

# ***What's New in Enterprise COBOL V4***

Tom Ross

Stephen Miller

GSE session S14

May 28, 2008

# ***Enterprise COBOL V4 overview***

- Performance improvements
- XML enhancements
- DB2 9 support
- Usability enhancements
- Debugging enhancements

## ***Performance enhancements***

- Compiler generates newer z/Architecture instructions to improve performance of COBOL programs
- Significantly improve performance of COBOL Unicode
  - z/Architecture instructions MVCLU, CLCLU
  - inline instructions instead of COBOL library calls where possible
    - Unicode MOVEs, comparisons
    - String operations STRING, UNSTRING, INSPECT...
  - tuning of Unicode conversions and COBOL interface to z/OS Unicode Services

# *Enterprise COBOL V4 overview*

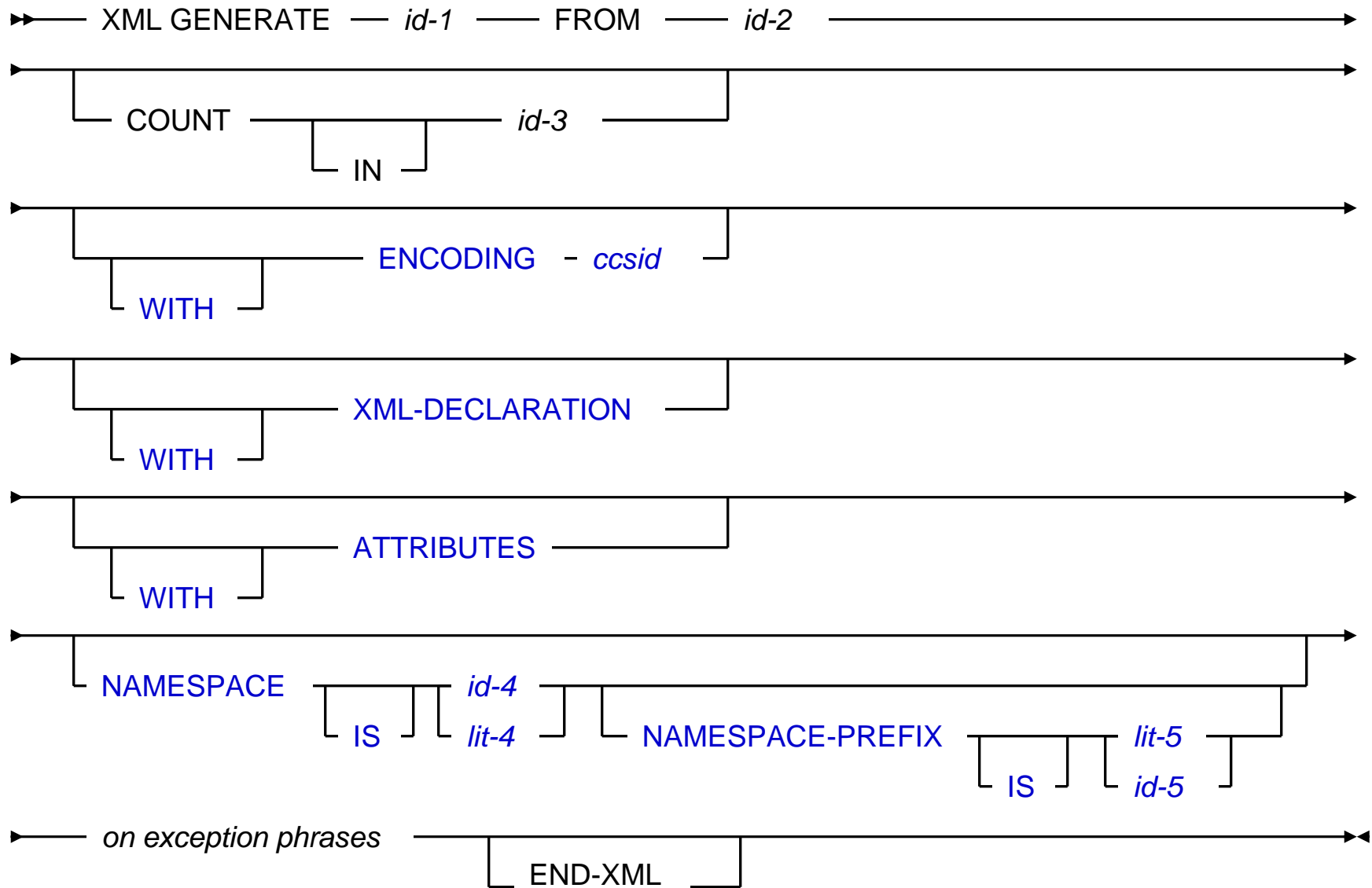
- Performance improvements
- **XML enhancements**
- DB2 9 support
- Usability enhancements
- Debugging enhancement

# *XML generation enhancements*

More control of output XML document:

- XML documents now can be encoded in UTF-8, as well as UTF-16 Unicode or various EBCDIC codepages
- Optional XML declaration  
`<?xml version="1.0" encoding="UTF-16"?>`
- Optional namespace support
- Option for XML element attributes

# XML GENERATE syntax



# ***XML GENERATE requesting XML declaration and an explicit encoding***

Data declaration:

01 Greeting.

05 msg pic x(80) value 'Hello, world!'.

Procedural code:

XML Generate Doc from Greeting

encoding 1208

with XML-declaration

End-XML

Display Doc

Result:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<Greeting><msg>Hello, world!</msg></Greeting>
```

## ***XML GENERATE with attributes***

Data declaration:

01 Greeting.

05 msg pic x(80) value 'Hello, world!'.

Procedural code:

XML Generate Doc from Greeting with attributes

Display Doc

Result:

<Greeting msg="Hello, world!"></Greeting>

## ***XML GENERATE with attributes ...***

Data declaration:

01 G.

05 A pic x(3) value "aaa".

05 B.

10 C pic x(3) value "ccc".

10 D pic x(3) value "ddd".

05 E pic x(3) value "eee".

Procedural code:

XML Generate Doc from G with attributes

Display Doc

Result:

```
<G A="aaa" E="eee"><B C="ccc" D="ddd"></B></G>
```

## ***XML GENERATE with default namespace***

Data declaration:

01 Greeting.

05 msg pic x(80) value 'Hello, world!'.

Procedural code:

XML Generate Doc from Greeting  
namespace is "http://example"

Display Doc

Result:

```
<Greeting xmlns="http://example"><msg>Hello,  
world!</msg></Greeting>
```

## ***XML GENERATE with explicit namespace and prefixed elements***

Data declaration:

01 Greeting.

05 msg pic x(80) value 'Hello, world!'.

Procedural code:

XML Generate Doc from Greeting  
namespace is "http://example"  
namespace-prefix is "pfx"

Display Doc

Result:

```
<pfx:Greeting xmlns:pfx="http://example"><pfx:msg>Hello, world!  
</pfx:msg></pfx:Greeting>
```

## ***XML PARSE enhancements***

- z/OS XML System Services parser:
  - a z/OS system component
  - high-speed non-validating XML parsing
  - available in z/OS 1.7 or later
  - **new underlying parser for XML PARSE statement**
- New compiler option: XMLPARSE(COMPAT | XMLSS)
  - **COMPAT**: use existing XML parser (built in to COBOL library)
  - **XMLSS**: use new z/OS XML System Services parser

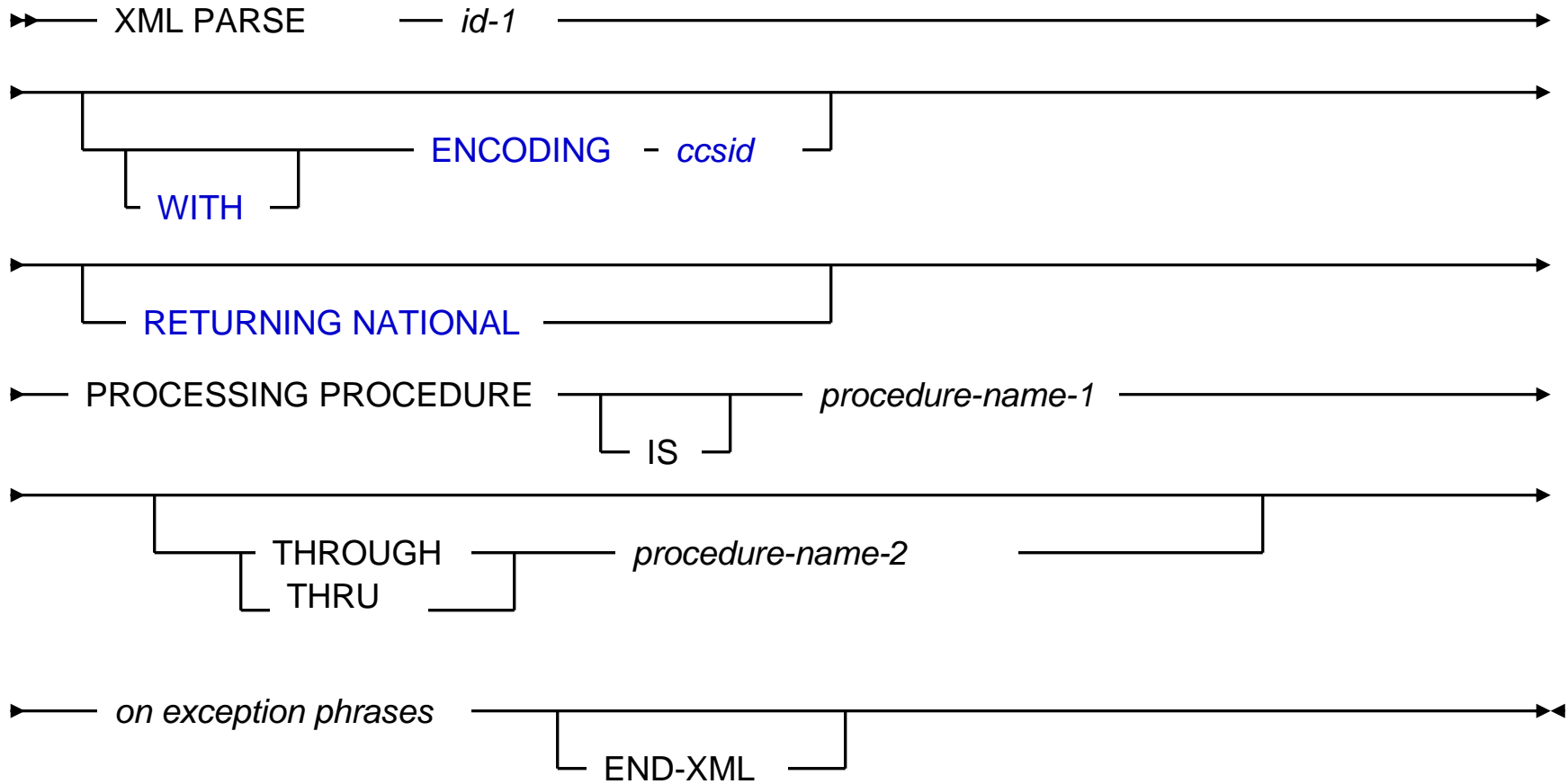
## ***XML PARSE enhancements***

- New XML System Services parser gives COBOL:
  - latest IBM parsing technology for XML PARSE statement
  - offload of COBOL XML parsing to zAAP specialty processor
  - XML namespace support
  - UTF-8 XML documents
  - support for very large XML documents
    - process a buffer of XML at a time
    - new XML-Event value END-OF-INPUT signals  
"provide next buffer of XML"
- Follow-on release of XML System Services will enable optional XML validation  
*(IBM Statement of Direction Announcement April 18, 2007)*

# ***XML PARSE enhancements***

- Additional syntax for XML PARSE statements:
  - WITH ENCODING *ccsid*
    - specify encoding of input XML document
    - may be 1208, for UTF-8 Unicode, or other codepages
  - RETURNING NATIONAL
    - return document fragments in Unicode UTF-16
- New XML-Event values
- New special registers for namespace processing:
  - XML-NAMESPACE
  - XML-NAMESPACE-PREFIX
  - XML-NNAMESPACE
  - XML-NNAMESPACE-PREFIX

# XML PARSE syntax



# XML PARSE with namespace support

XML document in data item Doc:

```
<pfx:Greeting xmlns:pfx="http://example"><pfx:msg type="brief">  
  Hello, world!</pfx:msg></pfx:Greeting>
```

COBOL program logic:

XML Parse Doc processing procedure P

...

P. Display XML-event XML-text XML-namespace-prefix XML-namespace.

XML-Event	XML-Text	XML-namespace-prefix	XML-namespace
START-OF-DOCUMENT			
START-OF-ELEMENT	Greeting	pfx	http://example
NAMESPACE-DECLARATION		pfx	http://example
START-OF-ELEMENT	msg	pfx	http://example
ATTRIBUTE-NAME	type		
ATTRIBUTE-CHARACTERS	brief		
CONTENT-CHARACTERS	Hello, world!		
END-OF-ELEMENT	msg	pfx	http://example
END-OF-ELEMENT	Greeting	pfx	http://example
END-OF-DOCUMENT			

# ***Parsing very large XML documents from a file***

Select Input-XML Assign to infile File status is Input-XML-status.

## **FD Input-XML**

Record is varying from 1 to 255 depending on Rec-length recording mode is V.  
1 fdrec.

2 pic X occurs 1 to 255 depending on Rec-length.

Procedure division.

Open input Input-XML

**Read Input-XML**

**XML parse fdrec** processing procedure Handle-parse-events

Close Input-XML

Stop Run.

Handle-parse-events.

Evaluate XML-event

When 'END-OF-INPUT'

**Read Input-XML**

Evaluate Input-XML-status

When 0

Move 1 to XML-code

Display 'Continuing with: ' fdrec

When 10

Display 'At EOF; no more input.'

When other

Display 'Read failed, file status:' Input-XML-status

End-evaluate

When ...

End-evaluate

# Processing UTF-8 XML documents with COBOL

Recommendations:

- Create UTF-8 XML documents using  
`XML GENERATE document FROM group  
WITH ENCODING 1208`
- Parse UTF-8 XML documents using  
`XML PARSE document  
WITH ENCODING 1208  
RETURNING NATIONAL`
- Process the XML as NATIONAL data (UTF-16 Unicode) rather than directly in UTF-8
  - Support for NATIONAL (UTF-16 Unicode) is built into COBOL
  - UTF-16 characters are fixed width (in general)
  - Variable width UTF-8 characters are unnatural for COBOL

## ***XML PARSE – caveats with new parser***

- Existing XML PARSE syntax is supported by new parser, but there are some operational differences:
  - XML exception codes based on XML System Services return-code / reason-code
  - new events
  - XML content may be subdivided and delivered differently
    - multiple events may be combined into one, for example:
      - Baker&apos;s dozen
      - delivered on 1 event (not 3) as
      - Baker's dozen
- IBM has attempted to minimize differences
  - but some changes are unavoidable ...
  - XMLPARSE(COMPAT) option provides parsing compatible with Enterprise COBOL V3
- Migration path documented in COBOL Migration Guide
  - XMLPARSE(COMPAT) -> XMLPARSE(XMLSS)

# *Enterprise COBOL V4 overview*

- Performance improvements
- XML enhancements
- **DB2 9 support**
- Usability enhancements
- Debugging enhancement

# *DB2 9 support in COBOL coprocessor*

- Enables new SQL function for COBOL coprocessor users
  - New data types:
    - new XML types
    - BINARY, VARBINARY    <- Use these instead of FOR BIT DATA clause
    - BIGINT
    - file reference variables
  - New SQL syntax:
    - XML manipulation
    - Large object manipulation enhancements
    - MERGE, SELECT FROM MERGE, ...
  - More complete support for DB2 SQL processing options
    - STDSQL, NOFOR, ...
- Coprocessor usability improvement:
  - COBOL listing includes list of DB2 options in effect
- Also available for Enterprise COBOL V3R4
  - delivered via service: [APAR PK09731](#)

# *Enterprise COBOL V4 overview*

- Performance improvements
- XML enhancements
- DB2 9 support
- **Usability enhancements**
- Debugging enhancement

## ***Usability enhancements***

- COBOL compiler options in a data set
- Cross reference of COPY statements, libraries, and datasets in compiler listing
- DB2 options in the compiler listing
  - *requires DB2 9 coprocessor*
- SQLCA and SQLDA expansion in compiler listing
  - *available with either DB2 V8 or DB2 9 coprocessors*
- Compile of very large programs

## ***OPTFILE compiler option***

- COBOL compiler options in a data set
- Avoids problems with:
  - 100 character limit for JCL PARM string options
  - Long strings of DB2 or CICS suboptions for coprocessors
- Specify OPTFILE as compiler invocation option or on PROCESS/CBL card
- Compiler opens and reads options from data set identified by DDNAME SYSOPTF
  - RECFM F or FB, LRECL 80
  - Asterisk in column 1 indicates comment
  - Free format options in columns 2 - 72
  - Sequence numbers in columns 73 - 80

# ***SYSOPTF options shown in compiler listing***

PP 5655-S71 IBM Enterprise COBOL for z/OS 4.1.0      Date 10/08/2007    Time 15:07:38    Page    1

Invocation parameters:

    SIZE(4000K) APOST LC(0) NOSEQ NONUM OPTFILE

PROCESS(CBL) statements:

    CBL ARITH(EXTEND),TRUNC(OPT)

Options from SYSOPTF:

    SSRANGE

    ZWB

    OPTIMIZE

    TEST(NOHOOK)

Options in effect:

    NOADATA

        ADV

        APOST

        ARITH(EXTEND)

    NOAWO

        BUFSIZE(4096)

    NOCICS

        CODEPAGE(1140)

    NOCOMPILE(S)

    NOCURRENCY

        DATA(31)

    NODATEPROC

        DBCS

    NODECK

    NODIAGTRUNC

    NODLL

    NODUMP

    NODYNAM

    NOEXIT

    NOEXPORTALL

    NOFASTSRT

        FLAG(I,I)

    ...

# COBOL listing: COPY statement cross reference

PP 5655-S71 IBM Enterprise COBOL for z/OS 4.1.0 DEMOXREF Date 10/08/2007 Time 17:03:29 Page 15

COPY/BASIS cross reference of text-names, library names and dataset information

Text-name (Member)	Library (DDNAME)	File name (Dataset name)	Concat Level	ISPF statistics	
				Created	Changed
ACTIONS	SYSLIB	USER1.COBOL.COPY	0	1992/07/11	1993/03/16 16:16:17
CUSTOMER	SYSLIB	USER1.COBOL.LIB2PDSE	1	2007/06/07	2007/06/07 11:28:14
CUSTOMER	ALTDDXXY	USER1.COBOL.LIB3	0	2007/06/01	2007/06/01 17:35:18
HOUSE	SYSLIB	USER1.COBOL.LIB2PDSE	1		
HOUSE	ALTDDXXY	USER1.COBOL.LIB2	1	2007/06/07	2007/06/07 11:39:02
IMOTOR	SYSLIB	USER1.COBOL.LIB4X	3	2007/06/07	2007/06/07 11:37:53
ISOVERFY	SYSLIB	USER1.COBOL.COPY	0		
NSMAP	SYSLIB	USER1.COBOL.LIB3	2		

# COBOL listing: HFS COPY statement cross reference

COPY/BASIS cross reference of text-names, library names and file names

Text-name	Library-name	File name
'copyA.cpy'	SYSLIB (default)	./copyA.cpy
'/copydir/copyM.cbl'	SYSLIB	/u/USER1/cobol//copydir/copyM.cbl
'/copyA.cpy'	SYSLIB	/u/USER1/cobol//copyA.cpy
'cobol/copyA.cpy'	ALTDD2	/u/USER1/cobol/copyA.cpy
'copy/stuff.cpy'	ALTDD2	/u/USER1/copy/stuff.cpy
'copydir/copyM.cbl'	SYSLIB	/u/USER1/cobol/copydir/copyM.cbl
'copydir/copyM.cbl'	SYSLIB (default)	./copydir/copyM.cbl
'stuff.cpy'	ALTDD	/u/USER1/copy/stuff.cpy
'reallyxxreallyyyreallyzzlongxxlongyylo>	SYSLIB (default)	./reallyxxreallyyyreallyzzlongxxlongyylongzzname.cpy

./ = /u/USER1/cobol

Note: Some names were truncated. > = truncated on right < = truncated on left

# COBOL listing: DB2 9 options in COBOL listing

PP 5655-S71 IBM Enterprise COBOL for z/OS 4.1.0 Date Date 10/08/2007 Time 15:04:08 Page 1

Invocation parameters:

SIZE(3048K),FLAG(I,I),LIB

PROCESS(CBL) statements:

CBL SQL,ARITH(EXTEND),XREF

Options in effect:

NOADATA

ADV

QUOTE

ARITH(EXTEND)

...

SQL

SQLCCSID

...

ZWB

SQL Options in effect:

ATTACH(TSO)

CCSID(1140)

CONNECT(2)

DEC(15)

NEWFUN(YES)

ONEPASS

PERIOD

QUOTESQL

STDSQL(NO)

SQL(DB2)

NO XREF

NO SOURCE

DSNHDECP LOADED FROM - (DSN910.SDSNLOAD(DSNHDECP))

## *Compile of very large programs*

- Internal compiler tables expanded
- Enable compile of very large COBOL source programs
  - large programs typically from COBOL generators
  - origin may have been from program generator, hand coded enhancements subsequently
  - we have seen programs over 400,000 source lines long!
- Avoid compiler abort 5092 ...

LineID Message code Message text

228012 IGYAS5125-U The compilation was terminated due  
to a compiler error in phase id: "IGYCASM1".

228012 IGYAS5092-U Overflow occurred on table "TRQB".

# *Enterprise COBOL V4 overview*

- Performance improvements
- XML enhancements
- DB2 9 support
- Usability enhancements
- **Debugging enhancement**

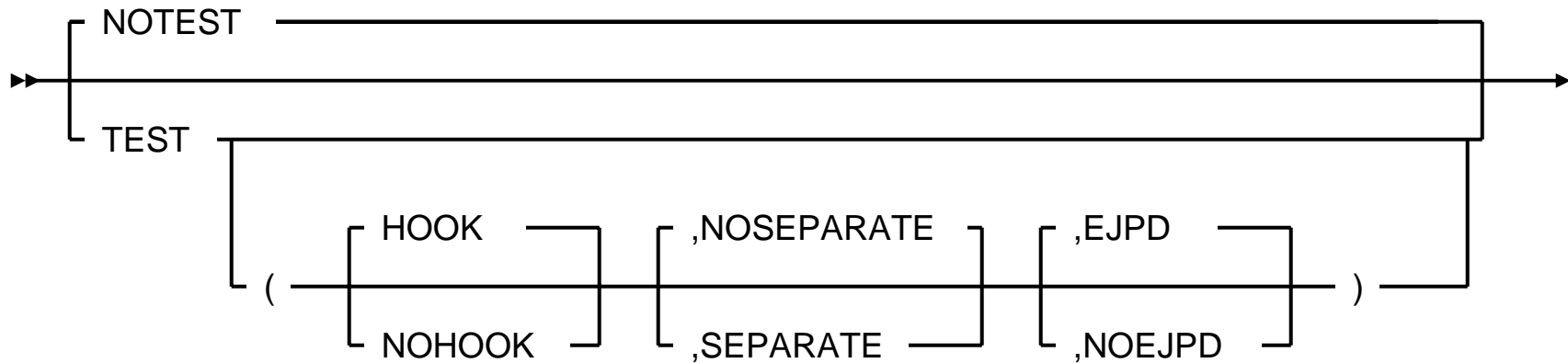
## ***Debugging enhancement - background***

- Debug Tool supports debugging of production COBOL programs
  - Compile options: OPTIMIZE, TEST(NONE,SYM,SEPARATE)
  - Debug Tool options: SET DYNDEBUG ON
  - Optimized program, no compiled in hooks, separate debug information tables, dynamic debug facility
- Current restriction:
  - Debug Tool commands **GOTO, JUMPTO** cannot be used with optimized programs
- Many companies have asked for this function...
- Cannot reliably be enabled for fully optimized programs
- Current (V3) TEST compiler option is too complicated
  - contains obsolete suboptions

## ***Production debugging: EJPD suboption***

- New suboption EJPD of compiler TEST option  
(Enable Jumpto for Production Debugging)
- Enables GOTO and JUMPTO commands for dynamically debugging optimized production programs
- Program optimization somewhat reduced:
  - optimization will be done within statements
  - most optimizations will not cross statement boundaries
  - program performance will be in between the performance of programs compiled with current NOOPTIMIZE and OPTIMIZE
    - sufficient optimization for production deployment in many scenarios
- TEST suboptions simplified

## *New TEST compiler option*



*(Old suboptions also supported for compatibility)*

Options for production debugging with JUMPTO/GOTO enabled:

**OPTIMIZE, TEST(NOHOOK,SEPARATE,EJPD)**

# ***Enterprise COBOL V4 prerequisites***

- z/OS: V1R7, V1R8, V1R9
- DB2: V7, V8, 9
- CICS: V2, V3
- Debug Tool: V7, V8
- z/OS XML System Services: APAR OA22777
- z/OS Language Environment: APAR PK55645