



# **Data Structures**

# Declarations, Dependencies and Usages: Now everything at a Glance

AMELIO Logic Discovery provides extensive data structure analyses for COBOL, PL/I and Delta ADS applications. With the new presentation of the analysis results, the knowledge about your data structures, their declaration, usage and dependencies is now available at a glance.

#### All about your data structures

To understand an application, it is necessary to understand the contained data structures, their declaration, provenance, usage and dependencies. To obtain this knowledge, an in-depth analysis of all involved source code modules is essential. For that reason AMELIO Logic Discovery provides extensive data structure analyses for COBOL, PL/I and ADS applications. They answer questions like

#### **Declaration**

- How is the data structure declared? What is the type and size of it?
- Where has the data structure been declared? Was it declared in the program itself or in a copybook, include module or macro?

Source		÷	Û				
	es definitions		****	^			
01 LOGMSG 05 FI	LER PIC X(6) V MSG-TEXT PIC X(50). ERR REDEFINES LOGMSG	0) VALUE "BUY E	RR =	»". <sub>V</sub>			
•C Dat	a Definition/Usage					-	Ē
Name		Туре	Size	Exp. Line	Copybook	Orig. Line	
~ 0 0	1 TPSTATUS-REC		264	78		78	
	05 APPL-RETURN-CODE	inative binary (9)	4	80	RETCODES	1	
	05 APPL-RETURN-CODE 05 APPL-SUB-CODE		4		RETCODES		
			4	81		2	
~00	05 APPL-SUB-CODE	native binary (9)	4	81 82	RETCODES	2	
~ O 0	05 APPL-SUB-CODE 05 APPL-MSG-CODE	native binary (9)	4 256	81 82 87	RETCODES	2	
~00	05 APPL-SUB-CODE 05 APPL-MSG-CODE 1 LOGMSG	native binary (9) character (256)	4 256 56	81 82 87 88	RETCODES	2 3 83	

... and specifically for PL/I

- Which type and size have implicitly declared structures or structures whose declarations have been abbreviated by using factoring?
- If a structure is declared multiple times, which declaration is valid in which context?

🛱 Data Definition Usage					
Name	Туре	Declared in	Line	^	
🗸 🔾 1 A1			53		
2 A	[3, 2] binary fixed (15)	DO_USERLOG	54	10	
2 B	[3, 2] binary fixed (15)	DO_USERLOG	56		
2 C	[3, 2] binary fixed (15)	DO_USERLOG	69		
2 D	[3, 2] binary fixed (15)	DO_USERLOG	84		
✓ ○ 1 INFOLOGMSG		DO_USERLOG	331		
5 *	character (6)	DO_USERLOG	332		
5 LOGMSG_TEXT	character (40)	DO_USERLOG	333		
5 *	character (34)	DO_USERLOG	334		
LOGMSG_TEXT	character (40)	READ_FILE	418		
if 📄 Source	e				- E
decla decla decla decla A	re 1 A1, 2(A,B,C,D are 1 A2, 2(B,C,D)	<pre>, F character ) (3,2) binar binary fixed</pre>	(10) y fix (15)	init ed (: ;	<pre>ial ((10)'A')) static;</pre>

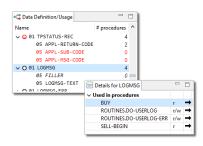
#### Usage

- In which paragraphs and procedures the data structure is accessed?
- Is the structure read-only or also modified?
- Is an element or a structure used at all?





• Is the complete structure used or only single elements from it?



## Dependencies

- What are the relationships between the structures?
- Is the structure redefined?

📲 Data Definition/Usage		-				
Name	Туре	# redefined	신		- F	7
✓ O 01 LOGMSG		1	-	Details for LOGMSG	- 1	-
05 FILLER	character (6)	0	~	Redefined by		
05 LOGMSG-TEXT	character (50)	0		LOGMSG-ERR		T
> O 01 LOGMSG-ERR	redefines LOGMSG	0	in a			
01 LOGMSG-LEN	native binary (9)	0	$\mathbf{v}$			

• Is there a pointer to the structure?

## ... and now everything at a glance

To understand data structures it is necessary to know its declarations as well as its usage and dependencies. That's why we have summarised both perspectives "Data Definition" and "Data Usage" in one comprehensive perspective.

🚟 AMELIO Logic Discovery - BUY												
File Options Help												
📮 🔗 🔟 🖄 🔍 🦕 🗢 🔍	9,											
🗄 Report Explorer 👘 🗖	Data Definition/Usage											
✓ im Structure Analysis	Name	Т	vpe			Size	Exp. Line	Copybook	Orig. Line	# procedures	# redefined	^
📜 Program Tree	V O 01 TPSTATUS-REC					264	. 78		78	. 4	0	
10: Physical Structure	05 APPL-RETURN-CODE		ative bir	ary (9)		4	80	RETCODES	1	2	0	
18: Logical Structure	05 APPL-SUB-CODE	n	ative bir	ary (9)		- 4	81	RETCODES	2	0	0	
E Procedure Graph	05 APPL-MSG-CODE	c	haracter	(256)		256	82	RETCODES	3	0	0	
🗸 📄 Usage Analysis	V O 01 LOGMSG					56	87		83	4	1	
👻 Include Analysis	05 FILLER	c	haracter	(6)		6	88		84	0	0	
Data Definition/Usage	05 LOGMSG-TEXT		haracter			50	89		85	6	0	
	> O 01 LOGMSG-ERR	D	edefines	LOGM	ISG	56	90		86	2	0	~
File Access	Details for LOGMSG			🗈 s	iource							
Database Access	✓ Redefined by			1 .	Log	mes	sages de	finitions				^
🗸 💼 Pattern Analysis	LOGMSG-ERR		t	I. ·	A1	LOGM	*******	• • • • • • • • • • • •	*******	• • • • • • • • • • • • • • • •	*****	
🔛 Patterns	✓ Used in procedures						SG. FTLLER	PTC X	(6) VALUE	"BUY =>".		
✓ im Reports	BUY	r	<b>→</b>				LOGMSG-T	EXT PIC X	(50).			
C Procedure Report	ROUTINES.DO-USERLOG	r/w	<b>→</b>				SG-ERR R FTLLER	EDEFINES L				
Metrics		r/w	+					ROUTINE PI	C X(10) V C X(10).	ALUE "BUY E	RR =>".	
	SELL-BEGIN	r -	<b>→</b>					DT	C.X(18).N	ATTE " EATTE	D+ STATUS	- • ×

In this way the entire knowledge about your data structures is available at a glance.

# The new AMELIO Logic Discovery version: Available now!

The display of the analysis results is now even more clearly. In this way you understand your data structures more quickly. See for yourself!

You don't use AMELIO Logic Discovery yet? We would be glad to show you the possibilities of the tool, preferably with your own sources.



Copyright © 2017 Delta Software Technology GmbH. All rights reserved.

Order number: MT21084.01 - July 2017

Delta, SCORE, ObjectBridge, SCOUT<sup>2</sup>, AMELIO, HyperSenses and the logo of Delta Software Technology are registered trademarks and SCORE Adaptive Bridges, SCORE Data Architecture Integration, Model Driven Legacy Integration, Integration in Motion, SCORE Transformation Factory, AMELIO Modernization Platform, AMELIO Logic Discovery, ADS, ANGIE and Active Intent are trademarks of Delta Software Technology GmbH in Germany and/or other countries. All other registered trademarks, trademarks, trade names or service marks are the property of their respective owners.