

IBM COBOL Seminar

Stephen Miller

IBM Software Group

December 4, 2008

Presentation overview

- IBM COBOL history, interoperability, migration
- Enterprise COBOL V4 new features

IBM COBOL history

- 1960s: OS/VS COBOL 5740-CB1
 - *ISO COBOL 68 and 74 standard*
- 1980s: VS COBOL II 5688-958
 - *Compiler rewrite: new technology*
 - *ISO COBOL 85 standard*
 - *31-bit addressing*
 - *Reentrancy*
 - *New CICS interface*
 - *DBCS*
- 1990s: COBOL/370 V1R1 5688-197
COBOL for MVS & VM V1R2
 - *Language Environment*
 - *Intrinsic functions*
 - *Debug Tool*
- 1997: COBOL for OS/390 & VM V2 5648-A25
 - *Dynamic Link Libraries*
 - *31-digit decimal data*
 - *OS/390 Unix*
 - *DB2 coprocessor*
- 2000s: Enterprise COBOL V3 5655-G53
 - ...
- **12/2007: Enterprise COBOL V4R1 5655-S71**

Enterprise COBOL V3

- Internationalization
 - Full Unicode support across the language
 - Coordinated with DB2 Unicode support
 - Based on z/OS Unicode Services
- XML support via native syntax
 - XML parsing
 - XML generation
- COBOL:Java interoperation
 - Mix COBOL and Java within an application
- CICS and DB2 coprocessors
- Production debugging with Debug Tool
- Large data items: raised 16 MB limit to 128 MB

Enterprise COBOL: interoperability

- CICS and DB2 coprocessors
 - Compile programs containing EXEC CICS, EXEC SQL, EXEC DLI commands in one step
 - Debug at original source level as written by programmer
 - COPY books can contain EXEC commands
- DB2 databases
 - New DB2 9 SQL syntax
 - Interoperation with DB2 Unicode databases
- IMS
 - Traditional MPR, BMP transaction processing and databases, or
 - New IMS Java regions: JMP, JBP
- WebSphere
 - COBOL clients can invoke methods on Enterprise Java Beans (EJBs)

Enterprise COBOL: interoperability ...

- Rational Developer for System z
 - Modern, Eclipse-based development environment
 - Remote z/OS development from Windows
 - XML and Web service tools
- Problem Determination Tools:
 - Debug Tool
 - Fault Analyzer
 - File Manager
 - Application Performance Analyzer
- WSAA / RAA
 - Asset analysis
 - Impact analysis

Enterprise COBOL migration

- Enterprise COBOL V3R4 and V4R1 are the only z/OS compiler levels now supported by IBM service
- Load modules created by previous compilers are still supported when run under Language Environment in z/OS batch, but ...
 - CICS TS V3 no longer supports OS/VS COBOL or VS COBOL II load modules
 - DB2 V8 and V9 support only Enterprise COBOL V3R4 or V4R1 compilers for new application development
- **Migration from prior compilers to Enterprise COBOL is strongly recommended!**
 - *Enterprise COBOL Compiler and Runtime Migration Guide* provides detailed guidance

Enterprise COBOL V4 new features

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

Performance enhancements

- Compiler generates new z/Architecture instructions to improve performance of COBOL programs
- Significantly improves 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 new features

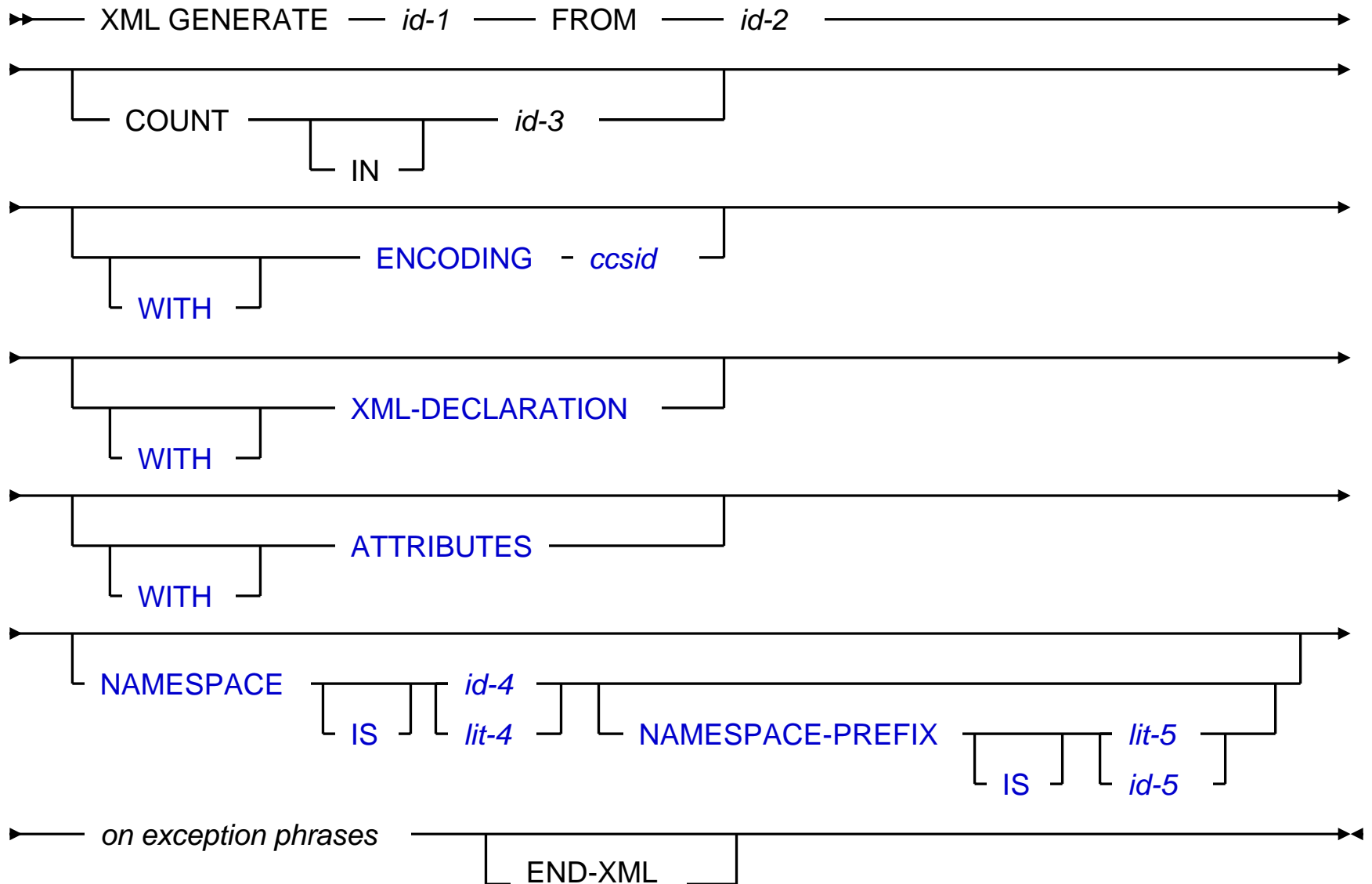
- Performance improvements
- **XML enhancements**
- DB2 9 support
- Usability enhancements
- Debugging enhancement

XML generation enhancements

More control over generated XML output:

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

XML GENERATE syntax



XML GENERATE with XML declaration and 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>

Another 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

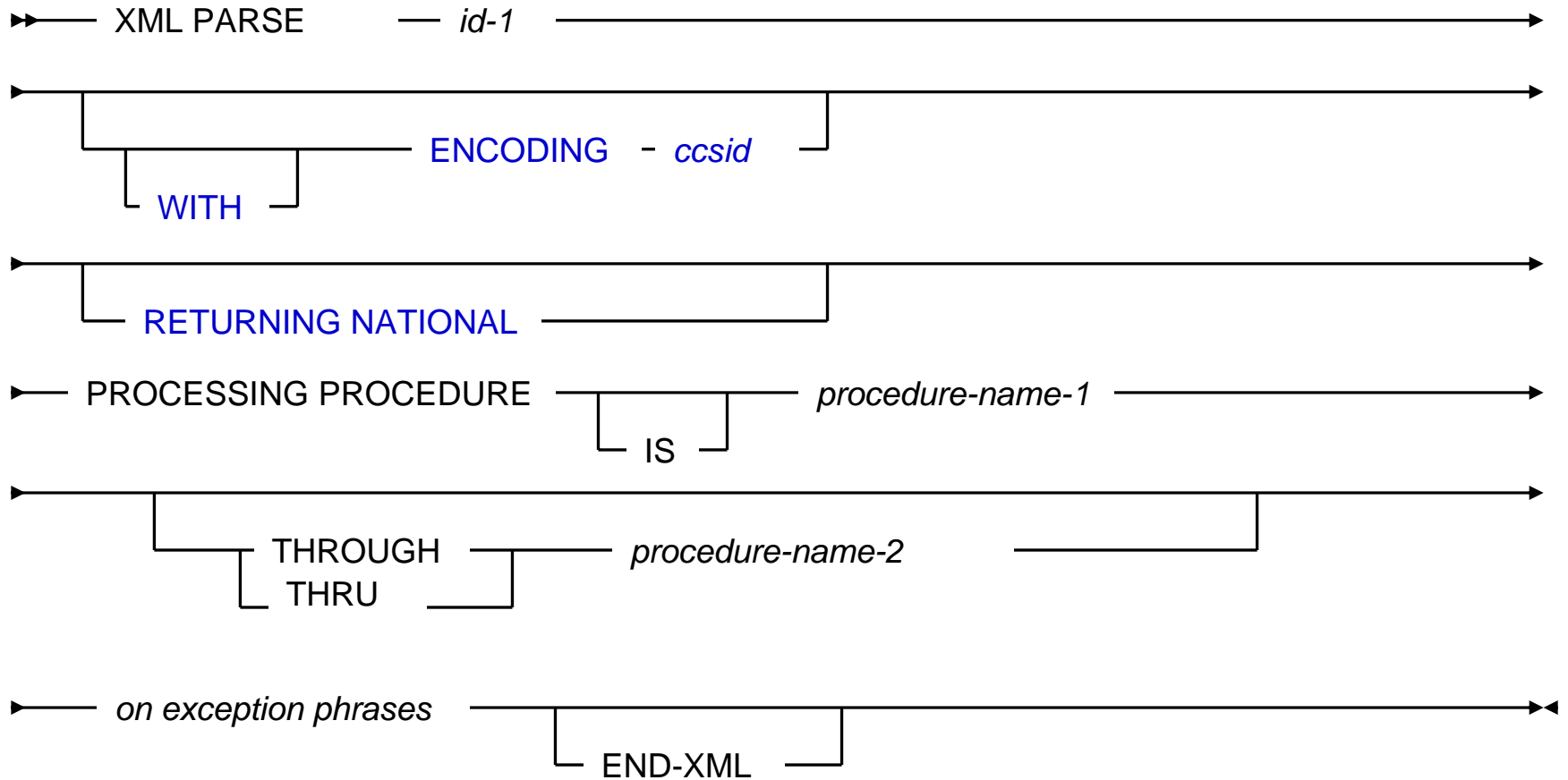
XML PARSE enhancements

- 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
- z/OS 1.10 XML System Services adds optional XML validation capability
 - IBM plans to exploit from COBOL in a future release

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 code pages
 - 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'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

Enterprise COBOL V4 new features

- 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 new features

- 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 data sets in compiler listing
- DB2 options in the compiler listing
 - *requires DB2 9 coprocessor*
- SQLCA and SQLDA expansion in compiler listing
 - *available with either V8 or DB2 9 coprocessors*
- Compilation 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

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: 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))

Enterprise COBOL V4 new features

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

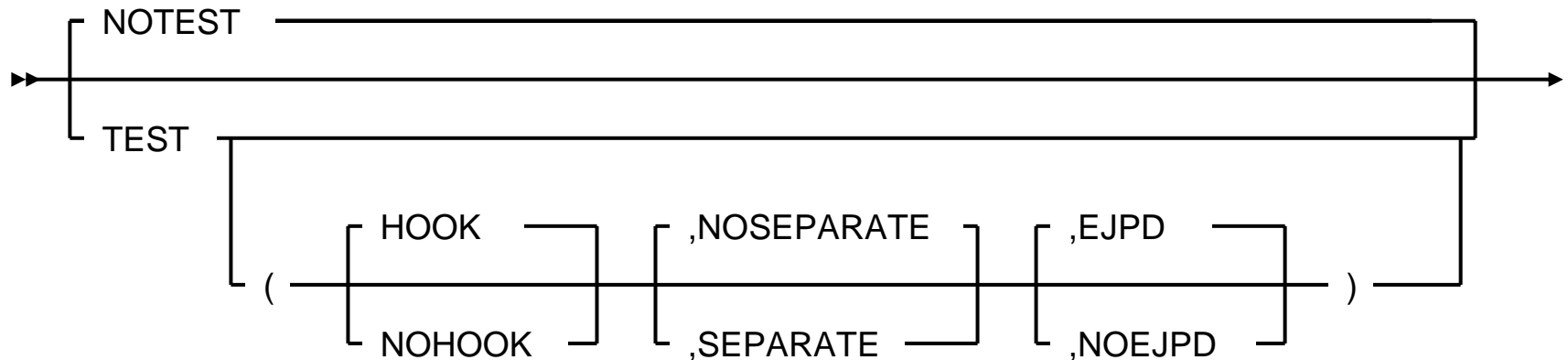
Debugging enhancement: background

- Debug Tool supports debugging of production COBOL programs
 - Compiler 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
- Enterprise COBOL V3 restriction:
 - Debug Tool commands **GOTO, JUMPTO** cannot be used with optimized programs

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
- TEST suboptions simplified

TEST compiler option: simplified



- Old suboptions also supported for compatibility
- Options for production debugging with JUMPTO/GOTO enabled:
OPTIMIZE(FULL), TEST(NOHOOK,SEPARATE,EJPD)

Enterprise COBOL V4 prerequisites

- z/OS: V1R7, V1R8, V1R9, V1R10
- DB2: V7, V8, 9
- CICS: V2, V3
- Debug Tool: V7, V8
- z/OS XML System Services: APAR OA22777
- z/OS Language Environment: APAR PK55645