data set must have the following attributes: PS,FB,80,8000.

CYG237E **Command is in error. The correct format is:**
LOGCOMMENT=*string* or COMMENT=*string***Explanation:**

The format of the LOGCOMMENT or COMMENT command is incorrect. The correct format is LOGCOMMENT=*'string'* or COMMENT=*'string'*.

System action:

The job step terminates.

User response:

Correct the format of the LOGCOMMENT or COMMENT command.

CYG238E **The OUTFILE DD must be specified.****Explanation:**

The WRITETOFILE command has been specified but no OUTFILE DD statement has been supplied.

System action:

The job step terminates.

User response:

Add an OUTFILE DD statement to the job step. The OUTFILE data set must have the following attributes:
PS,FB,80,8000.

CYG238I

***ccount* DSNs checked for activities**
***ccount* DSNs determined to have no activities**
***ccount* DSNs determined to have activities**
***ccount* DSNs failed the VIEW function**

Explanation:

Lists the data set counts for the SHOW=ACTIONS=LIST command.

System action:

The system continues.

User response:

None. Informational message.

CYG239W

Object is not in the STC memory or is incomplete.
DSN= *dsname*
VOL= *volser* FUNC=*code* CODE=*code*

Explanation:

A required record type for this PDS is missing from the z/OS Change Tracker Control File.

System action:

The command is not executed.

User response

Run the UNPROTECT command on the data set and then run the PROTECT command again. Alternatively, run the following commands and send the report to IBM Support.

```
OPTION=DIAGNOSE  
OBJECT=(DSN=protected-pds,VOL=volser)
```

Note that the VOL parameter is only required for an uncataloged protected resource.

CYG240W

Data set not found or insufficient access.
DSN=*dsname*

Explanation:

The indicated data set was not found in the catalog or the user has insufficient access to it.

System action:

The command is not executed.

User response:

Specify a VOLSER and retry or check with the security administrator.

CYG241I

Data set to act on:
VOL=*vol* DSN=*dsn*

Explanation:

The data set being acted on by the OBJECT command.

System action:

The system continues.

User response:

None. Informational message.

CYG242W

***volser* is not online.**

Explanation:

The indicated volume is not online.

System action:

The command is not executed.

User response:

Check that VOLSER is specified correctly.

CYG243I

TOKEN records are successfully removed.
VOL: *vol* DSN: *dsn*

Explanation:

The token records for the indicated data set have been removed.

System action:

The system continues.

User response:

None. Informational message.

CYG243W

REMOVAL of token records failed.
VOL: *volser* DSN: *dsname*

Explanation:

Either no token records exist for the specified data set or an error has occurred.

System action:

The tokens for the specified data set are not removed.

User response:

Check that the data set name and volume have been specified correctly. The SHOW=TOKENS commands can be used to produce a list of tokenized data sets in the repository.

CYG244I

Resource is protected successfully: *dsn*
VOL=*vol*

Explanation:

The indicated data set has been added to the protection list.

System action:

The system continues.

User response:

None. Informational message.

CYG244W

Resource is not found in the Control File:
VOL=*vol* DSN=*dsname*

Explanation:

The specified data set is not a protected resource.

System action:

The command is not executed.

User response:

The specified data set is not a protected data set. Run the command SHOW=PROTECTIONS for a list of all existing protected data sets.

CYG245W

Resource is already protected: *dsname*
VOL=*volser*

Explanation:

The specified data set is already protected.

System action:

The command is not executed.

User response:

No further action is required.

CYG246I

functype* skipped: VOL=*vol* DSN=*dsn

Explanation:

The verification or surveillance function was not performed for the specified data set because the specified volume is not online.

System action:

The system continues.

User response:

None. Informational message.

CYG246W	Resource protection failed. RC=code DSN= dsname VOL= volser
----------------	--

Explanation:

The PROTECT command failed for the indicated data set.

System action:

The PROTECT command is not executed.

User response:

If you cannot determine the cause of the error, contact IBM Support.

CYG247I	During LOAD, command USSINCLUDE is in effect.
----------------	--

User response:

None. Informational message.

CYG247W	View failed. RC: code Resource: dsn VOL: volser
----------------	--

Explanation:

The internal View function failed for the indicated data set.

System action:

The SHOW=ACTIONS=LIST command is not executed.

User response:

This is an internal error. Contact IBM for support.

CYG248W	TOKENIZE function failed. VOL: volser DSN: dsname
----------------	--

Explanation:

The TOKENIZE function failed for the indicated data set.

System action:

The TOKENIZE command is not executed.

User response:

Check the DSN and VOL parameters to make sure they are correctly specified.

CYG249E	Recovery failed with RC=code
----------------	-------------------------------------

Explanation:

■ An error occurred when attempting to recover a backup or archived backup.

System action:

■ The recovery is not performed.

User response:

Contact IBM Support.

CYG249W	Data set previously deleted. UNPROTECT continues. DSN= dsname
----------------	--

Explanation:

The indicated data set does not exist.

System action:

The UNPROTECT command continues.

User response:

No action required.

CYG250W **No checked-out members were found.**

Explanation:

For SHOW=CHECKOUT, no checked out members were found.

System action:

The SHOW=CHECKOUT command ends.

User response:

No action required.

CYG251I **No backups were found.**
DSN: *dsn*
VOL: *volser*
MEM: *member*

Explanation:

For SHOW=BACKUP, no backups were found for the indicated member.

System action:

The SHOW=BACKUP command ends.

User response:

No action required.

CYG252W **Backup not found.**
DSN: *dsname*
VOL: *volser*
Member: *mem*
Ver: *nr*

Explanation:

The indicated backup version was not found in the repository.

System action:

The recovery command is not executed.

User response:

Check that the specified parameters are correct.

CYG253I **Number of empty or orphaned backups: *nr***

Explanation:

The indicated number of empty or orphaned backups has been found

System action:

The system continues.

User response:

None. Informational message.

CYG254E **Invalid change event on the Data File: *address***

Explanation:

An invalid record has been found in the repository.

System action:

The command is not executed.

User response:

Contact IBM Support.

CYG254I **Number of backups where TOKENS is specified: *nr***

Explanation:

The indicated number of backups has been found.

System action:

The system continues.

User response:

None. Informational message.

CYG254W

Protection List is empty.

Explanation:

The SHOW=UNIVERSE command found no data sets on the started task protection list.

System action:

The system continues.

User response:

No action required.

CYG255I

Delete of backup complete. RC=code

Explanation:

The indicated backup has been deleted.

System action:

The system continues.

User response:

None. Informational message.

CYG255W

No Halted object exists.

Explanation:

The SHOW=HALT command found no halted objects.

System action:

The system continues.

User response:

No action required.

CYG256W

Backup not found.

DSN: *dataset*

VOL: *volser*

Member: *member*

Timestamp: *timestamp*

Explanation:

The indicated backup version was not found in the repository.

System action:

The recovery command is not executed.

User response:

No action required.

CYG257E

**Command is in error. The correct format is:
LIST**

Explanation:

The format of the LIST command is incorrect. The correct format is LIST.

System action:

The job step terminates.

User response:

Correct the format of the LIST command.

CYG257W	OBTAIN for this object has failed: DSN: <i>dsn</i> VOL: <i>vol</i>
Explanation: An internal error occurred when attempting to access the indicated data set.	
System action: The VERIFY command is not executed.	
User response: This is an internal error. Contact IBM support.	
CYG258I	Halted object exists. Use UNHALTOBJECT command to remove it from the Control File. Object: <i>dsn</i>
Explanation: The indicated data set is halted.	
System action: The system continues.	
User response: Use the UNHALTOBJECT command to release the halt.	
CYG258W	Backup comment not found. DSN: <i>dsn</i> VOL: <i>vol</i> Member: <i>mem</i> Date: <i>date</i>
Explanation: The indicated backup comment was not found in the repository.	
System action: The MEMBERCOMMENT command is not executed.	
User response: Check that the specified parameters are correct.	
CYG259W	Backup record not found. DSN: <i>dsname</i> VOL: <i>volser</i> Member: <i>member</i> Date: <i>timestamp</i>
Explanation: The indicated backup was not found in the repository.	
System action: The MEMBERCOMMENT command is not executed.	
User response Check that the specified parameters are correct. Run the following command to list all member backups for <i>dsname</i> :	
<pre>SHOW=BACKUPS=(DSN=<i>dsname</i>,MEM=*)</pre>	
CYG260W	Record type is not found: TYPE=<i>rtype</i>
Explanation: The indicated record type was not found in the repository.	
System action:	

The OBJECT command is not executed.

User response:

This is an internal error. Contact IBM support.

CYG261E

Installation verification failed: check your Install.

Explanation:

The installation verification program has failed.

System action:

The job step terminates.

User response:

Check the job output for errors to determine the cause of failure.

CYG262I

Parameters of the protected resource are modified:

DSN: *dsn*

VOL: *vol*

Explanation:

The protection parameters for the indicated data set have been modified.

System action:

The system continues.

User response:

None. Informational message.

CYG262W

Resource data has not changed. REFRESH not required.

DSN: *dsn*

VOL: *vol*

Explanation:

The protection parameters for the indicated data set have not been modified because the parameters specified on the REPROTECT command match the existing ones.

System action:

The REPROTECT command is not executed.

User response:

Change the REPROTECT command parameters if a change is required.

CYG262E

User is not an Administrator. Verification could not be completed.

Explanation:

Installation verification could not be completed because the user running it is not an Administrator.

System action:

The job step terminates.

User response:

Rerun the installation verification program as an Administrator.

CYG263W

Data set is not a protected resource.

DSN: *dsn*

VOL: *vol*

Explanation:

The indicated data set is not protected by z/OS Change Tracker. The REPROTECT command can only be used for protected data sets.

System action:

The REPROTECT command is not executed.

User response:

Specify a protected data set. Run the SHOW=UNIVERSE command to see a list of protected data sets.

CYG264W

REPROTECT failed.

DSN: *dsn*
VOL: *vol*
RC: *code*
R0: *code*

Explanation:

The REPROTECT command has failed for the indicated data set.

System action:

The REPROTECT command is not executed.

User response:

If you cannot determine the cause of the error, contact IBM Support.

CYG265I *func***type successful: VOL=vol DSN=dsn**

Explanation:

The SURVEILLANCE or VERIFY command for the indicated data set has successfully completed.

System action:

The system continues.

User response:

None. Informational message.

CYG266W **Verification OR Surveillance has failed.**
DSN: *dsn*
VOL: *vol*

Explanation:

The SURVEILLANCE or VERIFY command for the indicated data set has failed. This is due to an inconsistency in the control file records for that data set. The data set is still protected.

System action:

The command is not executed.

User response:

To fix the error in the control file, UNPROTECT and then PROTECT the data set again. If this does not solve the problem, contact IBM for support.

CYG267W **Cataloged data set expected.**
DSN: *dsn*

Explanation:

The indicated data set was protected as a cataloged data set but it no longer exists in the catalog.

System action:

The SURVEILLANCE command is not executed.

User response:

Either recatalog the data set or remove protection using the UNPROTECT command.

CYG282E **Process failed. VOL: volser DSN: dsn**

Explanation:

The COPYDIFF command has failed for the indicated data set.

System action:

The job step continues.

User response:

If you cannot determine the cause of the error, contact IBM Support.

CYG284W **Archive not found.**
DSN: *dsn*
VOL: *vol*
Member: *mem*
Timestamp: *timestamp*

Explanation:

The indicated backup to be archived was not found.

System action:

The ARCHIVE command is not executed.

User response:

Check that the specified parameters are correct.

CYG285I

Serialization requested. Function may have to wait.

Explanation:

A function has requested serialization on the repository control file and may have to wait.

System action:

The job waits until the control file is available.

User response:

None. Informational message.

CYG285W

Archive not found.

DSN: *dsn*

VOL: *vol*

Member: *mem*

Version: *nr*

Explanation:

The indicated backup to be archived was not found.

System action:

The ARCHIVE command is not executed

User response:

Check that the specified parameters are correct.

CYG286I

Archive successful.

DSN : *dsn*

VOL : *vol*

Member: *mem*

as of : *timestamp*

Explanation:

The ARCHIVE command for the indicated data set and member has successfully completed.

System action:

The system continues.

User response:

None. Informational message.

CYG286W

Archive failed.

DSN : *dsn*

VOL : *vol*

Member: *mem*

as of : *timestamp*

RC : *code*

Explanation:

Archival of the specified member has failed.

System action:

The ARCHIVE command is not executed.

User response

Run the following command to report all existing backups for this member:

```
SHOW=BACKUPS=(DSN=dsname, MEM=member-name)
```

Ensure that the member being archived exists in the live system.

CYG287I	Archive successful. DSN : <i>dsn</i> VOL : <i>vol</i> Archived Member : <i>mem</i> Version: <i>nr</i>
----------------	--

Explanation:
The ARCHIVE command for the indicated data set and member has successfully completed.

System action:
The system continues.

User response:
None. Informational message.

CYG287W	Archive failed. DSN : <i>dsn</i> VOL : <i>vol</i> Member : <i>mem</i> Version: <i>nr</i> RC : <i>code</i>
----------------	--

Explanation:
Archival of the specified member has failed.

System action:
The ARCHIVE command is not executed.

User response

Run the following command to report all existing backups for this member:

```
SHOW=BACKUPS=(DSN=dsname, MEM=member-name)
```

Ensure that the member being archived exists in the live system.

CYG290W	No Archived Backups found for the DSN or the MEM VOL= <i>vol</i> DSN= <i>dsn</i> Specify MEM=* if applicable
----------------	--

Explanation:
No backup for the specified member was found in the archive system.

System action:
The SHOW=ARCHIVES command is not executed.

User response:
Check that the specified parameters are correct. Specify MEM=* to show all archived members for the data set.

CYG291W	No Archived Actions found for the DSN or the MEM VOL= <i>vol</i> DSN= <i>dsn</i> Specify MEM=* if applicable
----------------	--

Explanation:
When a protected data set is removed from the system, all existing recorded change actions along with all existing member backups are moved from the live system to the archive system. This message indicates that a request to report change actions from the archive system has failed. Either there are no change actions existing in the archive system, or the request is incorrect.

System action:

The command is not executed.

User response

Run the following command to report all existing archived actions in the system:

```
SHOW=ARCHIVEDACTIONS=(DSN=dsname, MEM=*)
```

If you could not determine the cause of the warning, contact IBM Support.

CYG292I**Summary statistics:**

count **TOTAL MATCHED DATA SETS**

count **TOTAL MISMATCH DSN(S) ON BASE**

count **TOTAL MISMATCH DSN(S) ON TARGET**

count **TOTAL DATA SETS WITH DIFFERENCES**

Explanation:

Match and mismatch counts for the GROUPCOMPARE command.

System action:

The system continues.

User response:

None. Informational message.

CYG292W**AUTOARC function ended with error. R15=code****Explanation:**

The AUTOARC command failed with the indicated error code.

System action:

The AUTOARC command is not executed.

User response:

Check for other messages. If you cannot determine the cause of the error contact IBM for support.

CYG294I**Record in Control File resynchronized with directory after PDS compressed.**

DSN: *dsn*

VOL: *vol*

Explanation:

The RESYNC command for the indicated data set has successfully completed.

System action:

The system continues.

User response:

None. Informational message.

CYG294W**Function failed: R0=code, R15=code**

DSN: *dsn*

VOL: *vol*

Explanation:

The RESYNC command for the indicated data set has failed with the indicated error code.

System action:

The RESYNC command is not executed.

User response:

None. Informational message.

CYG295E**CHECKIN completed with error. rc=code**

Member=mem, VOL=vol DSN=dsn

Explanation:

A request to check in a member of a protected data set has failed. This message is issued if the member is not checked out or the member does not exist.

System action:

The job step terminates.

User response:

Check that the correct member has been specified. To see the list of protected data sets and the checkout status of their members, run the following command: SHOW=UNIVERSE

CYG296I
Summary statistics:

count **TOTAL MATCHED PATHS**
count **TOTAL MISMATCH PATHS ON BASE**
count **TOTAL MISMATCH PATHS ON TARGET**
count **TOTAL PATHS WITH DIFFERENCES**
count **EXCLUDED PATHS (BASE and TARGET)**
count **DUPLICATE PATHS (BASE and TARGET)**
count **REJECTED PATHS (BASE and TARGET)**
count **LONG-PATHS>504 (BASE and TARGET)**

Explanation:

Match and mismatch counts for the PATHCOMPARE command.

System action:

The system continues.

User response:

None. Informational message.

CYG297I

z/OS Change Tracker has been successfully installed.

Explanation:

The installation verification program has run successfully.

System action:

The system continues.

User response:

None. Informational message.

CYG297W

Duplicate data set name not placed in the LIST.

VOL: *vol*
DSN: *dsn*

Explanation:

The INCLUDE command has selected a name which is already on a data set name list.

System action:

The duplicate name is ignored.

User response:

None.

CYG298I

Number of changes written to TARCMD5 file: *count*

User response:

None. Informational message.

CYG300W

DSCB1 does not exist.

DSN: *dsn*
VOL: *vol*

Explanation:

While processing the TOKEN command an OBTAIN was issued for the indicated data set, but it was not found on the indicated volume. This could be an internal error.

System action:

The TOKEN command is not executed.

User response:

Contact IBM Support.

CYG302W

Relative Track Number is too large *nr*

DSN: *dsn*

Data sets on Extended Address Space are not supported.

Explanation:

The indicated data set has a relative track number which is too large. This may be because the data set is on an Extended Address Space.

System action:

The data set is not tokenized.

User response:

Contact IBM Support.

CYG303W

Begin TTR is not found *cchh*

Member Count= *nr*

Member Name = *mem*

DSN = *dsn*

Explanation:

The indicated data set has an invalid TTR.

System action:

The data set is not tokenized.

User response:

Contact IBM Support.

CYG304W

CCHHR= *cchhr*

DSN = *dsn*

has a wrong Segment Length.

Explanation:

The indicated data set has an invalid segment length.

System action:

The data set is not tokenized.

User response:

Contact IBM Support.

CYG305W

Relative track number is too large

Member = *mem*

DSN = *dsn*

Explanation:

The indicated data set has a relative track number which is too large.

System action:

The data set is not tokenized.

User response:

Contact IBM Support.

CYG306W

FB data set. BLKSIZE is 0 or is not evenly divisible by LRECL.

CCHHR = *cchhr*

DSN = *dsn*

LRECL = *size*

BLKSIZE= *size*

Explanation:

The indicated data set has an unsupported structure.

System action:

The data set is not tokenized.

User response:

Contact IBM Support.

CYG307W	VB data set. BLKSIZE not equal to size from BLK-Descriptor. CCHHR = <i>cchhr</i> DSN = <i>dsn</i> BLK-Descriptor = <i>nr</i> BLKSIZE = <i>size</i>
----------------	---

Explanation:

The indicated data set has an unsupported structure.

System action:

The data set is not tokenized.

User response:

Contact IBM Support.

CYG308W	VB data set. Segment too short or span error. CCHHR <i>cchhr</i> DSN = <i>dsn</i> RDW = <i>value</i>
----------------	---

Explanation:

The indicated data set has an unsupported structure.

System action:

The data set is not tokenized.

User response:

Contact IBM Support.

CYG309W	Allocation failed. VOLSER= <i>vol</i> RC =<i>code</i>
----------------	--

Explanation:

Allocation failed for the indicated volume.

System action:

The TOKEN command is not executed.

User response:

Contact IBM Support.

CYG310W	Unsupported data set: DSN=<i>dsn</i>
----------------	---

Explanation:

The indicated data set is not supported for tokenization.

System action:

The data set is not tokenized.

User response:

Check that the data set is a supported type. Partitioned data sets, physical sequential files, direct access files, and zFS files are supported. VSAM files are not supported.

CYG313W	PDS directory block encountered invalid indicator. Value: <i>blkind dsn</i> Use IEBCOPY to copy the PDS, then use copied version.
----------------	--

Explanation:

During the processing of the directory portion of a PDS, an error was encountered.

System action:

The data set is not tokenized.

User response:

IEBCOPY may resolve the problem. Otherwise contact IBM Support.

CYG314W

Obtain error for DSCB 5 *dsn*

Explanation:

An OBTAIN error occurred for the indicated data set.

System action:

The TOKEN command is not executed.

User response:

Contact IBM Support.

CYG315W

**Member *mem* is an alias of *mem*
They are at the same TTR *ttr***

Explanation:

An alias member is not truly an alias.

System action:

The TOKEN command is not executed.

User response:

Contact IBM Support.

CYG322W

No extents in data set: *dsn*

Explanation:

The indicated data set is corrupted. It has no extents.

System action:

The data set is not tokenized.

User response:

Contact your local support for assistance.

CYG323I

VTOC Index can not be processed: *dsn*

Explanation:

SYS1.VTOCIX data sets are not processed.

System action:

The system continues.

User response:

None. Informational message.

CYG324I

VVDS can not be processed: *dsn*

Explanation:

SYS1.VVDSIX data sets are not processed.

System action:

The system continues.

User response:

None. Informational message.

CYG325W

**Non-Executable Load Module: *member*
In library : *dsname*
Process is aborted for this data set**

Explanation:

During tokenization of a load library, a load module was encountered that did not fit the criterion of a load module. This data set can not be tokenized.

System action:

Tokenization is aborted for this data set.

User response:

If you cannot determine the cause of the error Contact IBM Support.

CYG326W **zFS files are skipped: dsn**

Explanation:

The indicated data set is a zFS data set and cannot be tokenized.

System action:

The data set is not tokenized.

User response:

You must specify a specific path.

CYG327I **Broadcast file can not be processed: dsn**

Explanation:

The SYS1.BROADCAST data set is not processed.

System action:

The system continues.

User response:

None. Informational message.

CYG328W **Encrypted file not processed: dsn**

Explanation:

The indicated data set is an encrypted data set and cannot be tokenized.

System action:

The data set is not tokenized.

User response:

None.

CYG401E **Module *module* was not found. Install of exit failed.**

Explanation:

The indicated module was not found in the LPA. The CYGMLPCL, CYGMLPG, CYGMLPOP, and CYGMLPST modules in the SCYGLPA library must be loaded into the LPA for use by the z/OS Change Tracker started task (STC).

System action:

The STC is not started.

User response

Use the following command to load the modules in the SCYGLPA library into the LPA:

```
SETPROG LPA,ADD,MASK(*),DSN(CYG.SCYGLPA)
```

CYG402W **Dynamic install of the exit failed: *exit*
Exit not activated.**

Explanation:

Installation of the indicated dynamic exit failed.

System action:

The STC is terminated.

User response:

Check for other messages. If you cannot determine the cause of the error, contact IBM for support.

CYG404E **CYGSTC task is already active. Execution terminated.**

Explanation:

Only one z/OS Change Tracker STC can be started at a time. Multiple STCs can be started on multiple LPARs.

CYG406I **Install step has failed. Shutdown initiated.****Explanation:**

Shutdown of the STC has been initiated because a dynamic exit did not install.

System action:

The STC is terminated.

User response:

None. Informational message.

CYG407I **Waiting for attached subtasks to end.****Explanation:**

The STC is waiting for the REFRESH and EXTRACT subtasks to terminate.

System action:

The STC waits until the subtasks have ended and then continues its shutdown.

User response:

None. Informational message.

CYG408I **REFRESH task detached. CC: *system code user code*****Explanation:**

The REFRESH subtask was successfully detached.

System action:

The STC continues its shutdown.

User response:

None. Informational message.

CYG408W **REFRESH task detached. CC: *system code user code*****Explanation:**

The REFRESH subtask detach completed with a non-zero return code.

System action:

The STC continues its shutdown.

User response:

Check for other messages. If you cannot determine the cause of this warning, contact IBM Support.

CYG409I **EXTRACT task detached. CC: *system code user code*****Explanation:**

The EXTRACT subtask was successfully detached.

System action:

The STC continues its shutdown.

User response:

None. Informational message.

CYG409W **EXTRACT task detached. CC: *system code user code*****Explanation:**

The EXTRACT subtask detach completed with a non-zero return code.

System action:

The STC continues its shutdown.

User response:

Check for other messages. If you cannot determine the cause of this warning, contact IBM Support.

CYG410I **Resource was not deleted using the UNPROTECT cmd.
VOL: *vol* DSN: *dsn*****Explanation:**

| The data set *dsn* is protected but no longer exists.

User response:

None. Informational message.

CYG411W

Member not backed up. READ access required.

VOL: *vol* **DSN:** *dsn*

MEM: *member*

Explanation:

The STC was unable to back up member *member* because it does not have read access to the data set *dsn*.

System action:

The system continues.

User response:

If backups of the member are required, grant the started task read access to the data set. Otherwise, unprotect and then protect the data set again, specifying INITIAL=NEVER on the PROTECT command so that backups are not performed.

CYG412W

90-day trial period expired. STC shutting down.

Explanation:

The z/OS Change Tracker started task is shutting down because it is running using the 90-day trial license feature and the 90-day trial period has expired.

System action:

The z/OS Change Tracker started task terminates.

User response:

The licensed feature must be purchased and enabled to continue using IBM z/OS Change Tracker.

CYG423I

Removing z/OS Change Tracker exits.

Explanation:

| The z/OS Change Tracker shutdown procedure is about to remove its dynamic exits.

System action:

| The STC continues its shutdown.

User response:

None. Informational message.

CYG427I

z/OS Change Tracker STC protection warning. RC=code

OBJ=dsn, vol

R14=address

Explanation:

The CYGSYSO subtask detected an error while processing the indicated data set. This message can occur if protection for a data set is removed or altered at the same time as it is being updated.

System action:

Processing of the indicated data set is skipped.

User response:

Check that the protection status of the data set is as it should be. If the message continues to occur, contact IBM Support.

CYG602I

CYGSYSO detached. CC: Scode Ucode

Explanation:

The CYGSYSO subtask was successfully detached.

System action:

The STC continues its shutdown.

User response:

None. Informational message.

CYG602W **CYGSYSO detached. CC: Scode Ucode****Explanation:**

The CYGSYSO subtask detach completed with a non-zero return code.

System action:

The STC continues its shutdown.

User response:

Check for other messages. If you cannot determine the cause of the error, contact IBM Support.

CYG603E **The CYGPARM DD cannot be specified in the STC job.****Explanation:**

A CYGPARM DD cannot be specified in the z/OS Change Tracker started task job or procedure.

System action:

The STC is not started.

User response:

Remove the CYGPARM DD statement from the started task job or procedure. The CYGPARM member is accessed from the PARMLIB concatenation.

CYG604E **Submitter of the STC job must be an Administrator.****Explanation:**

The z/OS Change Tracker started task must be started by a user who is defined as a z/OS Change Tracker Administrator.

System action:

The STC is not started.

User response:

Ensure that the submitter of the STC job or procedure is defined as an Administrator.

CYG605E **PARM is not supported for CYGSTC.****Explanation:**

A parameter cannot be specified on the EXEC PGM=CYGSTC statement in the z/OS Change Tracker started task job or procedure.

System action:

The STC is not started.

User response:

Remove any parameters specified on the EXEC PGM=CYGSTC statement.

CYG610W **Volser vol is NOT online.****Explanation:**

Reading of a VTOC of a volume has been requested but the volume is not online.

System action:

The command is not executed.

User response:

If you cannot determine the cause of this warning, contact IBM Support.

CYG611E **VTOC read failed.**
VOL: *vol*
CCHH: *cchh*
IOB: *iob*
ECB: *ecb*
WRK1: *work1*
WRK2: *work2*
PROG: *chpgm*
READ: *bytes*

Explanation:

An error occurred while reading the VTOC of the volume *vol*.

System action:

The command is not executed.

User response:

If you cannot determine the cause of this warning, contact IBM Support.

CYG612W

Unknown DASD device: *vol*

Explanation:

An unsupported device type has been detected when attempting to read a VTOC.

System action:

The command is not executed.

User response:

If you cannot determine the cause of this warning, contact IBM Support.

CYG621E

No room in QSLOT memory. Restart the STC. RC=*code*

Explanation:

An attempt is being made to start the STC but it is already active.

System action:

The STC is not started.

User response:

Only one instance of the STC can be active at a time. Shut down the currently running instance before attempting to start.

CYG622E

Severe Error: Vector is missing.

Explanation:

The common vector memory block is missing. This is an internal error.

System action:

The job step terminates.

User response:

Contact IBM Support.

CYG624E

Severe Error: Restart the task and report the problem.

Explanation:

An internal error has occurred.

System action:

The job step terminates.

User response:

If the problem persists, contact IBM Support.

CYG642W

**Maximum member record count of *cnt* exceeded.
Backup of this member is skipped.
Member=*mem* DSN=*dsn***

Explanation:

The number of records in the indicated member exceeds 500,000. z/OS Change Tracker does not back up members with more than 500,000 records.

System action:

The command is executed but the member backup is not performed.

User response:

To back up this member, first reduce the number of records.

CYG643W

Begin TTR not found *cchh* member *mem* of DSN *dsn*

Explanation:

During tokenization of a PDS member, an invalid address was found. This indicates a structural error within the PDS.

System action:

The member is not tokenized.

User response:

If the problem persists, contact IBM Support.

CYG644W
CCHHR *cchhr* of DSN *dsn* has a wrong segment length.
Explanation:

During Tokenization of a PDS member, an invalid TTR was found.

System action:

The member is not tokenized.

User response:

If the problem persists, contact IBM Support.

CYG645W
**Backup aborted. Previously added segments are removed:
Member=*mem*, DSN=*dsn***
Explanation:

The backup of the member was canceled due to an internal error.

System action:

The member is not tokenized.

User response:

If the problem persists, contact IBM Support.

CYG646W
**Incorrect blocksize.
CCHHR *cchhr* of dsn : *dsn*
LRECL = *size* BLKSIZE = *size***
Explanation:

Tokenization of an FB member failed due to an internal error.

System action:

The member is not tokenized.

User response:

If the problem persists, contact IBM Support.

CYG647W
**Incorrect block length.
CCHHR *cchhr* of dsn : *dsn*
BDW = *value* BLKL = *value***
Explanation:

Tokenization of a VB PDS member failed due to an internal error.

System action:

The member is not tokenized.

User response:

If the problem persists, contact IBM Support.

CYG648W
**Segment too short.
CCHHR *cchhr* of dsn : *dsn*
RDW = *value***
Explanation:

Tokenization of a VB PDS member failed due to an internal error.

System action:

The member is not tokenized.

User response:

If the problem persists, contact IBM Support.

CYG649E

**Member *mem* already exists in the recovery PDS.
DSN: *dsn***

Explanation:

A member with the same name exists in the recovery PDS.

System action:

The member is not recovered.

User response:

Rename or remove the member in the recovery PDS, or specify a different recovery PDS.

CYG650E

**STOW failed.
DSN: *dsn*
Member: *mem*
RC: *code*
Action flag: *flag***

Explanation:

During recovery of a member from the live backup, the STOW failed. This is probably due to insufficient directory space in the recovery PDS.

System action:

The member is not recovered.

User response:

Recreate the OUTDSN recovery data set as a PDSE or a PDS with enough directory space.

CYG651E

Output data set must be empty.

Explanation:

During recovery of a PDS member, a check is made to ensure that the recovery PDS is empty.

System action:

The member is not recovered.

User response:

Ensure you are using the correct recovery PDS.

CYG652E

**Recover failed: Member *mem* Token *token*
Reason: *code***

Explanation:

Recovery of a PDS member has failed.

System action:

The member is not recovered.

User response:

If you cannot determine the cause of the error, contact IBM Support.

CYG653E

Recovery needs a PDS with LRECL= *size*

Explanation:

Recovery of a PDS member failed because the LRECL of the recovery PDS is not the same as that of the original PDS.

System action:

The member is not recovered.

User response:

Check that the LRECL of the recovery PDS is the same as that of the original PDS. If you cannot determine the cause of the error, contact IBM Support.

CYG654E

Directory failed to open for PDS *dsn*

Explanation:

Recovery of a PDS member failed because of a directory error.

System action:

The member is not recovered.

User response:

Ensure that the recovery data set is a PDS with sufficient directory space. If you cannot determine the cause of the error, contact IBM Support.

CYG655E**Open failed for PDS *dsn*****Explanation:**

Recovery of a PDS member failed because of an open error.

System action:

The member is not recovered.

User response:

If you cannot determine the cause of the error, contact IBM Support.

CYG656E**Directory has only *nr* blocks but needs at least *nr* blocks.****Explanation:**

Recovery of a PDS member failed because the recovery PDS does not have sufficient directory space.

System action:

The member is not recovered.

User response:

Re-allocate the recovery PDS with enough directory space and try the recovery again. If the problem persists, contact IBM Support.

CYG657W**Expansion error.**

Text: *text*

cc: *code*

Explanation:

An error occurred when expanding a compressed backup.

System action:

The member is not recovered.

User response:

Check for other messages. If you cannot determine the cause of the error, contact IBM Support.

CYG658I**Member *mem* recovered (Add): *dsn*****Explanation:**

Member *mem* was recovered to the indicated data set.

User response:

None. Informational message.

CYG660W**DSCB1 does not exist.**

DSN: *dsn*

VOL: *vol*

Explanation:

An OBTAIN was issued for the indicated data set but the data set does not exist on volume *vol*.

System action:

The command is not executed.

User response:

Check for other messages. If you cannot determine the cause of the error, contact IBM Support.

CYG661W**Directory error.**

DSN: *dsn*

Explanation:

During processing of the directory portion of a PDS, a structural inconsistency was detected.

System action:

The command is not executed.

User response:

Check for other messages. If you cannot determine the cause of the error, contact IBM Support.

CYG662W

Volume *vol* allocation failed. RC = *code*

Explanation:

Allocation of the specified volume failed.

System action:

The command is not executed.

User response:

Check for other messages. If you cannot determine the cause of the error, contact IBM Support.

CYG663I

**Data set is not a PDS.
DSN: *dsn***

Explanation:

The indicated data set is not a PDS.

System action:

The system continues.

User response:

Check for other messages. If you cannot determine the cause of the error, contact IBM Support.

CYG663W

**Data set is not a PDS.
DSN: *dsn***

Explanation:

The indicated data set is not a PDS.

System action:

The command is not executed.

User response:

Check for other messages. If you cannot determine the cause of the error, contact IBM Support.

CYG664W

No volume list supplied.

Explanation:

An error occurred during directory processing.

System action:

The command is not executed.

User response:

Check for other messages. If you cannot determine the cause of the error, contact IBM Support.

CYG665W

**Multiple volume PDS is not supported.
DSN=*dsn***

Explanation:

Directory processing of a multi-volume PDS is not allowed.

System action:

The command is not executed.

User response:

Check for other messages. If you cannot determine the cause of the error, contact IBM Support.

CYG666W

PDS directory block has invalid indicator.

DSN: *dsn*
Value: *value*

Explanation:

Directory processing of the PDS has failed.

User response:

Try using IEBCOPY to copy the PDS and then use the copied version. Otherwise, check for other messages. If you cannot determine the cause of the error, contact IBM Support.

CYG667W

Obtain error for continuation DSCB in *dsn*

Explanation:

The PDS has a structural inconsistency.

System action:

The command is not executed.

User response:

Check for other messages. If you cannot determine the cause of the error, contact IBM Support.

CYG669W

**Relative track number is too large: *value*
Data sets on Extended Address Space are not supported.
DSN: *dsn***

Explanation:

The relative track number is too large. The data set may be on Extended Address Space, which is not supported.

System action:

The command is not executed.

User response:

Check where the data set resides.

CYG670E

**STOW failed.
DSN: *dsn*
Member: *mem*
RC: *code*
Action flag: *flag***

Explanation:

During recovery of an Archived version, the STOW failed. This is probably due to insufficient directory space in the recovery PDS.

User response:

If you cannot determine the cause of the error, contact IBM Support.

CYG670I

**Number of Alias Members = *cnt*
Number of Orphaned Members = *cnt***

User response:

None. Informational message.

CYG670W

**Number of Alias Members = *cnt*
Number of Orphaned Members = *cnt***

CYG671E

Directory must be empty.

Explanation:

During recovery of an Archived version, a check is made to ensure that the recovery PDS is empty.

User response:

Make sure you are pointing to the correct recovery PDS.

CYG672E

**Recover failed: Member *mem* Token *token*
Reason: *code***

Explanation:

During recovery of an Archived version, the recovery failed. This is an internal error.

User response:

If you cannot determine the cause of the error, contact IBM Support.

CYG673E	Archival recovery needs a PDS with LRECL= <i>size</i>
----------------	--

Explanation:

Recovery PDS must be compatible with the original PDS. Check the LRECL of the recovery PDS.

User response:

If you cannot determine the cause of the error, contact IBM Support.

CYG674E	Directory open failed for PDS <i>dsn</i>
----------------	---

Explanation:

This condition is detected during recovery of an Archived version. The Recovery PDS must be a PDS with proper directory space. This message indicates that the directory of the PDS was incorrect.

User response:

If you cannot determine the cause of the error, contact IBM Support.

CYG675E	Open failed for PDS <i>dsn</i>
----------------	---------------------------------------

Explanation:

This condition is detected during recovery of an Archived version. Recovery PDS failed to open.

User response:

If you cannot determine the cause of the error, contact IBM Support.

CYG676E	Directory has only <i>nr</i> blocks but needs at least <i>nr</i> blocks.
----------------	---

Explanation:

This condition is detected during recovery of an Archived version. Recovery PDS does not have sufficient directory space.

User response:

Re-allocate the Recovery PDS with the required directory space and try the recovery again. If the problem persists, contact IBM Support.

CYG677W	Expansion error. Text: <i>text</i> cc: <i>code</i>
----------------	---

Explanation:

This condition is detected during recovery of an archived version. Expansion of the compressed backup version failed.

User response:

Contact IBM Support.

CYG678I	Member <i>mem</i> added to data set <i>dsn</i>
----------------	---

User response:

None. Informational message.

CYG679I	Member <i>mem</i> replaced in data set <i>dsn</i>
----------------	--

User response:

None. Informational message.

CYG700E	Allocation failed. VOL: <i>vol</i> SVC99 Error: <i>code</i> Info: <i>code</i>
----------------	--

Explanation:

Volume allocation has failed.

User response:

Check the SVC 99 return code in the IBM Authorized Assembler Services Guide. If you cannot determine the cause of the error, contact IBM Support.

CYG700I	Dynamic allocation failed. SVC99 Error: code Info: code
----------------	--

User response:
None. Informational message.

CYG701W	Allocation failed. DSN: dsn SVC99 Error: code Info: code
----------------	---

Explanation:
Data set allocation has failed.

User response:
Check the SVC 99 return code in the IBM Authorized Assembler Services Guide. If you cannot determine the cause of the error, contact IBM Support.

CYG702E	open/close error on VSAM cluster dsn MACRF = macro format R15 = code R0 = code ERROR= acb error code
----------------	---

Explanation:
A VSAM OPEN or CLOSE SVC has failed.

User response:
This is an internal error. Contact IBM Support.

CYG703E	VSAM I/O request failed. KEY =key RPL =type CLUSTER =name MACRF =form RPL OPTCD=option codes R15 = code R0 = code FDBK = code
----------------	--

Explanation:
A VSAM I/O request has failed. This is an internal error.

User response:
Contact IBM Support.

CYG704E	OBTAIN ERROR - VOL= vol R15= code
----------------	--

Explanation:
During reading a VTOC, OBTAIN of VTOC has failed. This is an internal error.

User response:
Contact IBM Support.

CYG705E	OPEN error. VOL: vol
----------------	-----------------------------

Explanation:
During READ of a VTOC, OPEN of the VTOC failed. This is an internal error.

User response:
Contact IBM Support.

CYG705W	No USSENV record found.
----------------	--------------------------------

User response:
No action is required.

CYG706T	Control-File Add : key
----------------	-------------------------------

CYG707T	Control-File Upd : key
----------------	-------------------------------

CYG708T	Control-File Del : key
----------------	-------------------------------

CYG708W	No LPAR record is found.
User response: No action is required.	
CYG709W	No TOKEN record is found.
Explanation: No data set has been tokenized.	
User response: No action is required.	
CYG710I	count Tokenized files were found.
User response: None. Informational message.	
CYG711I	No PROTECTION record is found.
User response: None. Informational message.	
CYG711W	No PROTECTION record is found.
User response: No action is required.	
CYG712I	count protected libraries were found.
User response: None. Informational message.	
CYG800I	Located ENV record: <i>dswn</i> Date Record Added: <i>date</i> Date Last Tokenized: <i>date</i> USSENV Comment: <i>cmt</i>
User response: None. Informational message.	
CYG800W	USSENV record already exists. Skipped. Node: <i>dsn</i>
User response:	
CYG802E	Checkout is allowed for locked data sets only.
User response:	
CYG802I	Opened successfully : <i>dsn</i>
User response: None. Informational message.	
CYG803I	WARNINGS: <i>cnt</i> ERRORS : <i>cnt</i>
User response: None. Informational message.	
CYG803W	USSENV record not found. Node: <i>node</i>
User response:	
CYG804I	Copy operation from FB to VB begins: Input: <i>dsn</i> Output: <i>dsn</i>

User response:

None. Informational message.

CYG805I

USSENV record successfully created.
Node: *node*

User response:

None. Informational message.

CYG805W

Token File not found or allocation failed.
DSN: *dsn*

User response:

CYG806I

USSENV record successfully erased.
Node: *node*

User response:

None. Informational message.

CYG806W

Full-Track Read failed VOL: *vol*
CCH: *cchh*
ECB: *ecb*
IOBTYPE: *iobtyp*
IOBECB: *iobedb*
IOBWORK1: *iobwork1*
IOBWORK2: *iobwork2*
IOBCHPGM: *iobpgm*
IOBDCB: *iobdcb*
IOBWKCCW: *iobwkccw*
IOBMBB: *iobmbb*
IOBCCHH: *iobcchh*
BYTESRD: *bytes*

Explanation:

- A data set full-track read operation has failed. This is an internal error.

System action:

- The command was not executed.

User response:

Contact IBM Support.

CYG807W

Full-Track Read Verification failed.
REFCCHH: *cchh*
VOL: *vol*
CCHHRVER: *num*
Found: *num*

Explanation:

- A data set full-track verification operation has failed. This is an internal error.

System action:

- The command was not executed.

User response:

Contact IBM Support.

CYG808I

***cnt* DSNs checked for MODS**
***cnt* DSNs have REPORTABLE MODS**
***cnt* DSNs determined to be NEW**

User response:

None. Informational message.

CYG808W	Requested USSENV record is not found. Record must be added first using USSENV=(NODE=,ACT=ADD)
----------------	--

User response:

CYG809W	Obtain failed. Data set not protected. DSN: <i>dsn</i> VOL: <i>vol</i>
----------------	---

Explanation:

During protection of a data set, an OBTAIN was issued to ensure that the data set resides on the volume specified in the catalog. The OBTAIN SVC has failed.

User response:

If you cannot determine the cause of this error, contact IBM Support.

CYG810W	Open failed. DSN: <i>dsn</i>
----------------	-------------------------------------

Explanation:

During protection of a data set, an OPEN SVC was issued to ensure that the administrator has RACF authority to access the file. OPEN SVC has failed.

User response:

CYG811I	Open successful. DSN: <i>dsn</i>
----------------	---

User response:

None. Informational message.

CYG811W	Record already exists. Skipped. PATH: <i>path</i>
----------------	--

User response:

CYG812W	Close failed. DSN: <i>dsn</i>
----------------	--------------------------------------

Explanation:

During protection of a data set, a CLOSE SVC was issued immediately after OPEN. The CLOSE SVC has failed. This is an internal error.

User response:

If you cannot determine the cause of the error, contact IBM Support.

CYG813W	Data set is not a PDS. DSN: <i>dsn</i>
----------------	---

Explanation:

During protection of a data set, it was determined that the data set is not a PDS. Only PDS(E) data sets can be protected.

CYG814W	Data set cataloged to a migrate volume. Data set not protected. DSN: <i>dsn</i> VOL: <i>vol</i>
----------------	--

Explanation:

During protection of a data set, it was detected that the data set was migrated. "Protection" is skipped.

CYG815W	DSCB1 attribute is not valid: DSN : <i>dsn</i> VOL : <i>vol</i> DS1DSORG : <i>dsorg</i> DS1RECFM : <i>recfm</i> DS1LRECL : <i>recl</i> DS1BLKL : <i>blk</i>
----------------	--

Explanation:

During protection of a data set, it was determined that the data set has an invalid DSCB1 format.

User response:

If you cannot determine the cause of the warning, contact IBM Support.

CYG816W	Volume is offline. Data set not protected. DSN: <i>dsn</i> VOL: <i>vol</i>
----------------	---

Explanation:

During protection of a data set, the volume where the data set resides was not online.

CYG817W	Protection not removed. Resource has checked out members. DSN: <i>dsn</i> VOL: <i>vol</i>
----------------	--

Explanation:

An attempt was made to Remove Protection from a data set. It was determined that this data set has one or several checked out members. The Remove can not be accomplished until these members are checked in.

CYG819I	Protected resource is successfully removed. DSN: <i>dsn</i> VOL: <i>vol</i>
----------------	---

User response:

None. Informational message.

CYG819W	READ access is required to protect a data set. DSN: <i>dsn</i> VOL: <i>vol</i>
----------------	---

CYG820I	<i>cnt</i> component commands written to OUTFILE DD.
----------------	---

User response:

None. Informational message.

CYG820W	ADD of record failed. PATH: <i>path</i>
----------------	--

User response:

CYG821E	Unsupported file type.
----------------	-------------------------------

Explanation:

Partitioned data sets, physical sequential files, direct access files and zFS files are supported. VSAM files are not supported.

CYG821I	Record is successfully added. PATH: <i>path</i>
----------------	--

User response:

None. Informational message.

CYG822I	Reconcile statistics: <i>nr</i> SAME <i>nr</i> REP <i>nr</i> ADD <i>nr</i> DEL SOURCE: <i>dsn</i> TARGET: <i>dsn</i>
----------------	---

User response:

None. Informational message.

CYG823I	DELETE: MEM=<i>mem</i> DSN=<i>dsn</i>
----------------	--

User response:

None. Informational message.

CYG823W	DELETE: MEM=<i>mem</i> DSN=<i>dsn</i> RC : <i>code</i>
----------------	---

Explanation:

This warning indicates that during the reconciliation of two PDSs, the specified member is deleted from the target library due to DELETE=YES request in the job.

CYG824E	REFRESH of protection list is forced to exit. Run the commands RESETCOUNT and TRIGGERREFRESH.
----------------	--

User response:

Run the commands RESETCOUNT and TRIGGERREFRESH. If problem persists stop and restart the STC.

CYG824I	The two non-PO files have same contents.
----------------	---

Explanation:

During the compare of two files, it was detected that the two files have the same contents.

CYG824W	The two non-PO files have different contents.
----------------	--

Explanation:

During the compare of two files, it was detected that the two files have different contents.

CYG825I	Record is successfully erased. PATH: <i>path</i>
----------------	---

User response:

None. Informational message.

CYG825W	Dynamic allocation failed. DDname: <i>dd</i> DSN: <i>dsn</i> SVC99 error: <i>code</i> info: <i>code</i>
----------------	--

Explanation:

Dynamic allocation of the file has failed.

User response:

Check the SVC 99 return code in the IBM Authorized Assembler Services Guide. If you cannot determine the cause of the error, contact IBM Support.

CYG826E	Invalid PDS structure: <i>dsn</i>
----------------	--

Explanation:

During the tokenization of a PDS, a structural error was detected.

User response:

Contact IBM Support.

CYG826W	USS directory record not found for delete. PATH: <i>path</i>
----------------	---

CYG827I	Function completed.
----------------	----------------------------

User response:

None. Informational message.

CYG827W	Function completed.
----------------	----------------------------

CYG828I	Reference tokens generated successfully. RC: <i>code</i> VOL: <i>vol</i> DSN: <i>dsn</i>
----------------	---

User response:

None. Informational message.

CYG829W	Function must be performed by the Administrator.
----------------	---

Explanation:

Protection of a PDS can only be performed by the z/OS Change Tracker Administrator. The current user is not authorized to protect the PDS.

CYG832W	Unexpected error occurred.
----------------	-----------------------------------

Explanation:

An unexpected error was encountered during a read of a file.

User response:

Contact IBM for support.

CYG833I
Statistics follow:
***cnt* SAME**
***cnt* UPD**
***cnt* ADD**
***cnt* DEL**
***cnt* REN**
***cnt* TOTAL REPORTABLE CHANGES**
User response:

None. Informational message.

CYG835W
Invalid data set name generated by replacement string. Name exceeds 44 characters.
Original: *dsn*
Replacement: *dsn*

CYG836E
OUTFILE DD open failed.
Explanation:

When the command WRITETOFILE is specified an OUTFILE DD must be specified.

User response:

This DD must point to a file with these attributes: PS,FB,80,8000

CYG837E
OUTFILE DD open failed.
Explanation:

When the command WRITETOFILE is specified an OUTFILE DD must be specified.

User response:

This DD must point to a file with these attributes: PS,FB,80,8000

CYG838I
Real-time recorded actions written to OUTFILE DD.
User response:

None. Informational message.

CYG839E
Locate failed. RC: *code*
DSN: *dsn*
Explanation:

During update of a change action in the Control File, a LOCATE command was issued to find the protected data set in the catalog. The SVC 27 has failed.

CYG840I
***cnt* DSNs checked for activity.**
***cnt* DSNs had no activity.**
***cnt* DSNs had activity.**
User response:

None. Informational message.

CYG844E
IEBCOPY failed with severe error.
Explanation:

Internal invocation of IEBCOPY has failed. This is an internal error.

User response:

Contact IBM Support.

CYG845E
IEBCOPY failed with RC=*code*
Explanation:

Internal invocation of IEBCOPY has failed. This is an internal error.

User response:

Contact IBM Support.

CYG845W **IEBCOPY completed with RC=code**

Explanation:

Internal invocation of IEBCOPY has failed. This is an internal error.

User response:

Contact IBM Support.

CYG847E **Data File was not specified in the CYGPARMs member.**

Explanation:

The DATAFILE parameter in the CYGPARMs member has not been specified.

User response:

Edit the CYGPARMs member and correct the error.

CYG850W **Data File is not compatible.**

CYG851W **Control File segmented record is short.**
DSN: dsn

Explanation:

During reading all segments of a segmented record in the z/OS Change Tracker Control File, an error was encountered.

User response

Run the following diagnosis commands on this data set and contact IBM Support.

```
OPTION=DIAGNOSE  
OBJECT=(DSN=protected-pds,VOL=volser)
```

CYG852E **LOCATE for cluster failed: dsn**

Explanation:

During allocation of the Archive Data File, a LOCATE was issued to find the cluster but it failed.

User response:

Check the CYGPARMs member and verify that the setting of the ARCHIVEDATA parameter is correct.

CYG853W **VIEW function returned a ZERO area.**
VOL: vol DSN: dsn
Use the UNPROTECT command to delete and re-protect the resource.

User response:

Use command UNPROTECT to delete and re-protect data set.

CYG854W **Dynamic deallocation of DD AUTOCALL failed.**
SVC99 error: code info: code

Explanation:

During LOCATE of a data set, it was determined that the data set was migrated. De-allocation of the specified DD has failed. Check the SVC 99 return code in the IBM Authorized Assembler Services Guide.

User response:

If you cannot determine the cause of the error, contact IBM Support.

CYG855W **Dynamic allocation failed.**
DDname : dd
Member : mem
DSname : dsn
Volser : vol
SVC99 error: code

info: *code*

Explanation:

Allocation of a PDS member has failed.

User response:

Check the SVC 99 return code in the IBM Authorized Assembler Services Guide. If you cannot determine the cause of the error, contact IBM Support.

CYG856I	COMPARE statistics for: - DSN1: <i>dsn</i> - DSN2: <i>dsn</i> <i>nr</i> TOTAL MEMBER NAMES PAIRED <i>nr</i> MEMBER(S) HAD NO CHANGES <i>nr</i> MEMBER(S) HAD CHANGES <i>nr</i> NEW FILE MEMBER(S) NOT PAIRED <i>nr</i> OLD FILE MEMBER(S) NOT PAIRED
----------------	---

User response:

None. Informational message.

CYG856W	AUTOARC failed during initialization.
----------------	--

CYG857W	COMPARE statistics: <i>nr</i> TOTAL MEMBER NAMES PAIRED <i>nr</i> MEMBER(S) HAD NO CHANGES <i>nr</i> MEMBER(S) HAD CHANGES <i>nr</i> NEW FILE MEMBER(S) NOT PAIRED <i>nr</i> OLD FILE MEMBER(S) NOT PAIRED
----------------	---

Explanation

In comparing two PDSs, some differences were detected. This message displays counts (indicated by *nr*) in the following categories:

- The number of matching member names in both data sets
- The number of matching MEMs that have same content
- The number of matching MEMs that have changes
- The number of members in the NEW file not found in the OLD (Action=ADD)
- The number of members in the OLD file not found in the NEW (Action=DEL)

CYG858I	Number of token records removed: <i>cnt</i>
----------------	--

User response:

None. Informational message.

CYG858W	Structural error in definition records: RC =<i>code</i> TYPE=<i>record-type</i> SEQNR=<i>nr</i> VOL=<i>volser</i> DSN=<i>dsname</i> Use the UNPROTECT command to delete the resource and then re-protect it.
----------------	---

User response:

You must delete the data set using the UNPROTECT command, and then re-protect it.

CYG859E	SHOWCB failed. R15: <i>code</i> VSAM cluster: <i>cluster name</i>
----------------	--

Explanation:

SHOWCB macro has failed. This is an internal error.

User response:

Contact IBM Support.

CYG859I	Member archival limit of 500 has been reached.
----------------	---

User response:

None. Informational message.

CYG859W	AUTOARC can only be run when the STC is not running or has been deactivated using the DEACTIVATESTC command.
----------------	---

CYG860I	Temporary data set allocated successfully: DSN: <i>dsn</i>
----------------	---

User response:

None. Informational message.

CYG860W	Error during ACTADD processing. RC: <i>code</i> VOL: <i>volser</i> DSN: <i>dsname</i> Run CYGCHECK job to verify structure of all records.
----------------	---

Explanation:

An error was detected in the z/OS Change Tracker Control File during adding an activity event.

User response:

Run the job CYGCHECK and contact IBM Support.

CYG861E	Insufficient memory for APF list.
----------------	--

User response:

Report the problem.

CYG861W	Email Alert requires //SYSUT2 and //SYSUT3 DDs in the CYGSTC JCL. Email not sent.
----------------	--

CYG862W	Object is not found. Object: <i>dsn</i>
----------------	--

CYG863I	Function completed successfully. Object: <i>dsn</i>
----------------	--

User response:

None. Informational message.

CYG863W	STC is not active; HALT or UNHALT is skipped.
----------------	--

CYG864I	Number of protected resources: <i>cnt</i>
----------------	--

User response:

None. Informational message.

CYG865E	LPAR record is not found. Cannot proceed.
----------------	--

CYG865I	<i>ddname</i> has <i>cnt</i> bytes.
----------------	--

User response:

None. Informational message.

CYG866I	Numbers reset in REC 0: OBJECTs: <i>cnt</i> CHECKOUTs: <i>cnt</i>
----------------	--

User response:

None. Informational message.

CYG866W	Email text member <i>member</i> does not exist in the EMAILDSN data set. Default member CYGEMDAT will be used.
----------------	---

CYG868W	Default text member CYGEMDAT not found. Email not sent.
----------------	--

CYG869W	CYGENJCL job must exist in EMAILDSN data set. Email not sent.
----------------	--

CYG870W	EMAILDSN data set name not specified in CYGPARMs. Email not sent.
----------------	--

CYG871W	Function is disallowed. Run SHOW=HALT command.
----------------	---

User response:

HALT/UNHALT disallowed.

CYG879T	Data-File - Upd : <i>key</i>
----------------	-------------------------------------

CYG880T	Data-File - Del : <i>key</i>
----------------	-------------------------------------

CYG881E	Insufficient memory adding directory. Function aborted.
----------------	--

CYG881I	Tokens set in memory for path: <i>path</i>
----------------	---

User response:

None. Informational message.

CYG882E	Insufficient memory adding file. Function aborted.
----------------	---

CYG882W	Tokens for path already in memory: <i>path</i>
----------------	---

Explanation:

The Tokens for the files in the zFS path identified in the warning message above, are already present in memory.

User response:

Check your SYSUT1=PATH=*pathname* and SYSUT2=PATH=*pathname* commands to avoid duplication.

CYG883E	Open failed. DD: <i>ddname</i> RC: <i>code</i> DSN: <i>dsn</i>
----------------	---

Explanation:

The Tokens resulting from tokenization of a path are stored in a file pointed to by *ddname*. In this case, the file has failed to open.

User response

Check your JCL to make sure that the specified *ddname* in the command correctly points to a physical sequential file.

```
SYSUT1=PATH=pathname
SAVEDD=DD1
```

CYG884E	Close failed. DD: <i>dd</i> RC: <i>code</i> DSN: <i>dsn</i>
----------------	--

Explanation:

The Token file failed to CLOSE.

User response:

This is an internal error. Contact IBM Support.

CYG885I	Tokens are saved into DSN: <i>dsn</i> Number of files tokenized: <i>cnt</i>
----------------	--

User response:

None. Informational message.

CYG886E	Not an Open Edition token file: DDname: <i>ddname</i> DSname: <i>dsn</i>
----------------	---

Explanation:

During loading of the pre-stored tokens representing a path, the file containing the tokens is referenced by *ddname*. The file does not have proper information in the header record.

User response:

Make sure that *ddname* points to the correct physical sequential file.

CYG886I	USS path commands written to CMD file: <i>dsn</i> Number of commands stored in the file: <i>cnt</i>
----------------	--

User response:

None. Informational message.

CYG887E	The Open Edition Token file is incomplete: DDname: <i>dd</i> DSname: <i>dsn</i>
----------------	--

Explanation:

Reading of the token file into memory ended prematurely. This is an internal error.

User response:

Contact IBM Support.

CYG887W	The .PATHCMD file does not exist. Function aborted. File: <i>dsn</i>
----------------	---

CYG888E	READJFCB failed on DD: ,DDNAME, RC=,(R15)
----------------	--

CYG889I	SYSUT1 SYSUT2 <i>nr nr</i> Matched File Names <i>nr nr</i> SAME Content <i>nr nr</i> DIFFerent Content (UPD) <i>nr nr</i> Mismatched Files (ADD,DEL)
----------------	---

User response:

None. Informational message.

CYG890W	SYSUT1 SYSUT2 <i>nr nr</i> Matched File Names <i>nr nr</i> SAME Content <i>nr nr</i> DIFFerent Content (UPD) <i>nr nr</i> Mismatched Files (ADD,DEL)
----------------	---

Explanation

In comparing the two paths, some differences were detected. This message displays counts of the number of files in SYSUT1 and SYSUT2 (indicated by *nr nr*) in the following categories:

- The number of matching file names in both paths
- The number of matching files that have same content
- The number of matching files that have changes
- The number of files with no corresponding names

CYG891W	Resource is not found in the Control File: DSN= <i>dsn</i> VOL= <i>vol</i>
----------------	---

Explanation:

During an explicit request for a member backup, it was determined that the data set is not a protected data set. Function aborted.

CYG892W	Member is not found for backup: DSN= <i>dsn</i> MEM= <i>mem</i> VOL= <i>vol</i>
----------------	--

CYG893I	Member is successfully backed up. Member= <i>mem</i> VOL= <i>vol</i> DSN= <i>dsn</i> CMT= <i>comment</i>
----------------	---

User response:

None. Informational message.

CYG894W	Member failed to backup: RC= <i>code</i> Member= <i>mem</i> VOL= <i>vol</i> DSN= <i>dsn</i>
----------------	--

Explanation:

An explicit member backup request has failed. This is an internal error.

User response:

If you cannot determine the cause of the error, contact IBM Support.

CYG895W

Resource does not exist or is partially defined.
To clear partial records use the UNPROTECT command.
DSN: *dsn*
VOL: *vol*

Explanation:

During an explicit request for a member backup, it was determined that the data set does not exist.

CYG896W

Member is already checked out. RC=4
Member: *mem* VOL: *vol* DSN: *dsn*

Explanation:

The member is already checked out. It can not be checked out again.

CYG897W

Checkout failed because backup is not generated.
Member: *mem* VOL: *vol* DSN: *dsn*

Explanation:

During a checkout request z/OS Change Tracker attempted to back up the member first, but it failed the backup. Therefore, the member could not be successfully checked out.

User response:

If you cannot determine the cause of the error, contact IBM Support.

CYG898I

Member is checked out: Counted=*cnt*
Member: *mem* VOL: *vol* DSN: *dsn*

User response:

None. Informational message.

CYG898W

Checkout failed because checkout record not added.
Member: *mem* VOL: *vol* DSN: *dsn*

CYG899W

Checked out member was not found. RC=4
Member: *mem* VOL: *vol* DSN: *dsn*

Explanation:

During a checkin request, a READ was issued to find the "Checkout" record in the Control File, but the record was not found.

User response:

Check the data set, the VOLSER and the member name to make sure they are supplied correctly.

CYG900I

Member is checked in: Counted=*cnt*
Member: *mem* VOL: *vol* DSN: *dsn*

User response:

None. Informational message.

CYG901E

Member failed to backup: RC=, (*R15*)
Member: *mem* VOL: *vol* DSN: *dsn*

Explanation:

During a checkin request, an attempt was made to back up the member first. The backup request failed.

User response:

If you cannot determine the cause of the error, contact IBM Support.

CYG902W

Not authorized to use the CHECKIN command.

Explanation:

Checkin can be performed only by the administrator.

CYG903I	Create IEBCOPY RC=code
User response: None. Informational message.	
CYG907W	Adding CSAVE Record to the Control File failed.
CYG911I	Resource has successfully been archived/removed: VOL=vol DSN=dsn
User response: None. Informational message.	
CYG911W	Archive failed for resource. VOL=vol DSN=dsn CC=code RC=code
Explanation: Archive of the protected resource has failed.	
User response: If you cannot determine the cause of the error, contact IBM Support.	
CYG912I	Resource has no backups. Archive not performed.
User response: None. Informational message.	
CYG916I	Reference Tokens generated and written to SYSUT2 VOL=vol DSN=dsn
User response: None. Informational message.	
CYG916W	Insufficient storage for tokenization of PDSE object.
CYG918E	Data set is unavailable. It is already allocated as DISP=OLD. Removal operation begins.
CYG918I	Number of data sets tokenized. Count= cnt
User response: None. Informational message.	
CYG919E	SYSUT1 DD is unavailable. A previous function has failed. To clear the problem LOGOFF and LOGON and then resubmit the job. Removal operation begins.
CYG920E	Allocation failed. Removal operation begins.
CYG930I	SELECTED: VOL=vol dsn
User response: None. Informational message.	
CYG931E	OUTFILE DD supplied for USS but open failed. RC=code
CYG931I	REJECTED: DSN=dsn
User response: None. Informational message.	
CYG932I	DEL/REN : MEM=mem, DSN=dsn
User response: None. Informational message.	
CYG933I	OUTFILE Processing for USS is in effect.
User response: None. Informational message.	
CYG934W	IBM SUPERF failed with RC=code
Explanation:	

A SuperC compare was requested to compare two members but the request failed. This is an internal error.

User response:

If you cannot determine the cause of the error, contact IBM Support.

CYG935E **OUTDD is missing in the job: RC: code**

Explanation:

SuperC requires an OUTDD, but OUTDD was missing when SuperC was called. For a foreground request, it may be an internal error. For batch jobs, make sure that //OUTDD exists.

CYG936E **Error in IEBCOPY routine. Contact support**
R15=code RC=code
VOL=volser DSN=dsname

Explanation:

During creation of an IEBCOPY unload an error was encountered. This is an internal error.

User response:

If you cannot determine the cause of the error, contact IBM Support.

CYG946E **LPAR record was not found in the Control File.**
Dynamic add of the record has failed.

Explanation:

Select "LPAR Settings" to add this record.

User response:

Follow the instructions above.

CYG947I **Action is logged.**

User response:

None. Informational message.

CYG948E **Unknown action type. Function aborted.**

Explanation:

During reading of the log records, an unknown action record was found. This is an internal error.

User response:

Contact IBM Support.

CYG948I **Total number of records: cnt**
Number of records reported: cnt
Number of records erased: cnt

User response:

None. Informational message.

CYG949I **A new LPAR record is added:**
Code: code LPAR-name: lpar

User response:

None. Informational message.

CYG950I **Feedback record is added successfully.**

User response:

None. Informational message.

CYG951I **DASD Protection record is updated.**

User response:

None. Informational message.

CYG952E **LPAR record 2000 was not found.**

Explanation:

Each Control File on a specific LPAR must have an LPAR record identifying that specific LPAR. This is an internal error.

User response:

Contact IBM Support.

CYG954E	Dynamic allocation failed for the unload file.
----------------	---

Explanation:

During unloading the tokens to an unload file, dynamic allocation of the file has failed. This is an internal error.

User response:

Contact IBM Support.

CYG954I	Unload file is created successfully: CC: <i>code</i> VOL: <i>vol</i> DSN: <i>dsn</i>
----------------	---

User response:

None. Informational message.

CYG954W	No CHECKOUT records exists.
----------------	------------------------------------

CYG955E	Open of unload file failed. Open RC=<i>code</i>
----------------	--

Explanation:

Open of the unload file has failed. This is an internal error.

User response:

Contact IBM Support.

CYG955I	<i>cnt</i> checkout records are removed from the Control File.
----------------	---

User response:

None. Informational message.

CYG955W	No log records exists.
----------------	-------------------------------

CYG956E	Bad record segment found. Seqnr=<i>nr</i> of <i>dsname</i>
----------------	---

Explanation:

During reading the token records a segment with an invalid size was encountered. This is an internal error.

User response:

Contact IBM Support.

CYG956I	<i>cnt</i> log records are removed from Control File
----------------	---

User response:

None. Informational message.

CYG956W	FACILITY class not defined to RACF. Class: <i>class</i>
----------------	--

CYG958I	<i>cnt</i> token records in Control File <i>cnt</i> members in all records
----------------	---

User response:

None. Informational message.

CYG959E	Bad record segment found. Seqnr=<i>nr</i> of <i>dsname</i> bytes requested <i>nr</i> bytes in segment <i>nr</i>
----------------	--

Explanation:

During reading the token records a segment with an invalid size was encountered. This is an internal error.

User response:

Contact IBM Support.

CYG960E**Open of *ddname* failed. RC: *code*****Explanation:**

During REMOTECOMPARE of two unload files, open of the file pointed to by the specified *ddname* has failed. This is an internal error.

User response:

Contact IBM Support.

CYG961E***ddname* is not a valid TOKEN FILE.****ErrType=*code* InBlkCnt=*count*****0: Invalid file****1: Invalid Blksize****2: Invalid Lrecl****3: Invalid Record 0****4: Unexpected EOF: CLOSE needed at tokenization****5: Invalid first record****6: Invalid nr of nodes****7: Incompatible token file****Explanation:**

During REMOTECOMPARE of two unload files, the file pointed to by *ddname* is not a valid token file.

User response:

Check the file name to ensure you are pointing to the right file. If you cannot determine the cause of the error, report the problem, including the ErrType and InBlkCnt values, to IBM Support.

Appendix A. Accessibility

Accessible publications for this product are offered through [IBM Documentation \(www.ibm.com/docs/en/zos\)](http://www.ibm.com/docs/en/zos).

If you experience difficulty with the accessibility of any z/OS information, send a detailed message to the [Contact the z/OS team web page \(www.ibm.com/systems/campaignmail/z/zos/contact_z\)](http://www.ibm.com/systems/campaignmail/z/zos/contact_z) or use the following mailing address.

IBM Corporation
Attention: MHVRCFS Reader Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
United States

Notices

This information was developed for products and services that are offered in the USA or elsewhere.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for IBM Documentation. Use of hyperlinks in other output formats of this information is at your own risk.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Corporation
Site Counsel
2455 South Road*

Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or

reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at ibm.com/privacy and IBM's Online Privacy Statement at ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at ibm.com/software/info/product-privacy.

Policy for unsupported hardware

Various z/OS elements, such as DFSMSdfp, JES2, JES3, and MVS™, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those

products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: [IBM Lifecycle Support for z/OS](#)
- For information about currently-supported IBM hardware, contact your IBM representative.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. For a current list of IBM trademarks, refer to the *Copyright and trademark information* at <https://www.ibm.com/legal/copytrade>.



SC31-5707-50

