Mainframe - Analysis results

On this page:

- What results can you expect?
  - Objects
    - Cobol
    - JCL
    - IMS
    - CICS
  - Links
    - Cobol
    - JCL
    - IMS
    - CICS
- Miscellaneous Cobol information
- Miscellaneous JCL Information
- Display in CAST Enlighten

Summary: This document provides information about the analysis results you can expect from a Mainframe analysis.

What results can you expect?

Once the analysis/snapshot generation has completed, you can view the results in the normal manner (for example via CAST Enlighten):

Objects

Cobol

	Program
	Copy Book
	Paragraph
	Section
	Division
	File Link (Sequential, Partitioned, VSAM)
	Data Link
	Class
	Directory/Root Directory
	Entry Point
	Interface
	Program Specification Block
	Database Definition
	IMS Segment
	Cobol conditional test

	Cobol data
	Cobol literal, Cobol structure data, Cobol constant data
	External File
	Transaction
	Screen Map
	Transient Data
	Project
	Cobol Subset

JCL

	Catalogued Job
	Catalogued Procedure
	Step
	Instream Procedure
	Dataset (Sequential, Partitioned, VSAM)
	External Program
	Included File
	Index
	Project
	Directory/Root Directory
	Program Specification Block
	JCL Subset
	JCL to Java
	JCL SQL Query

IMS

	Directory/Root Directory
	IMS Database
	IMS Field
	IMS Alternate PC Block
	IMS PC Block type Database
	IMS PC Block type GSAM

	IMS Project
	IMS PS Block
	IMS Segment
	IMS GSAM File
	IMS Subset
	IMS Transaction File
	IMS Transaction
	IMS Message Format Service
	IMS Message Input Descriptor
	IMS Message Output Descriptor

CICS

	CICS Basic Mapping Support
	CICS System Definition file
	CICS System Definition group
	CICS Dataset
	Directory/Root Directory
	CICS Map
	CICS Mapset
	CICS Project
	CICS TD Queue
	CICS Transaction
	CICS TS Model

Links

Cobol

Link Type		Linked Objects		Code Example
		Calling	Called	
CALL	PROG*	Program, section or copybook	Program or entrypoint	CALL "CC2DISPLAY"
	TRANSAC*			EXEC CICS XCTL PROGRAM(TEST) END EXEC

	PERFORM	Program, section or sub-object	Section or paragraph	MODULE-ENTREE SECTION. * * LA LIGNE CI DESSOUS DOIT ETRE DELETEE APRES LES TESTS * MOVE '*** EXECUTION STARTEE NORMALEMENT ***' TO MESS. DISPLAY MESS. MOVE SPACE TO MESS. * **PERFORM LECTURE-PARAM**. * FIN-MODULE-ENTREE. EXIT. EJECT * COPY SYBPINIT.
	GO TO	Program	First executed section or paragraph	*************** LECTURE-PARAM SECTION. *************** * READ MEF001. IF FILE-STATUS NOT = '00' DISPLAY 'CODE RETOUR' FILE-STATUS DISPLAY 'CARTE PARAMETRE INEXISTANTE : ' CODE-ABEND **GO TO SORTIE-ERREUR** END-IF. *
		Section	First executed paragraph in section	
		Paragraph	Next executed paragraph in same section	
	-	Section /Paragraph	Section/Paragraph	When Sections or Paragraphs are sequentially executed (one after the other) a Call link will be generated. For example, if a program is made of Section A then Section B then Paragraph C then Section D, the links will be resolved as A -> B -> C -> D.
INCLUDE		Program or Copybook	Copybook	* -ACDA * MNPW * FD DU FICHIER INF103 PAR LE CHEMIN INF103 * BLOCK CONTAINS 0. * **COPY INF103 REPLACING INF103 BY ING103.**
USE		Calls to matched character strings		Display "a **string**"
		• Cobol Program • Cobol Section • Cobol Paragraph	Program Specification Block	EXEC DLI SCH PSB (**MYPSB**) END-EXEC
ACCESS	**OPEN** **CLOSE**	• Cobol Program or sub-object	• Cobol File Link • Cobol Data Link	*========== * OUVERTURE DU FICHIER IAHCDECE *========== OPEN-IAHCDECE. **OPEN INPUT IAHCDECE**. *========== * LECTURES DU FICHIER IAHCDECE *========== LECT-IAHCDECE. **READ IAHCDECE**. *========== * FERMETURE DU FICHIER IAHCDECE *========== CLOSE-IAHCDECE. **CLOSE IAHCDECE**.
	READ **WRITE**	• Cobol Program or sub-object • Cobol Section • Cobol Paragraph	• Cobol File Link • Cobol Data Link • Constant Data • Structured Data • Data • Conditional Test • Program Communication Block (IMS PCB) • IMS Segment	*========== * LECTURE-SEGMENT-CISRAPMP *========== EXEC DLI GU USING PCB (3) SEGMENT (**CISRAPMP**) WHERE (CIOEAX = IN-PREM-H) INTO (WS-DUMMY-AREA) END-EXEC. *========== COMMANDE-CALL-LEVEL *========== CALL 'CBLTDLI' USING GU **CICSDLI-PCB** CICSDLI-SEGM CICSDLI-QSSA

For **Embedded SQL links**, the following are valid for **all** servers.

USE	**SELECT**	This type is reserved for server side object referencing	EXEC SQL... SELECT
	UPDATE		EXEC SQL... UPDATE
	INSERT		EXEC SQL... INSERT

	DELETE		EXEC SQL... DELETE
CALL			EXEC SQL... CAL

For link types CALL PROG and CALL TRANSAC, two limitations exist when the call is in "string" form:

- If the string is constant and declared in the "data-division" section, the entry point will be resolved in the normal way.
- If the string is dynamic, the program may be found by the **Dynamic Link Manager**.

In addition, the following **Embedded SQL links** are valid for **DB2 only**:

DEPEND ON		This type is reserved for server side object referencing on structured or distinct UDTs.	-
DDL	**CREATE**	This type is reserved for server side object referencing on Tables	-
DDL	**DROP**		

JCL

Type		**Linked Objects**	
		Calling	**Called**
ACCESS	**WRITE**	JCL Step	JCL Data Set
	READ		
	EXECUTE		
PROTOTYPE		Cobol File Link	JCL Data Set
		Cobol Data Link	JCL Data Set
		JCL Data Set	Cobol JCL Program
CALL		JCL Step	Cobol JCL Program
		JCL Job	JCL Step
		JCL Procedure	JCL Step
		JCL Step	JCL Procedure
USE		JCL Step	IMS DBD
MONITOR	**AFTER**	JCL Step	JCL Step

IMS

Type		**Linked Objects**	
		Calling	**Called**
ACCESS	**WRITE**	IMS PC Block	IMS Segment
USE		IMS PC Block	IMS DBD or IMS GSAM File

CICS

Type		**Linked Objects**	
		Calling	**Called**
CALL	**TRANSAC**	CICS Transid	Client/Cobol Program

For **Transactional Code**, the following are valid:

Type		**Linked Objects**		**When does this link occur?**
		Calling	**Called**	
CALL	**TRANSAC**	Client/Cobol Program or its Sub object	Client/Cobol Program	EXEC CICS XCTL PROGRAM(TEST) END EXEC or EXEC CICS LINK PROGRAM(TEST) END EXEC

CALL	**TRANSAC**	Client/Cobol Program or its Sub object	CICS Transaction	EXEC CICS START TRANSID(TRANSID) END EXEC or EXEC CICS RETURN TRANSID (W-XFR-CTRS) COMMAREA (W-CIC-CMA-LDON) LENGTH (W-CIC-CMA-QLENDON) END-EXEC
MONITOR		Client/Cobol Program or its Sub object	CICS Map	EXEC CICS SEND MAP (W-CIC-CMAPCUR) MAPSET (W-CIC-CMPSCUR) FROM (MGAB10O) ERASE FREEKB FRSET CURSOR END-EXEC or EXEC CICS RECEIVE MAP (W-CIC-CMAPCUR) MAPSET (W-CIC-CMPSCUR) FROM (MGAB10O) END-EXEC
ACCESS	**OPEN** **CLOSE** **READ** **WRITE**	Client/Cobol Program or its Sub object	CICS Dataset	ENDBR, DELETE, LINK, READ, READNEXT, READPREV, REWRITE, STARTBR, XCTL, WRITE ex: EXEC CICS READ DATASET ('C1MASTR') INTO (BLKBLK(BLKBLK-OCUR)) RIDFLD (CICS-RRN) RRN RESP (WS-RESP) END-EXEC.
	READ **WRITE**	Client/Cobol Program or its Sub object	CICS Transient Data	DELETEQ, READQ, WRITEQ ex: EXEC CICS DELETEQ TD QUEUE (W-CIC-TSQ-LNOM)

Miscellaneous Cobol information

Rules for resolving Paragraph names in a Section

The Mainframe Analyzer (Cobol) uses the following rules when resolving **Paragraph names** defined in **Sections**:

1. If the referenced Paragraph **is located** in the current Section, Cobol Analyzer will link the calling Paragraph to the called Paragraph
2. If the referenced Paragraph **is not located** in the current section, Cobol Analyzer will issue a syntax error
3. If the referenced Paragraph has a unique declaration **outside the Section**, Cobol Analyzer will create a link to this Paragraph.
4. The following syntax is also resolved: "Paragraph Name IN/OF Section Name". A link will be created to the **correct Paragraph** (which will be outside the current Section).

Access type links

Access type links are created when your Cobol program calls an **external file** (Data Set).

1. The instructions that manage classic files ('File Link' type) are:
   Instructions for opening a file: OPEN Reading data: READ Writing data: WRITE, REWRITE Closing the file: CLOSE
2. Sorting operations on data set files (Data Link type) are carried out via the instructions MERGE and/or SORT. These instructions should generate CALL + PERFORM type links on paragraphs or sections. OPEN and CLOSE instructions are also used.
3. Access (Read/Write) links to "Cobol Constants" and "Cobol Data":
   - Variables used in the flow control: IF, PERFORM, EVALUATE, SEARCH.
   - Inclusion of the instructions: MOVE, INITIALIZE, SET, CALL, OPEN, CLOSE, READ, WRITE, REWRITE, MERGE, SORT.
   - The following instructions are not yet included: MULTIPLY, SUBTRACT, ADD, DIVIDE, EXHIBIT.
4. CICS files read via CICS instructions in an EXEC CICS ... END-EXEC
   - Opening of file - Instructions: STARTBR -- Acesse(Open) on the file
   - Data reading - Instructions: DELETE, WRITE, REWRITE -- Access(Write)
   - Closing of file - Instruction: ENDBR -- Access(Close)
5. Access to the segment in the IMS databases:
   - Links to the segment (Access + Write/Read)
   - Links to the Program Specification Block

File Selection

Logical Files are declared in the **"File Section"** part of the program with the FD or SD tag.

- FD: declaration of 'file link' type
- SD: declaration of 'data link' type

Example declaration of a logical file called MAIL-XREF (type FILE-LINK) in the Cobol program:

```
004600 FILE SECTION.                                                    00015900
004700                                                                                00016000
009000 FD MAIL-XREF                                                   00020700
009100 LABEL RECORDS ARE STANDARD                                  00020800
009200 BLOCK CONTAINS 0 RECORDS.                                   00020900
009300 01 MAIL-XREF-REC.                                                  00021000
009400 03 MAIL-XREF-KEY.                                                  00021100
009500 07 MAIL-XREF-ADDRESS.                                            00021200
009600 11 MAIL-XREF-ZIP-PRE PIC X(05).                              00021300
009700 11 MAIL-XREF-ZIP-SUF PIC X(04).                              00021400
009800 07 MAIL-XREF-STATE PIC X(03).                                00021500
009900 07 MAIL-XREF-CITY PIC X(25).                                 00021600
010000 07 MAIL-XREF-POSTAL-CODE PIC X(02).                         00021700
010100 03 MAIL-XREF-DATA PIC X(324).                                  00021800
```

Replacement used in COPY REPLACING directives

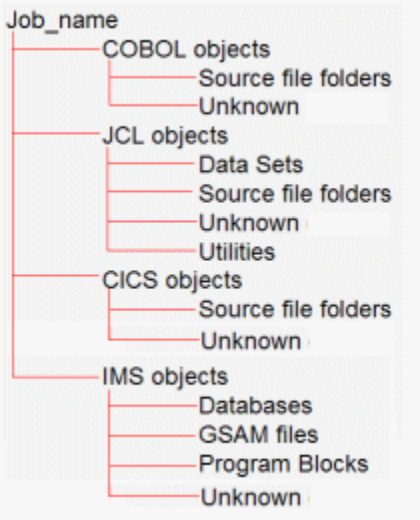
- Replacements can be applied to words or parts of words
- Patterns used to replace parts of words must be delimited by the following characters: :, (,), \ or "
- Patterns that are not delimited by the above characters are considered as being used to replace entire words
- LEADING and TRAILING clauses mean that the replacement will be applied on parts of words and as such, patterns must respect rule two (first character and last character will be removed from the pattern).

Miscellaneous JCL Information

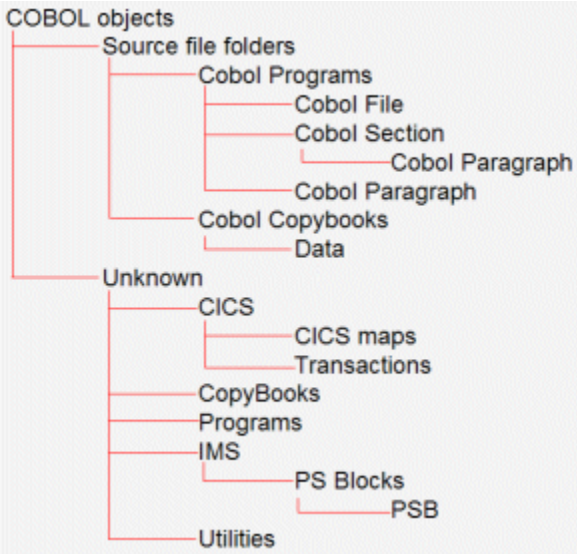
JCL is a **command language** used to execute programs on large systems - primarily Cobol oriented. CAST's Mainframe Analyzer (JCL) is targeted at **IBM's JCL language for 3090 systems** used in conjunction with **Cobol projects**.

Display in CAST Enlighten

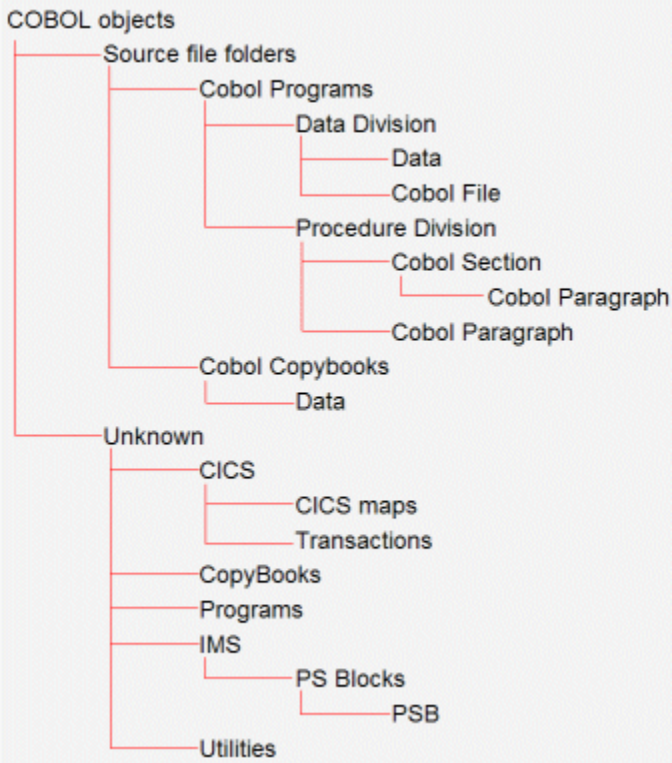
In CAST Enlighten, a full Mainframe Analysis (i.e. including COBOL, JCL, CICS and IMS) may be represented as follows:



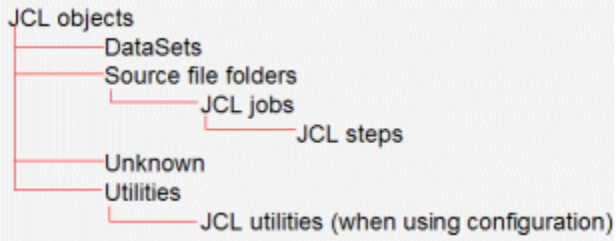
COBOL Objects branch - if the data has NOT been saved in the Analysis Service (**Data Structures > Save Data Only** option in the Mainframe Technology options is not activated). Note that the **Cobol Copybooks** and its sub-heading **Data** in the **Source file folders** heading will only be visible if the **Save data found in copy books** option in the Mainframe Technology options is activated):



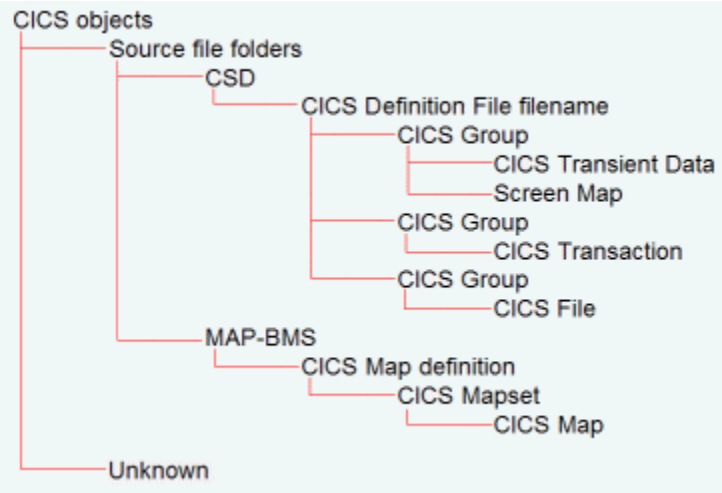
COBOL Objects branch - if the data has been saved in the Analysis Service (**Data Structures > Save Data Only** option in the Mainframe Technology options is activated). Note that the **Cobol Copybooks** and its sub-heading **Data** in the **Source file folders** heading will only be visible if the **Save data found in copy books** option in the Mainframe Technology options is activated):



JCL Objects branch



CICS Objects branch



IMS Objects branch