

# Mainframe - Analysis messages

- [Mainframe.01](#)
- [Mainframe.02](#)
- [Mainframe.03](#)
- [Mainframe.04](#)
- [Mainframe.05](#)
- [Mainframe.06](#)
- [Mainframe.07](#)
- [Mainframe.08](#)
- [Mainframe.09](#)
- [Mainframe.10](#)
- [Mainframe.11](#)
- [Mainframe.12](#)
- [Mainframe.13](#)
- [Mainframe.14](#)
- [Mainframe.15](#)
- [Mainframe.16](#)
- [Mainframe.17](#)

## Mainframe.01

<b>Identifier</b>	Mainframe.01
<b>Message</b>	A potential recursive copybook '<COPYBOOK_NAME>' inclusion has been found.
<b>Severity</b>	Warning
<b>Explanation</b>	The COBOL analyzer detected a potential recursive copybook inclusion (COPY directive) in the source code. Meaning that a copybook A refers to a copybook B that refers to copybook A.
<b>User Action</b>	Check if the copybooks involved in the recursive reference are the expected source files. If not, then replace them by the proper files or remove the file that creates the recursive reference.

## Mainframe.02

<b>Identifier</b>	Mainframe.02
<b>Message</b>	Cannot resolve Copybook <COPYBOOK_NAME>.
<b>Severity</b>	Warning
<b>Explanation</b>	A copybook is referenced (COPY directive) in the source code but cannot be found by the analyzer. This can occur when source files have not been delivered or when the source files are not accessible by the analyzer.
<b>User Action</b>	Select the object <COPYBOOK_NAME> in the list of objects in the Unknown\Copybooks folder of the technical browser in CAST Enlighten to see where it is referenced. If this is an application specific copybook, then it is recommended to ask the application team for it. This could affect reference resolution and any diagnostic rules you might want around data components. You will find the complete list of missing source elements at the end of the analysis log.

## Mainframe.03

<b>Identifier</b>	Mainframe.03
-------------------	--------------

<b>Message</b>	Ambiguous resolution of <OBJECT_TYPE> '<OBJECT_NAME>'.
<b>Severity</b>	Warning
<b>Explanation</b>	The analyzer found a source file or an object with a name containing characters like '[', ']', '(', or ')'. These characters are removed leading to duplicate names. This message can be also generated when the analysis option "Platform: IBM z/OS" is set to true and the source code comes from another platform that allows using long names. In that case, names are truncated to 8 characters leading to duplicated names. Having duplicate names prevents to resolve references.
<b>User Action</b>	Check the source files and the analysis options.

## Mainframe.04

<b>Identifier</b>	Mainframe.04
<b>Message</b>	The picture <PICTURE> may not be analysed properly.
<b>Severity</b>	Warning
<b>Explanation</b>	The COBOL analyzer is not able to parse a variable picture correctly. This can affect quality rules related to data handling.
<b>User Action</b>	Check the source code containing the variable declaration. If the picture is correct, then please contact the <a href="#">CAST support</a> .

## Mainframe.05

<b>Identifier</b>	Mainframe.05
<b>Message</b>	Cannot resolve included file '<INCLUDE_NAME>'.
<b>Severity</b>	Warning
<b>Explanation</b>	A copybook is referenced (INCLUDE directive) in the source code but cannot be found by the analyzer. This can occur when source files have not been delivered or when the source files are not accessible by the analyzer.
<b>User Action</b>	Select the object <INCLUDE_NAME> in the list of objects in the Unknown\Copybooks folder of the technical browser in CAST Enlighten to see where it is referenced. If this is an application specific copybook, then it is recommended to ask the application team for it. This could affect reference resolution and any quality rules you might want around data components.  You will find the complete list of missing source elements at the end of the analysis log.

## Mainframe.06

<b>Identifier</b>	Mainframe.06
<b>Message</b>	Unclosed string is found.
<b>Severity</b>	Warning

<b>Explanation</b>	An alphanumeric literal has been found in the source code but the COBOL analyzer is not able to the end of the string.
<b>User Action</b>	Open the source file. Check if it contains COBOL code. If yes, then verify if the Starting Column analysis parameter is set properly. It is possible to have copybooks that do not have the same Starting column than calling programs. In this case, adapt the copybook to the expected format. If no, then remove file from the analyzed folder and exclude it from source selection. Generally, this means that source code from other technology than Cobol has been delivered in the folder. If you select source code via directories, then the analyzer is going to take into account all files belonging to this directory.

## Mainframe.07

<b>Identifier</b>	Mainframe.07
<b>Message</b>	The file '<FILE_NAME>' contains invalid '0' characters replaced by '#'.
<b>Severity</b>	Information
<b>Explanation</b>	A binary character has been find in the source code and replaced by a text character.
<b>User Action</b>	No action to perform. The binary character has been replaced automatically by an ASCII character.

## Mainframe.08

<b>Identifier</b>	Mainframe.08
<b>Message</b>	The advanced user configuration file is not found at specified location. As a consequence, JCL files could not be analyzed. Please check the analysis options:
<b>Severity</b>	Error
<b>Explanation</b>	The "AdvancedJCL.xml" file has not been found in the folder specified in the analysis settings. This prevents the analyzer to parse JCL source files.
<b>User Action</b>	Review the Mainframe technology "Platform settings" to specify the correct location of the AdvancedJCL.xml file and restart the analysis.

## Mainframe.09

<b>Identifier</b>	Mainframe.09
<b>Message</b>	Cannot resolve <OBJECT_TYPE> '<OBJECT_NAME>'.
<b>Severity</b>	Warning

<b>E x p l a n a t i o n</b>	The analyzer encountered a reference to a source code element (for example: a COBOL program, a JCL procedure, a CICS map, a CICS Transient Data item etc....) but cannot find this item in the analysis folder.
<b>U s e r A c t i o n</b>	<p>Select the object &lt;OBJECT_NAME&gt; in the list of objects in the Unknown\&lt;OBJECT_TYPE&gt; folder of the technical browser in CAST Enlighten to see where it is referenced. If this is an application specific element, then it is recommended to ask the application team for it. It can affect any quality rules.</p> <p>If the object is a program, it can be called through its PROGRAM-ID name (this is the most frequent case) which is different from its source file name. In this case you must search for it by using a GREP tool.</p> <p>If the message is about a reference to a DD name, then check if there is a JCL step calling the program and defining a DD card with the same DD name. If there is no such JCL element and if there is a supported batch part, then ask the application team for it. Note that the resolution of this type of links is not supported in case the program for which the message has been emitted is a subprogram and you can ignore that message. If the program is a main batch program and there is a JCL step that calls it, then please contact the CAST Support.</p> <p>You will find the complete list of missing source elements at the end of the analysis log.</p>

## Mainframe.10

<b>Ident ifier</b>	Mainframe.10
<b>Mes sage</b>	Potential comment is found.
<b>Seve rity</b>	Warning
<b>Expl anati on</b>	The analyzer detected a potential comment line in the COBOL source code. This can be the consequence of an improper indicator area column and the '*' character has not been found at the expected location.
<b>User Acti on</b>	Check the source code and the "Column of the Indicator area" analysis option. You can also check if the corresponding line of code comes from a copybook. In that case, change the indentation of the copybook lines to match the indicator area specified in the analysis option.

## Mainframe.11

<b>Identifi er</b>	Mainframe.11
<b>Message</b>	Reference not found for DSN clause in DD card.
<b>Severity</b>	Warning
<b>Explan ation</b>	The JCL analyzer found a DD card referring to another DD card (its name contains a prefix corresponding to another step in the same job or in an external procedure) but is not able to find that DD card.
<b>User Action</b>	Check if the JCL code containing the referred DD card has been delivered and is part of the analysis scope. It could be the case for JCL external procedures.

## Mainframe.12

<b>Identifier</b>	Mainframe.12
<b>Message</b>	Analyzing <FILE_TYPE> <OBJECT_NAME> (<FILE_NO>/<FILE_NB>).
<b>Severity</b>	Information
<b>Explana tion</b>	The Mainframe Analyzer started to analyze the file. It also displays the file number <FILE_NO> compare to the total number of files <FILE_NB> of type <FILE_TYPE> to be analyzed.

<b>User Action</b>	Nothing to do.
--------------------	----------------

## Mainframe.13

<b>Identifier</b>	Mainframe.13
<b>Message</b>	Invalid option value : <ANALYSIS_OPTION>.
<b>Severity</b>	Error
<b>Explanation</b>	An analysis option has not been set correctly. This prevents to perform the analysis.
<b>User Action</b>	Review the analysis settings and restart the analysis.

## Mainframe.14

<b>Identifier</b>	Mainframe.14
<b>Message</b>	Potential mismatch between the program and the PSB '<PSB_NAME>'. The PCB number <NUMBER> has not been found.
<b>Severity</b>	Warning
<b>Explanation</b>	The number of PCBs in PSB and in COBOL program are different. This can lead to wrong links between COBOL code and IMS PCB.
<b>User Action</b>	Check the PCB list in the PSB and the parameters specified in the PROCEDURE division and the ENTRY DLITCBL or in the LINKAGE SECTION in the COBOL program. Both must have the same number of PCBs in the same order.

## Mainframe.15

<b>Identifier</b>	Mainframe.15
<b>Message</b>	Cannot analyse <LANGUAGE_ELEMENT> for the moment.
<b>Severity</b>	Information
<b>Explanation</b>	The COBOL analyzer does not support Object Oriented version of the COBOL programming language and no objects will be created from source code.
<b>User Action</b>	Remove corresponding source files from the source files folder. Note that the list of supported COBOL version is available in the documentation.

## Mainframe.16

<b>Identifier</b>	Mainframe.16
<b>Message</b>	A potential recursive INCLUDE '<INCLUDE_NAME>' inclusion has been found.
<b>Severity</b>	Warning
<b>Explanation</b>	The COBOL analyzer detected a potential recursive copybook inclusion (INCLUDE directive) in the source code. Meaning that a copybook A refers to a copybook B that refers to copybook A.
<b>User Action</b>	Check if the copybooks involved in the recursive reference are the expected source files. If not, then replace them by the proper files or remove the file that creates the recursive reference.

## Mainframe.17

<b>Identifier</b>	Mainframe.17
-------------------	--------------

<b>Message</b>	Ambiguous resolution for copybook '<INCLUDE_NAME>' in the folders.
<b>Severity</b>	Warning
<b>Explanation</b>	The COBOL analyzer detected more than two copybooks with the same name when expanding. The first copybook will be selected.
<b>User Action</b>	Check why there are multiple copybooks with the same name in the project and verify if the selected copybook is correct in the Cobol file.