

## **2B**

# **Generating DDL from a DDS 2E Model with No Changes**

**Crispin Bates – CPU Inc**



# About Me

- I work for CPU, Inc.
- CPU, Inc. (<http://www.cpulink.net>) is a CA Partner based in Cincinnati, Ohio focusing on IBM i, Microsoft Windows and Internet application development. Our expertise is in designing, constructing and deploying CA 2E/Plex and RPG applications, as well as Sencha 's Ext JS user interfaces using Zend Server.
- I started my programming career in 1988 using RPGII in the UK.
- I have been a 2E user since 1989.
- I have been a Plex user since 1998.

# Session Abstract

- This presentation will look at getting a 2E Model that had been generating DDS and RPG4 Code to generate DDL for the database definitions without changing the model generation options.
- We wanted minimal impact to existing code, but wanted to generate DDL files for our DB2 Web Query implementation.
- We will look at the generator exit points, calling API's and parsing the generated DDL.

# Agenda

- 1. What are we doing and why?
- 2. How are we going to do this?
- 3. Putting it all together.
- 4. Questions & Answers.

# What are we doing and why?

- We are a fairly traditional 2E RPG shop developing 5250 applications using DDS for the Database
- We also use Web Option
- We looked at DB2 Web Query and decided we wanted to implement it for our users.
- DB2 Web Query promised to be a new Business Analytics tool for our users to access the wealth of data that they have in our application.

# What are we doing and why?

- One of the first requests that came from our business group was that the field names in the DB2 Web Query tool match the fields business name, and not the DDS Name.
- We also noticed that for some of the reports that we were creating we were getting poor performance. The system was suggesting that we create SQL Index's to improve performance.
- At this point I started looking at the SQL Generation options in 2E, and quickly determined that it would be a huge impact to change to SQL Generation.

# What are we doing and why?

- The next thing I started looking at was manually creating Views and Index's that would provide the Long Field names and the Access plans that the system was suggesting.
- My first pass at these produced good results, and our Business people were really happy with the results.
- But...this was all manual work, and I did not want to continue doing it this way.

# What are we doing and why?

- It was around this time that I was browsing the CA Community Forums and I came across a really interesting thread that was discussing mixing DDS and DDL within the same 2E Model
- The question that was asked was whether it was possible to mix DDS and DDL in the same Model.
- Several references were posted.
  - Dan Cruickshank of IBM had produced a document about Database Modernization that talked about replacing DDS with DDL without having to recompile any programs and this was what was being used for reference
  - There was also a good article on the Plex/2E Wiki
    - [http://wiki.2einfo.net/index.php?title=Using\\_SQL\\_with\\_2E](http://wiki.2einfo.net/index.php?title=Using_SQL_with_2E)



# How are we going to do this?

- So, I read Dan's document on database modernization (not for the first time) and then started playing around with some of the newer DDL options that were in V5R4, in particular the RCDFMT clause on the CREATE TABLE specification
- After an hour or two I had recreated a DDS PF as a DDL TABLE, and that TABLE had the same Record Format Level Identifier as the original DDS PF. I could use this TABLE without affecting any programs in my 2E Model. Now I was getting somewhere.

# Data Validation DDS vs. DDL

- One thing to note is the difference between the way data is validated in a DDL TABLE vs. a DDS PF.
- Data is validated when written to a DDL TABLE
- Data is validated when read from a DDS PF
- We ran some analysis over our PF's looking at the Read vs. Write statistics returned via DSPFD.
- We found that about 98% or more of the data access for almost all of our PF's were Read Access.
- This suggested that DDL would provide better performance because validation is not done on Read Access.

# How are we going to do this?

- I then came across the QSQGNDDL API. This is where things got really interesting
- Additionally I read that at IBM i 6.1 the INDEX DDL would allow the specification of the RCDFMT clause. This was the last piece in the puzzle, as I could now replace the DDS LF's with DDL INDEX's
- So, now for the interesting stuff...

- From the infocenter
  - The Generate Data Definition Language (QSQGNDDL) API generates the SQL data definition language statements required to recreate a database object. The results are returned in the specified database source file member.

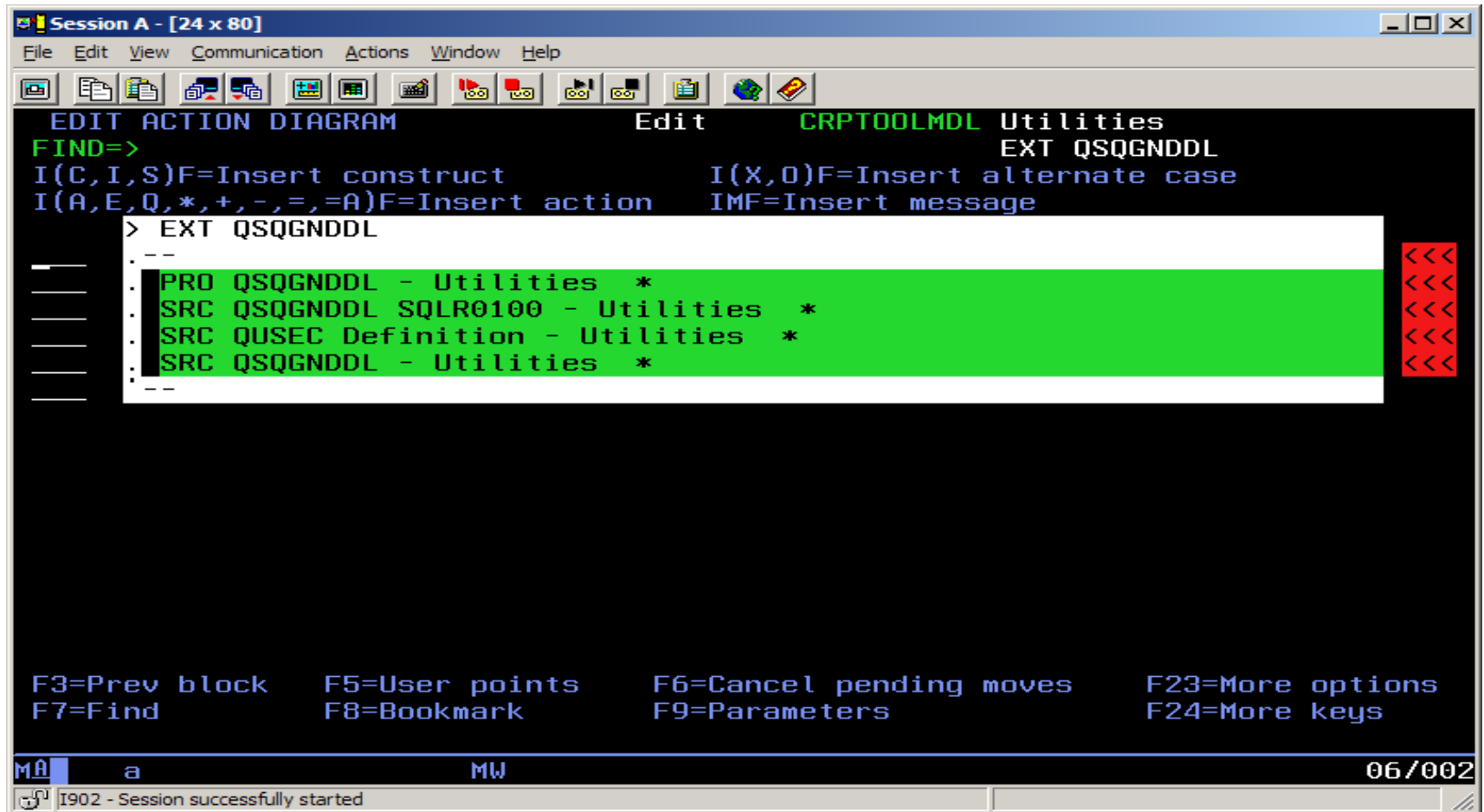
# Prerequisites

- Some things that I need to say here
  - DDL TABLE's force REUSEDLT(\*YES)
  - Multi Member PF's not supported in SQL
  - Joins (virtuals) not supported in Index
  - QSQGNDDL does not correctly generate DDL for Select/Omit specifications when an INDEX is the target object
    - Note: DCREQ Opened with IBM Rochester and was accepted
  - So, I need to cater for this later on...

- There are several good resources where you can find downloadable code that implements the QSQGNDDL API.
- Carsten Flensburg provided a really good article here, with downloadable code. Requires Pro Membership to SystemiNetwork
  - <http://systeminetwork.com/article/apis-example-reverse-engineering-database-files-and-objects-sql-ddl-statements>

- Tom Holden has a simplified version that works well here
  - <http://www.tommyholden.com/downloads/Save%20Files/index.html>
- You can also roll your own in 2E...

# QSQGNDDL Implementation



The screenshot shows a terminal window titled "Session A - [24 x 80]" with a menu bar (File, Edit, View, Communication, Actions, Window, Help) and a toolbar. The main text area displays the following content:

```
EDIT ACTION DIAGRAM          Edit      CRPTOOLMDL Utilities
FIND=>                        EXT QSQGNDDL
I(C,I,S)F=Insert construct    I(X,O)F=Insert alternate case
I(A,E,Q,*,+,-,=,=A)F=Insert action  IMF=Insert message
> EXT QSQGNDDL
--
. PRO QSQGNDDL - Utilities *
. SRC QSQGNDDL SQLR0100 - Utilities *
. SRC QUSEC Definition - Utilities *
. SRC QSQGNDDL - Utilities *
--
```

On the right side of the terminal, there are four red arrows pointing left, each preceded by two red less-than signs (<<).

At the bottom of the terminal, there is a status bar with the following information:

- MA a MW
- 06/002
- I902 - Session successfully started

Below the terminal window, there is a table of function key shortcuts:

F3=Prev block	F5=User points	F6=Cancel pending moves	F23=More options
F7=Find	F8=Bookmark	F9=Parameters	F24=More keys



# QSQGNDDL Prototype

```
Session A - [27 x 132]
File Edit View Communication Actions Window Help

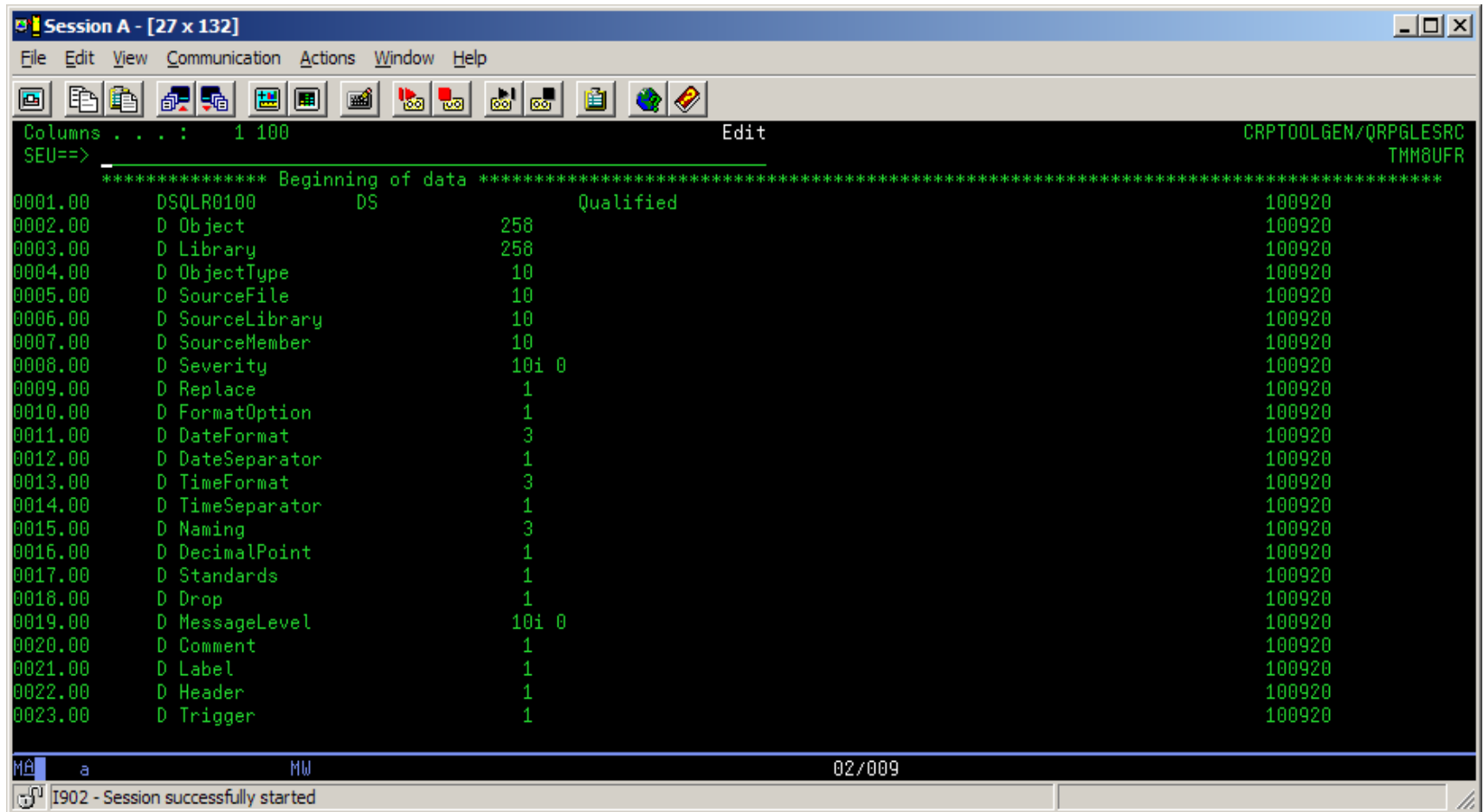
Columns . . . : 6 100
SEU==>

***** Beginning of data *****
0001.00 dQsqGnDDL      pr      ExtPgm('QSQGNDDL')      100920
0002.00 d Template      1000a  Options(*VarSize) Const 100920
0003.00 d TemplateLength 10i 0  Const      100920
0004.00 d Format        8a      Const      100920
0005.00 d ErrorCode     Like(QUSEC)      100920
***** End of data *****

(C) COPYRIGHT IBM CORP. 1981, 2007.

MA a MW 02/009
I902 - Session successfully started
```

# QSQGNDDL Datastructure



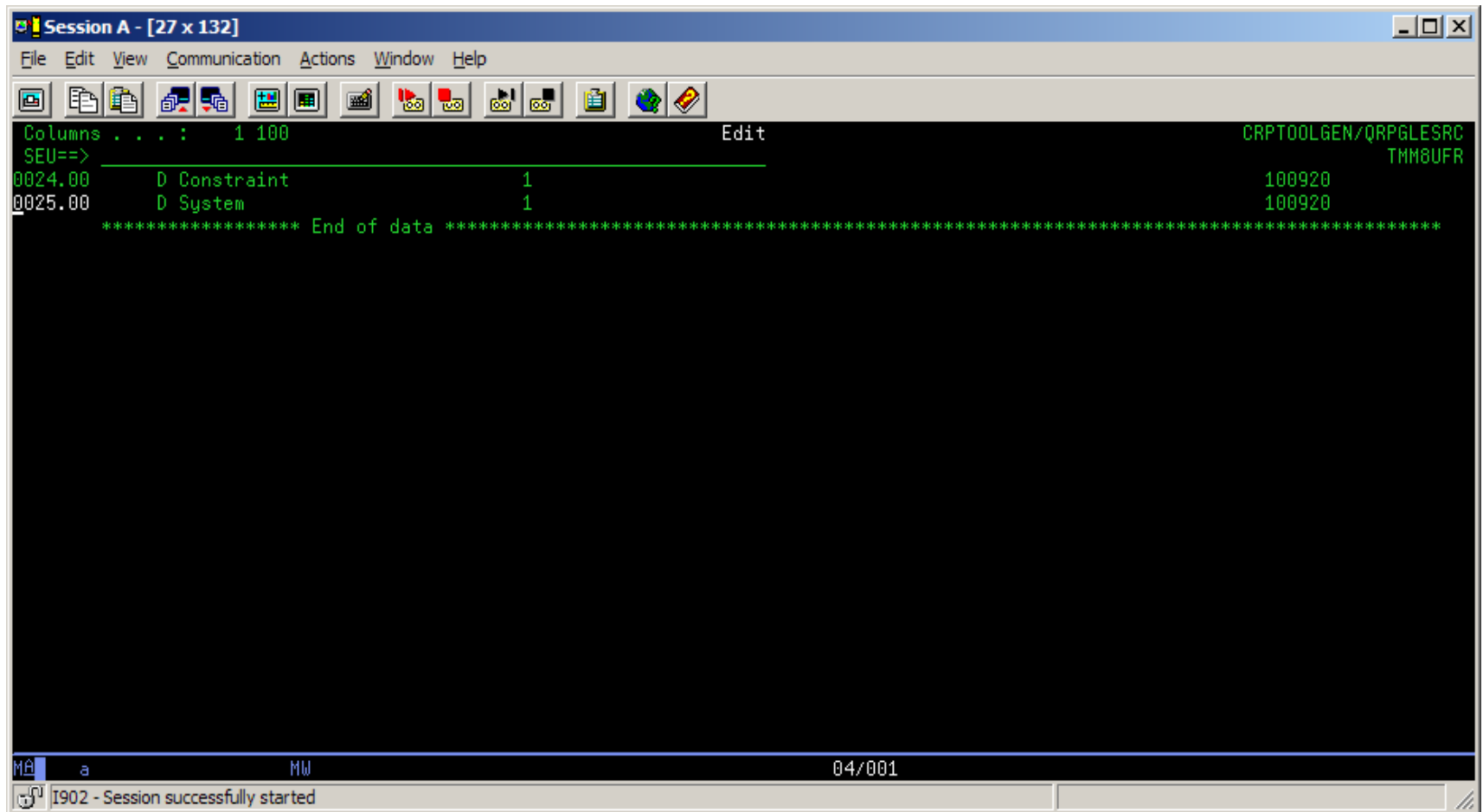
The screenshot shows a window titled "Session A - [27 x 132]" with a menu bar (File, Edit, View, Communication, Actions, Window, Help) and a toolbar. The main area displays a table of data structure fields. The table has columns for a numeric identifier, a field name, a data type, a length, and a value. The data is organized into sections: a header section, a "Beginning of data" section, and a "Qualified" section. The "Beginning of data" section lists fields from 0001.00 to 0023.00. The "Qualified" section lists fields from 0002.00 to 0023.00. The table is titled "CRPTOOLGEN/QRPGLESRC TMM8UFR".

Columns . . . :	1	100	Edit	CRPTOOLGEN/QRPGLESRC TMM8UFR
***** Beginning of data *****				
0001.00	DSQLR0100	DS	Qualified	100920
0002.00	D Object	258		100920
0003.00	D Library	258		100920
0004.00	D ObjectType	10		100920
0005.00	D SourceFile	10		100920
0006.00	D SourceLibrary	10		100920
0007.00	D SourceMember	10		100920
0008.00	D Severity	10i 0		100920
0009.00	D Replace	1		100920
0010.00	D FormatOption	1		100920
0011.00	D DateFormat	3		100920
0012.00	D DateSeparator	1		100920
0013.00	D TimeFormat	3		100920
0014.00	D TimeSeparator	1		100920
0015.00	D Naming	3		100920
0016.00	D DecimalPoint	1		100920
0017.00	D Standards	1		100920
0018.00	D Drop	1		100920
0019.00	D MessageLevel	10i 0		100920
0020.00	D Comment	1		100920
0021.00	D Label	1		100920
0022.00	D Header	1		100920
0023.00	D Trigger	1		100920

MA a MW 02/009

I902 - Session successfully started

# QSQGNDDL Datastructure 2



The screenshot shows a terminal window titled "Session A - [27 x 132]". The window has a menu bar with "File", "Edit", "View", "Communication", "Actions", "Window", and "Help". Below the menu bar is a toolbar with various icons. The main area displays the following text:

```
Columns . . . : 1 100 Edit CRPTOOLGEN/QRPGLESRC
SEU=> TMM8UFR
0024.00 D Constraint 1 100920
0025.00 D System 1 100920
***** End of data *****
```

At the bottom of the window, there is a status bar showing "MA a MW 04/001" and a message "I902 - Session successfully started".

# QUSEC Definition

```
Session A - [27 x 132]
File Edit View Communication Actions Window Help

Columns . . . : 6 100
SEU=>

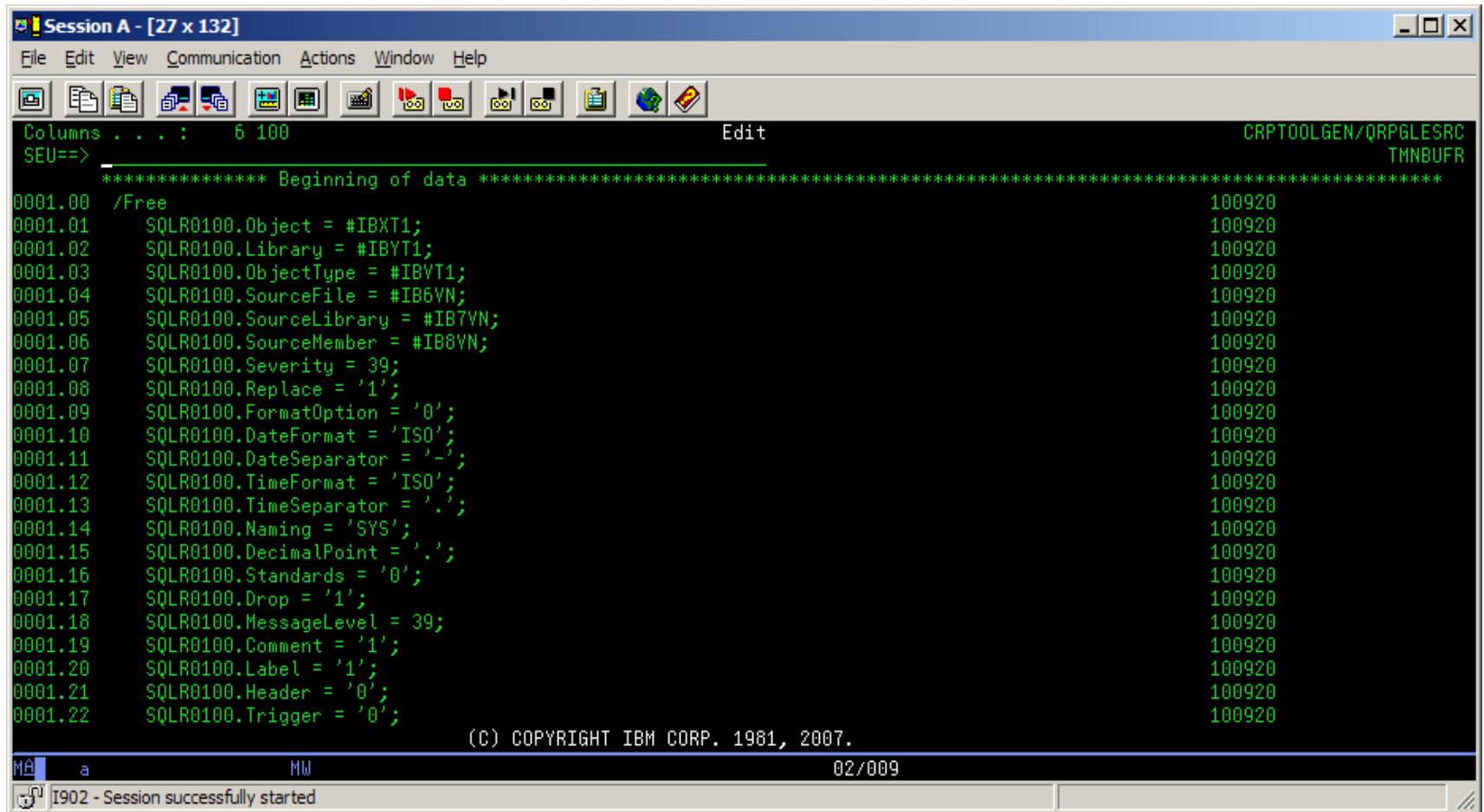
***** Beginning of data *****
0001.00 /IF NOT DEFINED(QUSEC)
0002.00 /DEFINE QUSEC
0003.00 /ELSE
0004.00 /EOF
0005.00 /ENDIF
0006.00 D QUSEC DS
0007.00 D QUSBPRV 10I 0
0008.00 D BytesProv 10I 0 Overlay(QUSBPRV)
0009.00 D QUSBAYL 10I 0
0010.00 D BytesAvail 10I 0 Overlay(QUSBAYL)
0011.00 D QUSEI 7A
0012.00 D APIErrMsgID 7A Overlay(QUSEI)
0013.00 D 1A
0014.00 D QUSEDTA 512A
0015.00 D APIErrDta 512A Overlay(QUSEDTA)

***** End of data *****

(C) COPYRIGHT IBM CORP. 1981, 2007.

MA a MW 02/009
I902 - Session successfully started
```

# QSQGNDDL Implementation



The screenshot shows a window titled "Session A - [27 x 132]" with a menu bar (File, Edit, View, Communication, Actions, Window, Help) and a toolbar. The main area displays the following text:

```
Columns . . . : 6 100                      Edit                      CRPTOOLGEN/QRPGLESRC
SEU==>                                           TMNBUFR

***** Beginning of data *****

0001.00 /Free                                     100920
0001.01   SQLR0100.Object = #IBXT1;                100920
0001.02   SQLR0100.Library = #IBYT1;                100920
0001.03   SQLR0100.ObjectType = #IBYT1;              100920
0001.04   SQLR0100.SourceFile = #IB6VN;              100920
0001.05   SQLR0100.SourceLibrary = #IB7VN;           100920
0001.06   SQLR0100.SourceMember = #IB8VN;            100920
0001.07   SQLR0100.Severity = 39;                    100920
0001.08   SQLR0100.Replace = '1';                    100920
0001.09   SQLR0100.FormatOption = '0';                100920
0001.10   SQLR0100.DateFormat = 'ISO';                100920
0001.11   SQLR0100.DateSeparator = '-';              100920
0001.12   SQLR0100.TimeFormat = 'ISO';                100920
0001.13   SQLR0100.TimeSeparator = '.';              100920
0001.14   SQLR0100.Naming = 'SYS';                   100920
0001.15   SQLR0100.DecimalPoint = '.';                100920
0001.16   SQLR0100.Standards = '0';                   100920
0001.17   SQLR0100.Drop = '1';                       100920
0001.18   SQLR0100.MessageLevel = 39;                 100920
0001.19   SQLR0100.Comment = '1';                     100920
0001.20   SQLR0100.Label = '1';                       100920
0001.21   SQLR0100.Header = '0';                      100920
0001.22   SQLR0100.Trigger = '0';                     100920

(C) COPYRIGHT IBM CORP. 1981, 2007.
```

At the bottom, there is a status bar showing "a" and "MW" on the left, "02/009" in the center, and "I902 - Session successfully started" on the right.

# QSQGNDDL Implementation

```
Session A - [27 x 132]
File Edit View Communication Actions Window Help

Columns . . . : 6 100
SEU=>
0001.23 SQLR0100.Constraint = '0'; 100920
0001.24 SQLR0100.System = '0'; 100920
0001.25 BytesProv = %size(APIErrDta); 100920
0001.26 100920
0001.27 QsqGnDDL(SQLR0100 : 100920
0001.28 %Size(SQLR0100): 100920
0001.29 'SQLR0100': 100920
0001.30 QUSEC); 100920
0002.00 /End-Free 100920
***** End of data *****

a MW 02/009
I902 - Session successfully started
```

# Parameters

Session A - [24 x 80]

File Edit View Communication Actions Window Help

Op: CBATES QPADEV000Q 5/16/11 19:10:20  
CRP Tools Model

EDIT FUNCTION PARAMETERS

Function name. . : EXT QSQGNDDL Type : Execute external function  
Received by file : Utilities Acpth: \*NONE

?	File/*FIELD	Access path/Field/Array	Passed	Seq	Pgm Ctx	Par Ctx
—	*Arrays	QSQGNDDL Parameters	FLD	1		
—						
—						
—						
—						
—						
—						
—						
—						

Values

FLD: One parameter per field  
RCD: One parameter for all fields  
KEY: One parameter for key fields only

SEL: Z-Parameter details X-Object details D-Delete parameter N-Narrative  
F3=Exit F5=Reload F22=File locks F23=More options

MA a MW 07/003

I902 - Session successfully started

# Parameters

```
Session A - [24 x 80]
File Edit View Communication Actions Window Help

Op: CBATES QPADEV000Q 5/16/11 19:10:39
CRP Tools Model

EDIT FUNCTION PARAMETER DETAILS
Function name. . : EXT QSQGNDDL Type : Execute external function
Received by file : Utilities Array: QSQGNDDL Parameters
Parameter (file) : *Arrays Passed as: FLD

? Field Usage Role Flag error
- SQL Database Object I
- SQL Database Library I
- SQL Type I
- Source File I
- Source Library I
- Source Member I
- Source Text I

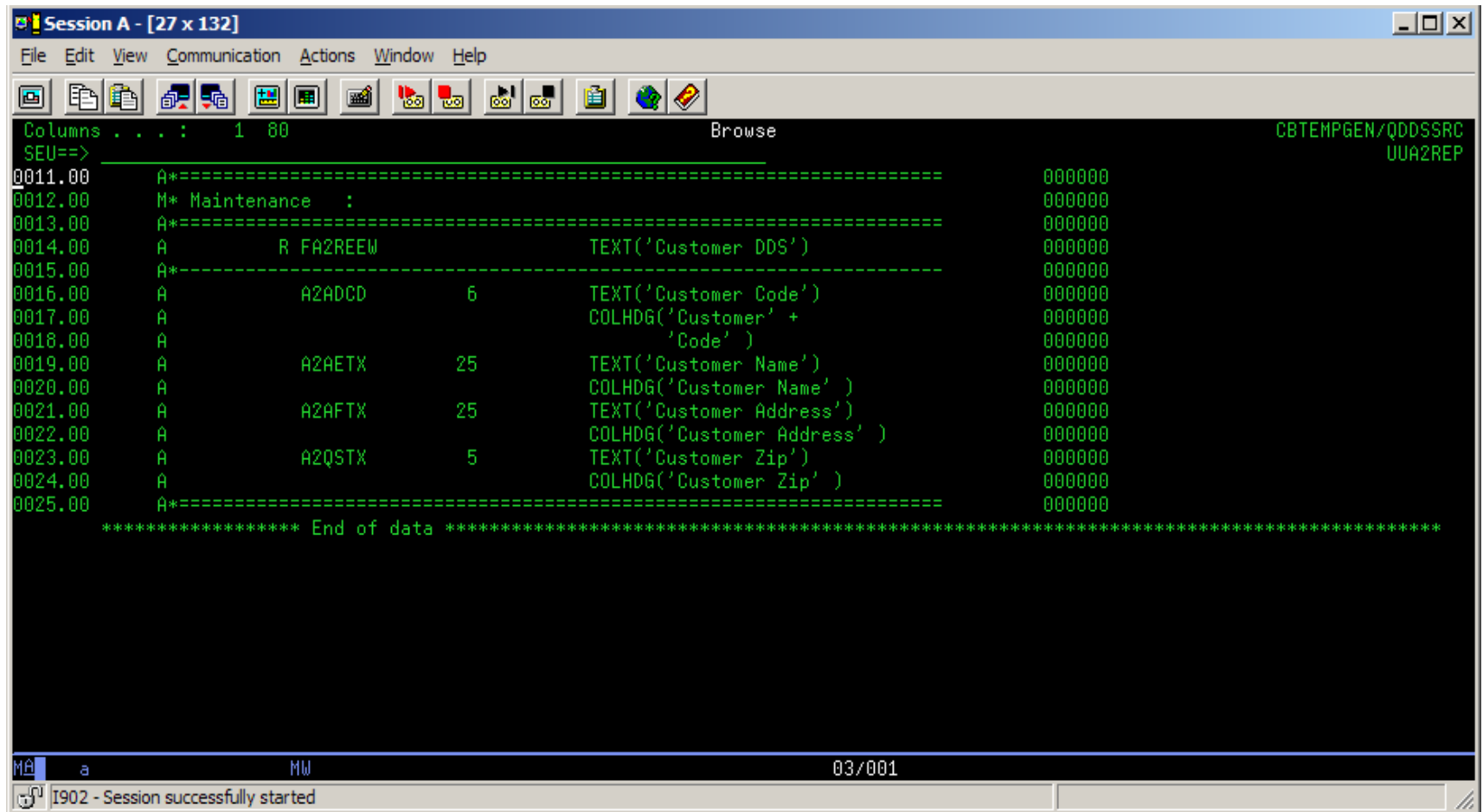
SEL: Usage: I-Input, O-Output, B-Both, N-Neither, D-Drop.
Role: R-Restrict, M-Map, V-Vary length, P-Position. Error: E-Flag Error.
F3=Exit

MA a MW 08/005
I902 - Session successfully started
```



- The QSQGNDDL API does not automatically create the Source Member in the Source file that is specified on the API. The API will fail if the member is not found. So, you have to make sure that the member exists, and is empty before calling the API.

# Example DDS

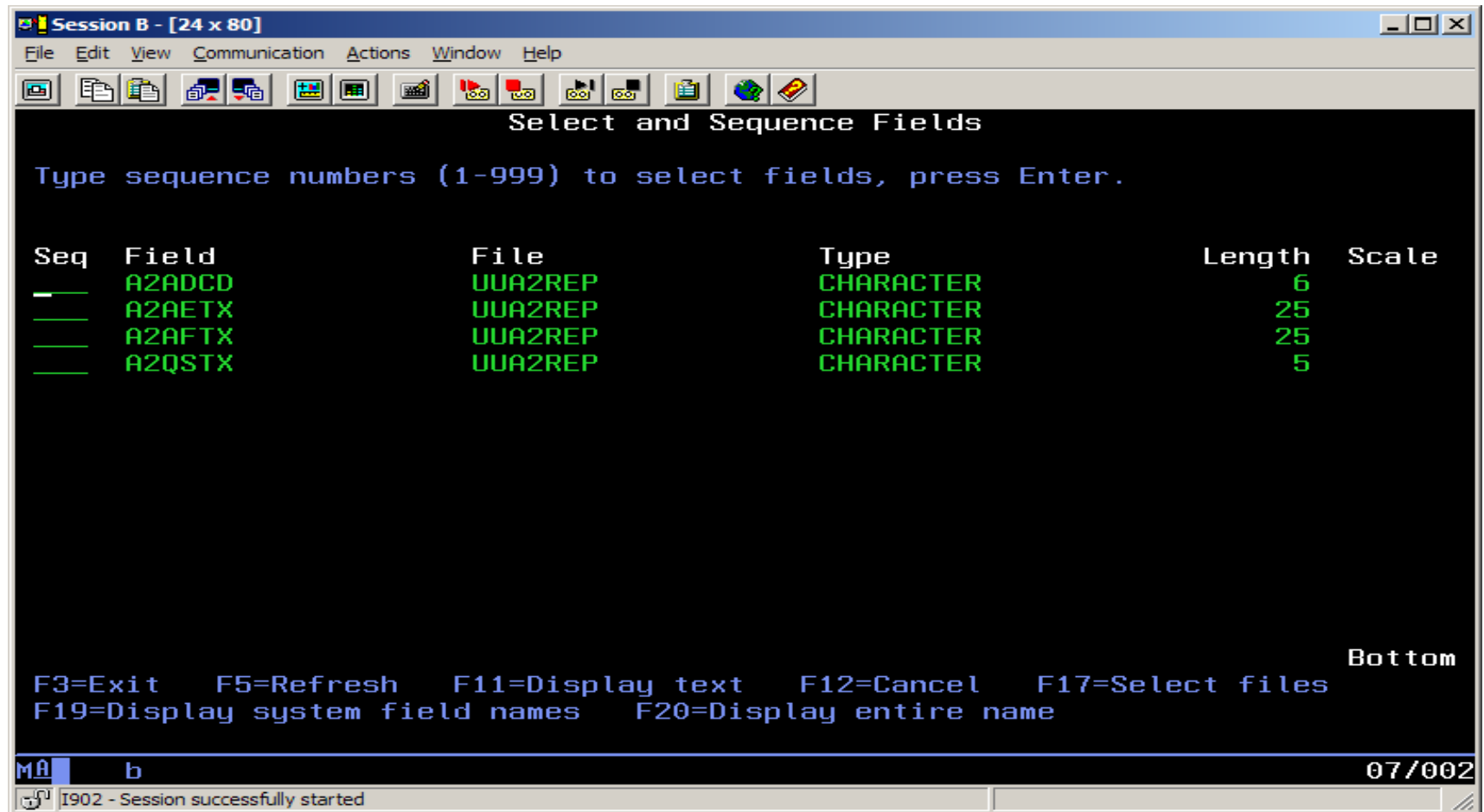


The screenshot displays the SAP Session A interface with a Data Dictionary Structure (DDS) for a customer table. The window title is "Session A - [27 x 132]". The menu bar includes File, Edit, View, Communication, Actions, Window, and Help. The toolbar contains various icons for file operations and navigation. The main area shows the DDS structure with columns and rows. The columns are labeled "Columns . . . : 1 80" and "Browse". The rows are labeled "SEU=>" and "CBTEMPGEN/QDDSSRC UUA2REP". The structure is defined by the following lines:

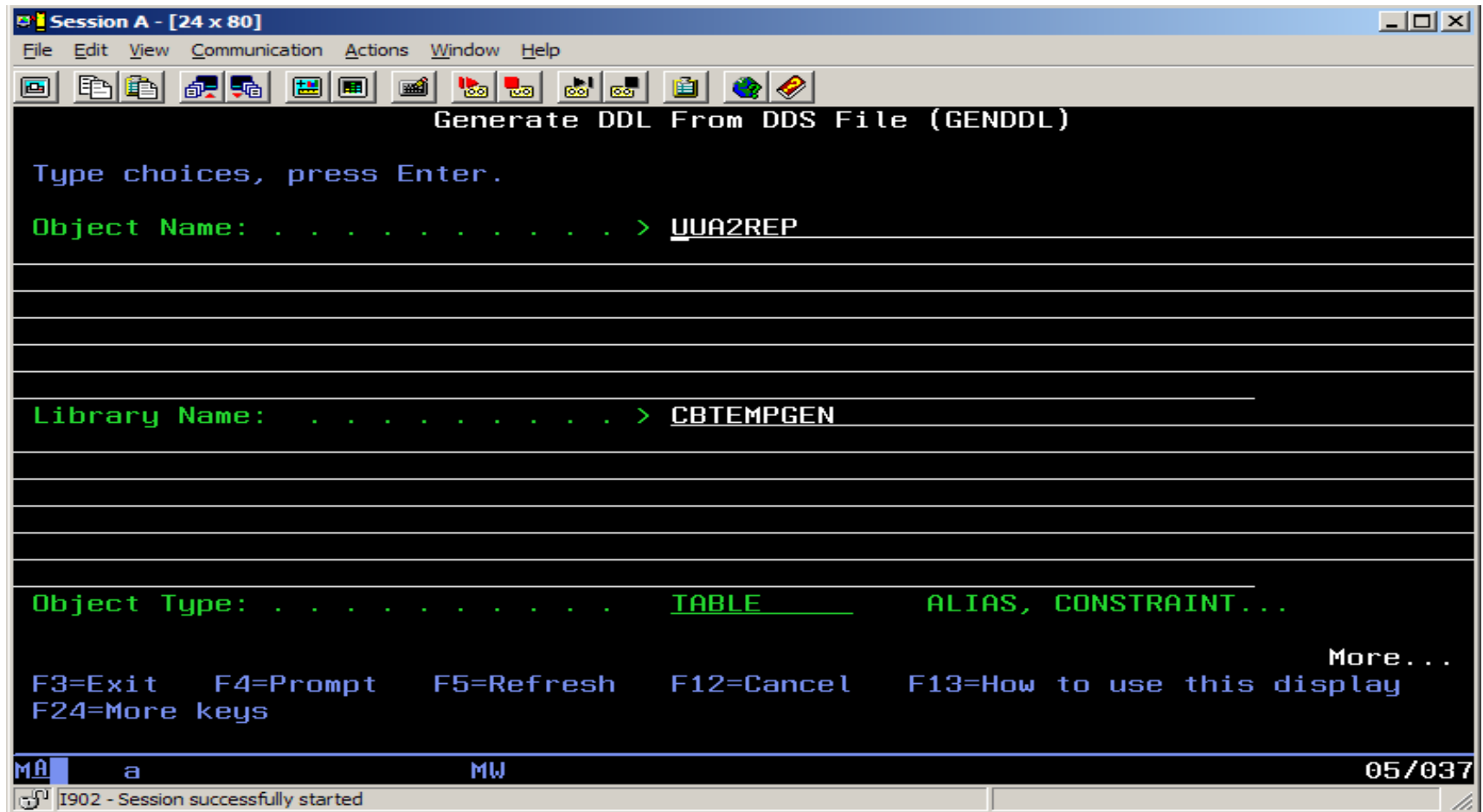
```
0011.00  A*=====
0012.00  M* Maintenance :
0013.00  A*=====
0014.00  A          R FA2REEW          TEXT('Customer DDS')
0015.00  A*-----
0016.00  A          A2ADCD          6          TEXT('Customer Code')
0017.00  A          COLHDG('Customer' +
0018.00  A          'Code' )
0019.00  A          A2AETX          25          TEXT('Customer Name')
0020.00  A          COLHDG('Customer Name' )
0021.00  A          A2AFTX          25          TEXT('Customer Address')
0022.00  A          COLHDG('Customer Address' )
0023.00  A          A2QSTX          5          TEXT('Customer Zip')
0024.00  A          COLHDG('Customer Zip' )
0025.00  A*=====
***** End of data *****
```

The status bar at the bottom shows "MA a MW 03/001" and "I902 - Session successfully started".

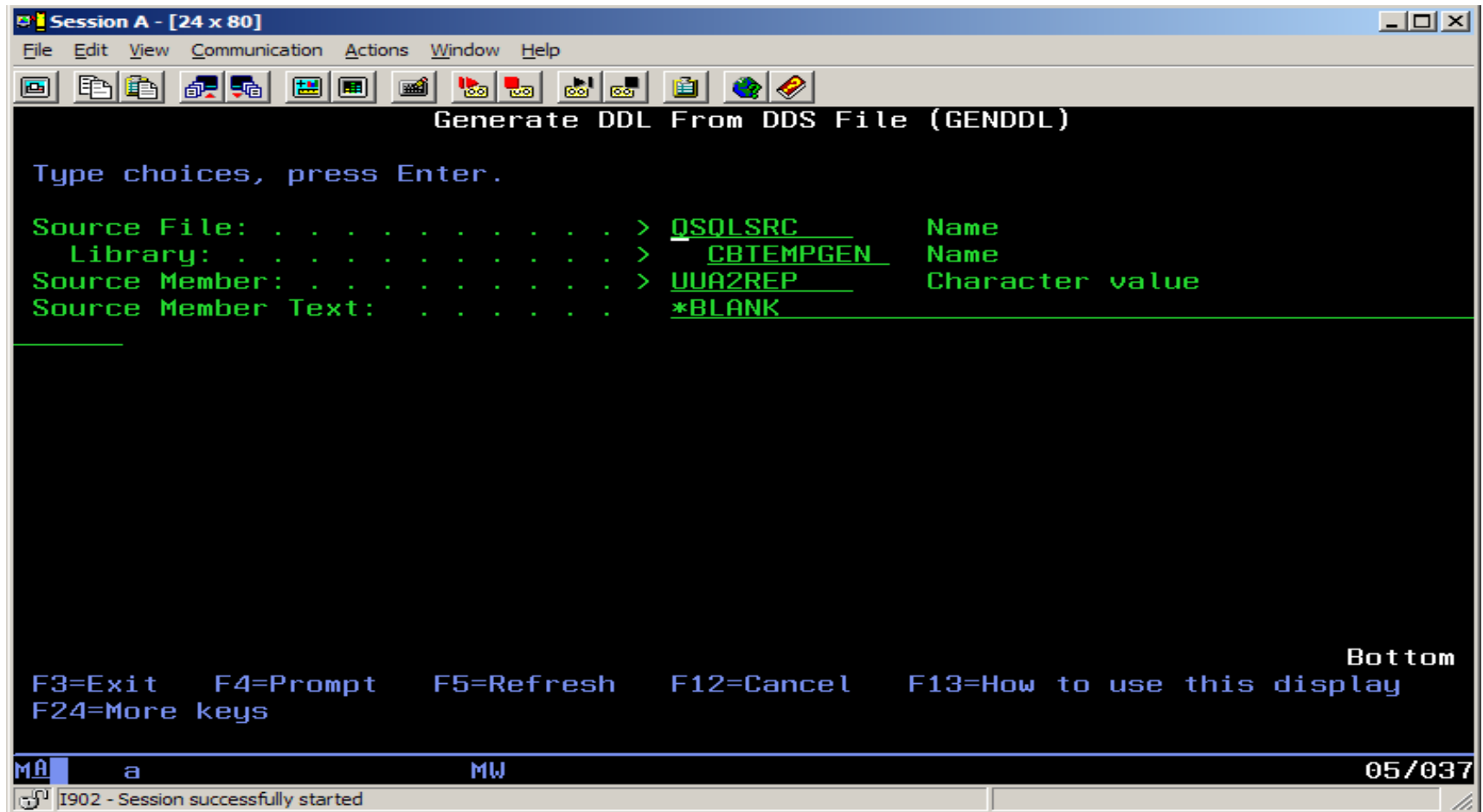
# Fields in SQL



# Run GENDDL



# GENDDL



# Resulting DDL

```
Session A - [27 x 132]
File Edit View Communication Actions Window Help

Columns . . . : 1 80 Browse CBTEMPGEN/SQLSRC
SEU=> UUA2REP

***** Beginning of data *****
0001.00 DROP TABLE CBTEMPGEN/UUA2REP ; 110518
0002.00 110518
0003.00 CREATE TABLE CBTEMPGEN/UUA2REP ( 110518
0004.00 A2ADCD CHAR(6) CCSID 37 NOT NULL DEFAULT '' , 110518
0005.00 A2AETX CHAR(25) CCSID 37 NOT NULL DEFAULT '' , 110518
0006.00 A2AFTX CHAR(25) CCSID 37 NOT NULL DEFAULT '' , 110518
0007.00 A2QSTX CHAR(5) CCSID 37 NOT NULL DEFAULT '' ) 110518
0008.00 RCDfmt FA2REEW ; 110518
0009.00 110518
0010.00 LABEL ON TABLE CBTEMPGEN/UUA2REP 110518
0011.00 IS 'Customer DDS Physical file' ; 110518
0012.00 110518
0013.00 LABEL ON COLUMN CBTEMPGEN/UUA2REP 110518
0014.00 ( A2ADCD IS 'Customer Code' , 110518
0015.00 A2AETX IS 'Customer Name' , 110518
0016.00 A2AFTX IS 'Customer Address' , 110518
0017.00 A2QSTX IS 'Customer Zip' ) ; 110518
0018.00 110518
0019.00 LABEL ON COLUMN CBTEMPGEN/UUA2REP 110518
0020.00 ( A2ADCD TEXT IS 'Customer Code' , 110518
0021.00 A2AETX TEXT IS 'Customer Name' , 110518
0022.00 A2AFTX TEXT IS 'Customer Address' , 110518
0023.00 A2QSTX TEXT IS 'Customer Zip' ) ; 110518
(C) COPYRIGHT IBM CORP. 1981, 2007.

MA a MW 02/009
I902 - Session successfully started
```

# Resulting DDL missing Long Names

- The resulting DDL is missing long field names which I had originally added in the manual DDL that I had created.
- This is one of the things that will be added in the Post-Compile processing

# 2E Compile Processing

- The 2E Compile Processor allows for User defined Exit Programs to be called.
- Data Area YBRT PXA controls the pre and post compile exit programs.
- 1 - 10 Pre-compile exit program
- 11 - 20 Pre-compile exit program library
- 21 - 30 Post-compile exit program
- 31 - 40 Post-compile exit program library



# 2e Compile Processing

- For the DDL Generation Process we take advantage of the Post-Compile Exit program.

```
Session A - [24 x 80]
File Edit View Communication Actions Window Help
Data area . . . . . : YB RTPXA
Library . . . . . : Y1SY85
Type . . . . . : *CHAR
Length . . . . . : 42
Text . . . . . : YC RT0VR Compile preprocessor default exit programs
System: ISSCDEV2

Offset      Value
0           '*...+...1...+...2...+...3...+...4...+...5'
           '*NONE' TMMFUPC CRPDEVUTL 1

Press Enter to continue.
F3=Exit  F12=Cancel
Bottom

M0 a MW 01/001
I902 - Session successfully started
```

# Exit Program

- CL Program TMMFUPC in library CRPDEVUTL is called as part of the post compile processing by 2E
- We can then take control of any processing that we want to happen.
- We have many pre and post compile steps that we need to happen each time an object is generated and compiled.
- After many years of changing CL Programs to cater for it we externalized it all into a database

# Processing File

```
Session A - [24 x 80]
File Edit View Communication Actions Window Help

Op: CBATES      QPADEV000Q  5/16/11 19:22:06
CRP Tools Model

DISPLAY FILE ENTRIES

File name. . . . : Post Gen Processes

? Field              Type      Ocr  Et  Ksq  GEN name  Length  Renamed
- PGM Process Stage   STS              K    1  ELS1      1
- PGM Process         CDE              K    2  JXCD      6
- PGM Program         VNM              A      F9VN     10
- PGM Process Status   STS              A      EKS1      1
- PGM Process Description TXT              A      A3T1     30
- PGM Process Sequence NBR              A      HDNB      5.0

SEL: Z-Field details, U-Field references, L-Locks.
F3=Exit

MA  a      MW      08/002
I902 - Session successfully started
```

# Processes

```
Session B - [24 x 80]
File Edit View Communication Actions Window Help

TMYEFR CHANGE 5/16/11 19:23:46
Edit Post Gen Processes CBATES

Process .

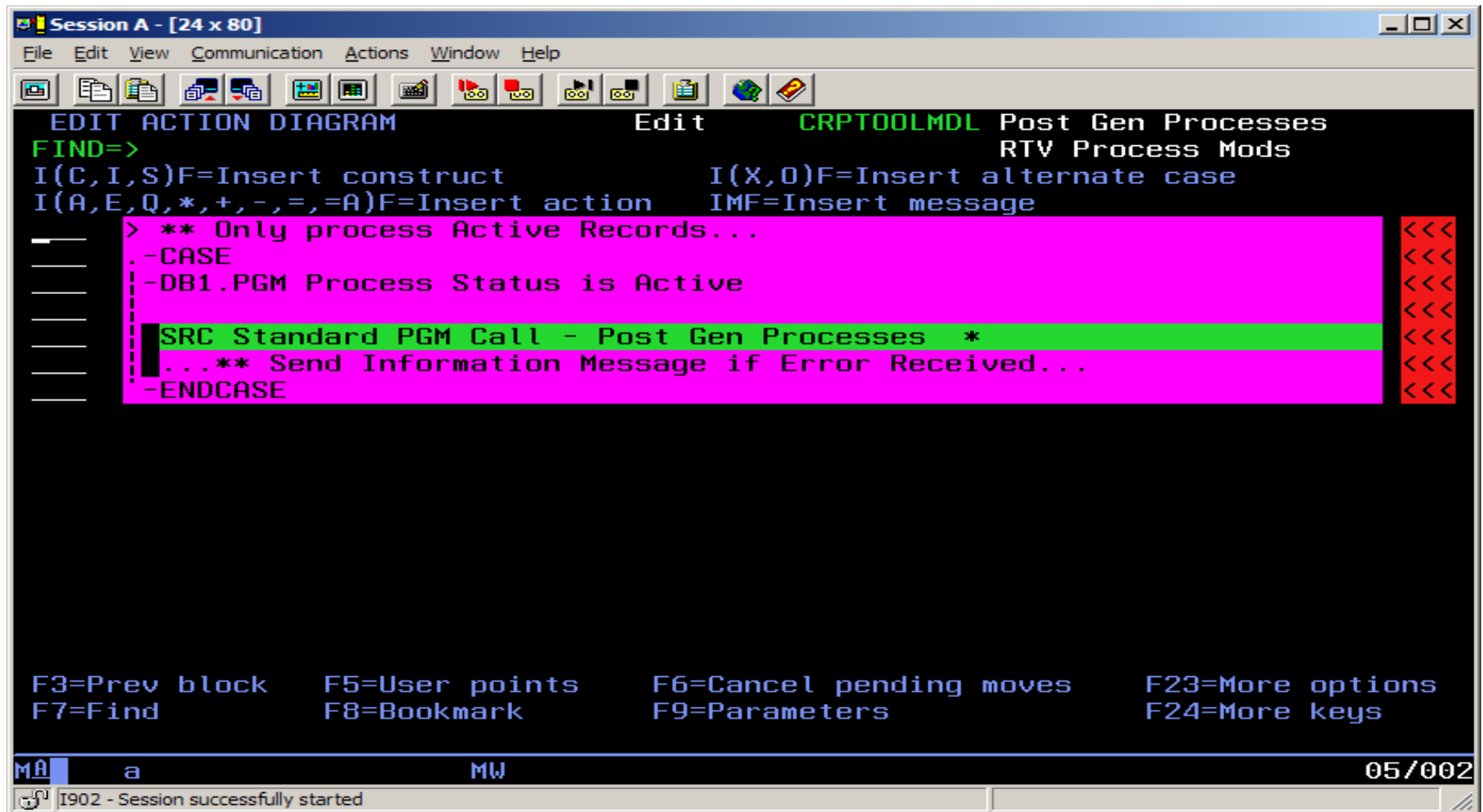
Type options, press Enter.
4=Delete

? Stage Process Program Status Sequence Description
- 0 ADDGEN TML3XFR A 1 Add Generation Tracking Record
- 0 MLSGEN TML4XFR A 3 YGENMLS Processing
- 0 PGMMOD TML2XFR A 2 Post Generation Mod Engine
- 1 ADDSRC TML7XFR A 1 Add Source Change Date/Time
- 1 DDLGEN TML6XFR A 2 DDL Generation
- 1 PGMADP TML9XFR A 3 Program Adoption Processing
- 1 TRNGEN TML5XFR A 4 Translation Analysis

F3=Exit F4=Prompt F9=Add

MA b 10/002
I902 - Session successfully started
```

# Post Processing



```
Session A - [24 x 80]
File Edit View Communication Actions Window Help

EDIT ACTION DIAGRAM          Edit      CRPTOOLMDL Post Gen Processes
FIND=>                        RTV Process Mods
I(C,I,S)F=Insert construct    I(X,O)F=Insert alternate case
I(A,E,Q,*,+,-,=,=A)F=Insert action IMF=Insert message

> ** Only process Active Records...
.-CASE
  -DB1.PGM Process Status is Active
  SRC Standard PGM Call - Post Gen Processes *
  ...** Send Information Message if Error Received...
.-ENDCASE

F3=Prev block  F5=User points  F6=Cancel pending moves  F23=More options
F7=Find        F8=Bookmark    F9=Parameters       F24=More keys

MA a MW 05/002
I902 - Session successfully started
```

# Each Program has the same Parameter Interface

```
Session A - [24 x 80]
File Edit View Communication Actions Window Help

Op: 5/16/11 19:26:30

EDIT FUNCTION PARAMETERS
Function name. . : SRC Standard PGM Call      Type : Execute user source
Received by file : Post Gen Processes         Acpth: *NONE

? File/*FIELD      Access path/Field/Array    Passed   Seq   Pgm   Par
- Post Gen Processes Retrieval index         FLD      1    Ctx   Ctx
- *Arrays          PGM Parameters             FLD      2
- *FIELD           *Return code                FLD      3
-
-
-
-
-
-
-
-
-
-

Values
FLD: One parameter per field
RCD: One parameter for all fields
KEY: One parameter for key fields only

SEL: Z-Parameter details  X-Object details  D-Delete parameter  N-Narrative
F3=Exit  F5=Reload  F22=File locks  F23=More options

MA a MW 07/003
I902 - Session successfully started
```

# Parameters

```
Session A - [24 x 80]
File Edit View Communication Actions Window Help

Op: 5/16/11 19:26:50

EDIT FUNCTION PARAMETER DETAILS
Function name. . : SRC Standard PGM Call      Type : Execute user source
Received by file : Post Gen Processes         Array: PGM Parameters
Parameter (file) : *Arrays                    Passed as: FLD

? Field          Usage  Role  Flag error
- Library        I
- Source File    I
- Object         I
- PGMD Mod Type  I
- Object Type    I
- Model Library  I
- Web Option Library  I

SEL: Usage: I-Input, O-Output, B-Both, N-Neither, D-Drop.
      Role: R-Restrict, M-Map, V-Vary length, P-Position. Error: E-Flag Error.
F3=Exit

MA a MW 08/005
I902 - Session successfully started
```

# Source Code to call each Process

```
Session A - [27 x 132]
File Edit View Communication Actions Window Help

Columns . . . : 6 100 Edit CRPTOOLGEN/QRPGLESRC
SEU=> TML0UFR

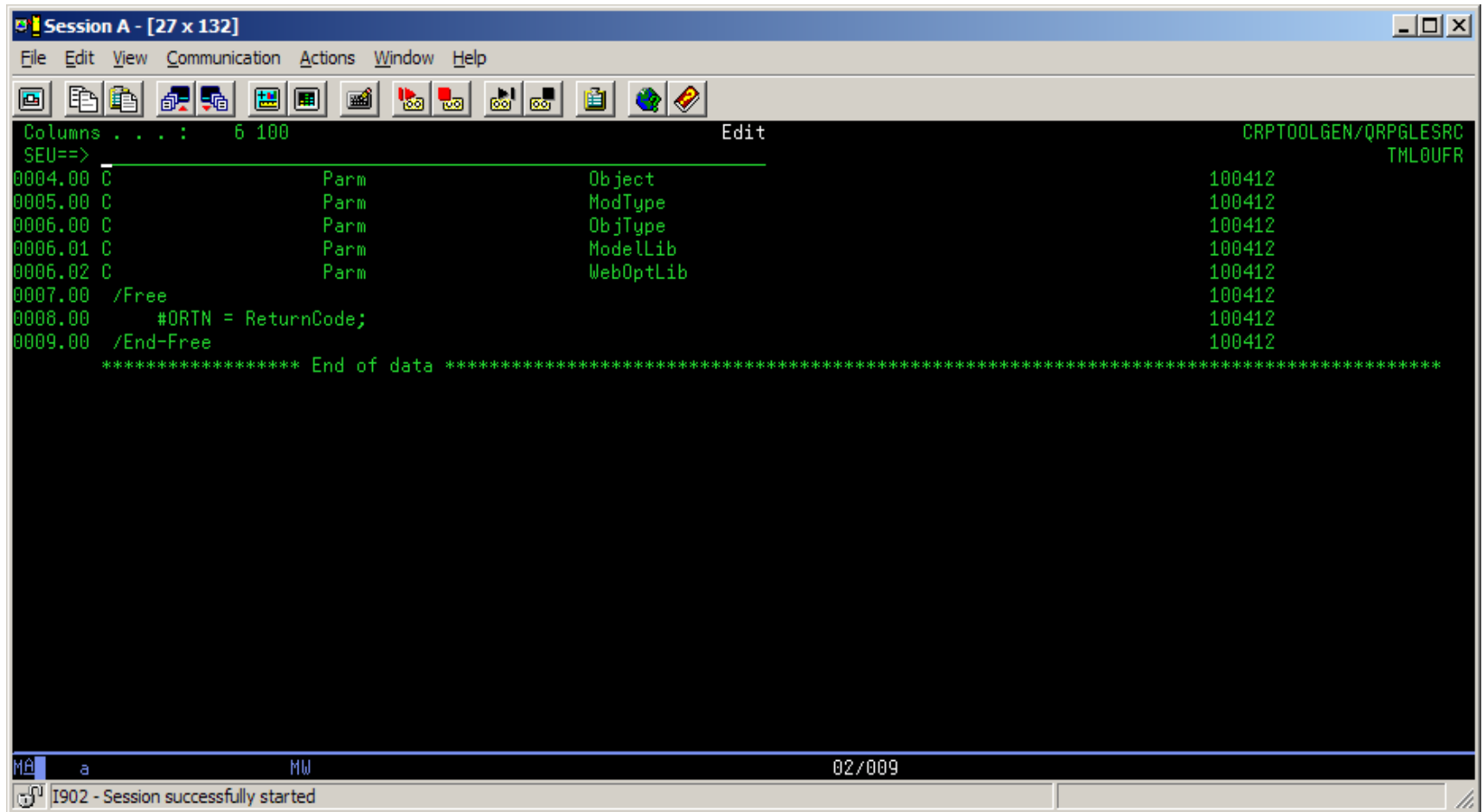
***** Beginning of data *****
0000.01 DPgmName S 10 100412
0000.02 DLibrary S 10 100412
0000.03 DSourceFile S 10 100412
0000.04 DObject S 10 100412
0000.05 DModellib S 10 100412
0000.06 DWebOptLib S 10 100412
0000.07 DModType S 3 100412
0000.08 DObjType S 8 100412
0000.09 DReturnCode S 7 100412
0000.10 /Free 100412
0000.11 PgmName = #IF9VN; 100412
0000.12 Library = #IGJCD; 100412
0000.13 SourceFile = #IB6VN; 100412
0000.14 Object = #IGKCD; 100412
0000.15 ModType = #I04ST; 100412
0000.16 ObjType = #IEDST; 100412
0000.17 Modellib = #IFTVN; 100412
0000.18 WebOptLib = #IGAVN; 100412
0000.19 /End-Free 100412
0001.00 C Call PgmName 100412
0001.01 C Parm ReturnCode 100412
0002.00 C Parm Library 100412
0003.00 C Parm SourceFile 100412

(C) COPYRIGHT IBM CORP. 1981, 2007.

MA a MW 02/009
I902 - Session successfully started
```



# Source Code to call each Process



The screenshot shows a SAP SEU editor window titled "Session A - [27 x 132]". The menu bar includes File, Edit, View, Communication, Actions, Window, and Help. The toolbar contains various icons for file operations and development tools. The main text area displays the following source code:

```
Columns . . . : 6 100                                Edit                                CRPTOOLGEN/QRPGLESRC
SEU=>                                                    TML0UFR
0004.00 C          Parm          Object                100412
0005.00 C          Parm          ModType               100412
0006.00 C          Parm          ObjType               100412
0006.01 C          Parm          ModelLib              100412
0006.02 C          Parm          WebOptLib              100412
0007.00 /Free                                           100412
0008.00      #ORTN = ReturnCode;                        100412
0009.00 /End-Free                                       100412
***** End of data *****
```

The status bar at the bottom shows "a" in the left field, "MW" in the middle field, and "02/009" in the right field. A message bar at the very bottom indicates "I902 - Session successfully started".

# DDLGEN Post Processing function

- The DDL Generation function that gets called checks to see if the passed object type is a file, and if so, finds the Object Attribute.
- If it is a PF or LF, then the process of generating the DDL is called.

# DDLGEN Post Compile Process

The screenshot displays the SAP CRPTOOLMDL editor window. The title bar reads "Session A - [24 x 80]". The menu bar includes "File", "Edit", "View", "Communication", "Actions", "Window", and "Help". The toolbar contains various icons for file operations and editing. The main text area shows the "EDIT ACTION DIAGRAM" for "CRPTOOLMDL Post Gen Processes". The "FIND=>" function is active. The diagram content includes a block for "EXT DDLGEN Processing" which is expanded to show a CASE statement for file types and a sub-block for "EXT Build DDL - Post-Gen DDL Exceptions". The status bar at the bottom shows "MA a MW" and the date "07/002". A message bar at the very bottom indicates "I902 - Session successfully started".

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help

EDIT ACTION DIAGRAM
FIND=>
I(C,I,S)F=Insert construct
I(A,E,Q,*,+,-,=,=A)F=Insert action
> EXT DDLGEN Processing
--
> ** If a *FILE, see if PF/LF and go to DDL Processing...
.-CASE
.-PAR.Object Type is File
  USP Object Attribute - Post Gen Processes *
  > ** Gen DDL if PF/LF....
  .-CASE
  - c1 OR c2
    - c1: LCL.Object Attribute is Physical File
    - c2: LCL.Object Attribute is Logical File
  -
  EXT Build DDL - Post-Gen DDL Exceptions *
  -ENDCASE
.-ENDCASE
--

F3=Prev block F5=User points F6=Cancel pending moves F23=More options
F7=Find F8=Bookmark F9=Parameters F24=More keys

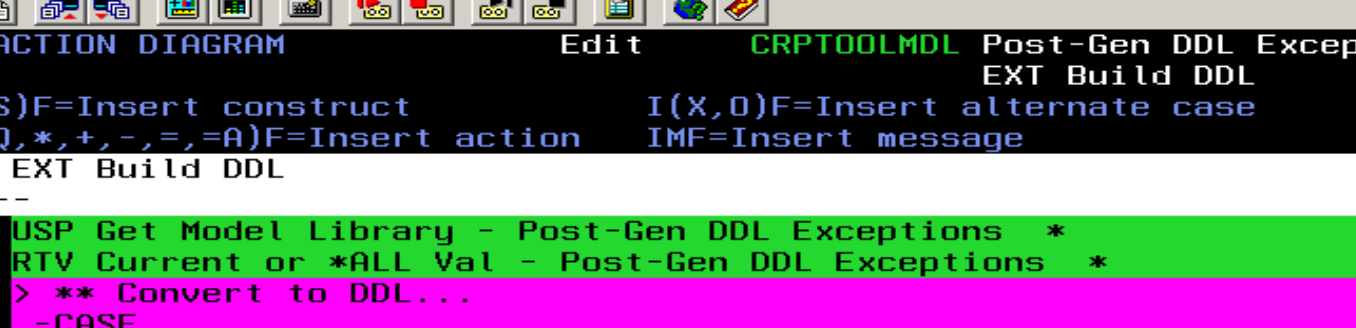
MA a MW 07/002
I902 - Session successfully started

```

# Building the DDL

- Earlier on I noted that there were some exceptions that could not be converted to DDL Index's.
- The DDL Build process checks a file to see if the current file that is being processed should be converted to DDL.
- As long as there is no exception record, the DDL Build process continues.

## Build DDL



```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
EDIT ACTION DIAGRAM Edit CRPTOOLMDL Post-Gen DDL Exceptions
FIND=> EXT Build DDL
I(C,I,S)F=Insert construct I(X,O)F=Insert alternate case
I(A,E,Q,*,+,-,=,=A)F=Insert action IMF=Insert message
> EXT Build DDL
--
. USP Get Model Library - Post-Gen DDL Exceptions *
. RTV Current or *ALL Val - Post-Gen DDL Exceptions *
. > ** Convert to DDL...
. -CASE
. -PAR.Exception Status is Convert to DDL
. LCL.File Type = SUBSTRING(PAR.2E File Name,CND.Seven,CND.One)
. ...** PF or LF...
. USP Generate DDL - Post-Gen DDL Exceptions *
. > ** Run the SQL Create Statement...
. -CASE
. - c1 OR c2
. | - c1: LCL.File Type is Physical
. | - c2: LCL.File Type is Logical
. | -
. USP Run SQL Stmt - Post-Gen DDL Exceptions *
F3=Prev block F5=User points F6=Cancel pending moves F23=More options
F7=Find F8=Bookmark F9=Parameters F24=More keys
MA a MW 15/002
I902 - Session successfully started

```

# Adding the long field names

- The Generated DDL is then parsed, and each field is given a FOR COLUMN value.
- The field name is derived from the 2E Model Name for the field
- Over the years there we have lots of fields that contain invalid characters for SQL field names
- We built an exception table for the internal DDS name for fields that caused us problems
- We also removed invalid characters during the parsing process

# Some examples

```
Session B - [27 x 132]
File Edit View Communication Actions Window Help

Display Physical File Member
File . . . . . : TMB1REP      Library . . . . . : CRPDEVUTL
Member . . . . . : TMB1REP      Record . . . . . : 1
Control . . . . . :           Column . . . . . : 1
Find . . . . . :

*...+...1...+...2...+...3...+
CEPC      Allowed_Outs_Pct
CDPC      Expected_Outs_Pct
JNST      Quantity_Conversion_850
JPST      Quantity_Conversion_852
JOST      Quantity_Conversion_855
I7ST      Item_Code_Conversion_852
I8ST      Item_Code_Conversion_855
KCST      Processing_Status_850
KEST      Processing_Status_852
KDST      Processing_Status_855
NYST      UPC_Code_Conversion_850
NZST      UPC_Code_Conversion_852
NOST      UPC_Code_Conversion_855
KGST      Simulation_Status_852
B4ST      Data_Transmitted_850_855
N2ST      Status_Field_852
OCST      Quantity_Qualifier_852
JFCD      Case_Item_Code_852

More...

F3=Exit  F12=Cancel  F19=Left  F20=Right  F24=More keys

MA  b                                04/023
I902 - Session successfully started
```

# Resulting DDL

```
Session B - [27 x 132]
File Edit View Communication Actions Window Help

Columns . . . : 1 80 Browse CBTEMPGEN/SQLSRC
SEU=> UUA2REP

***** Beginning of data *****
0001.00 DROP TABLE CBTEMPGEN/UUA2REP ; 110518
0003.00 110518
0005.00 CREATE TABLE CBTEMPGEN/UUA2REP ( 110518
0006.00 Customer_Code FOR COLUMN 110518
0007.00 A2ADCD CHAR(6) CCSID 37 NOT NULL DEFAULT '' , 110518
0008.00 Customer_Name FOR COLUMN 110518
0009.00 A2AETX CHAR(25) CCSID 37 NOT NULL DEFAULT '' , 110518
0010.00 Customer_Address FOR COLUMN 110518
0011.00 A2AFTX CHAR(25) CCSID 37 NOT NULL DEFAULT '' , 110518
0012.00 Customer_Zip FOR COLUMN 110518
0013.00 A2QSTX CHAR(5) CCSID 37 NOT NULL DEFAULT '' ) 110518
0015.00 RCDfmt FA2REEW ; 110518
0017.00 110518
0019.00 LABEL ON TABLE CBTEMPGEN/UUA2REP 110518
0021.00 IS 'Customer DDS Physical file' ; 110518
0023.00 110518
0025.00 LABEL ON COLUMN CBTEMPGEN/UUA2REP 110518
0027.00 ( A2ADCD IS 'Customer Code' , 110518
0029.00 A2AETX IS 'Customer Name' , 110518
0031.00 A2AFTX IS 'Customer Address' , 110518
0033.00 A2QSTX IS 'Customer Zip' ) ; 110518
0035.00 110518
0037.00 LABEL ON COLUMN CBTEMPGEN/UUA2REP 110518

(C) COPYRIGHT IBM CORP. 1981, 2007.

MA b 02/009
I902 - Session successfully started
```



# DSPFFD on the file

```

Session B - [27 x 132]
File Edit View Communication Actions Window Help

Display Spooled File

File . . . . . : QPDSPFFD                                     Page/Line 1/26
Control . . . . . :                                          Columns 1 - 130
Find . . . . . :
*.....1.....2.....3.....4.....5.....6.....7.....8.....9.....0.....1.....2.....3
Field Level Information
  Data      Field  Buffer  Buffer      Field  Column
  Type      Length Length Position  Usage  Heading
A2ADCD     CHAR      6      6         1    Both  Customer
                                         Code

  Field text . . . . . : Customer Code
  Alternative name . . . . . :
    CUSTOMER_CODE
  Default value . . . . . :

  Coded Character Set Identifier . . . . . : 37
  Data      Field  Buffer  Buffer      Field  Column
  Type      Length Length Position  Usage  Heading
A2AETX     CHAR     25     25         7    Both  Customer Name

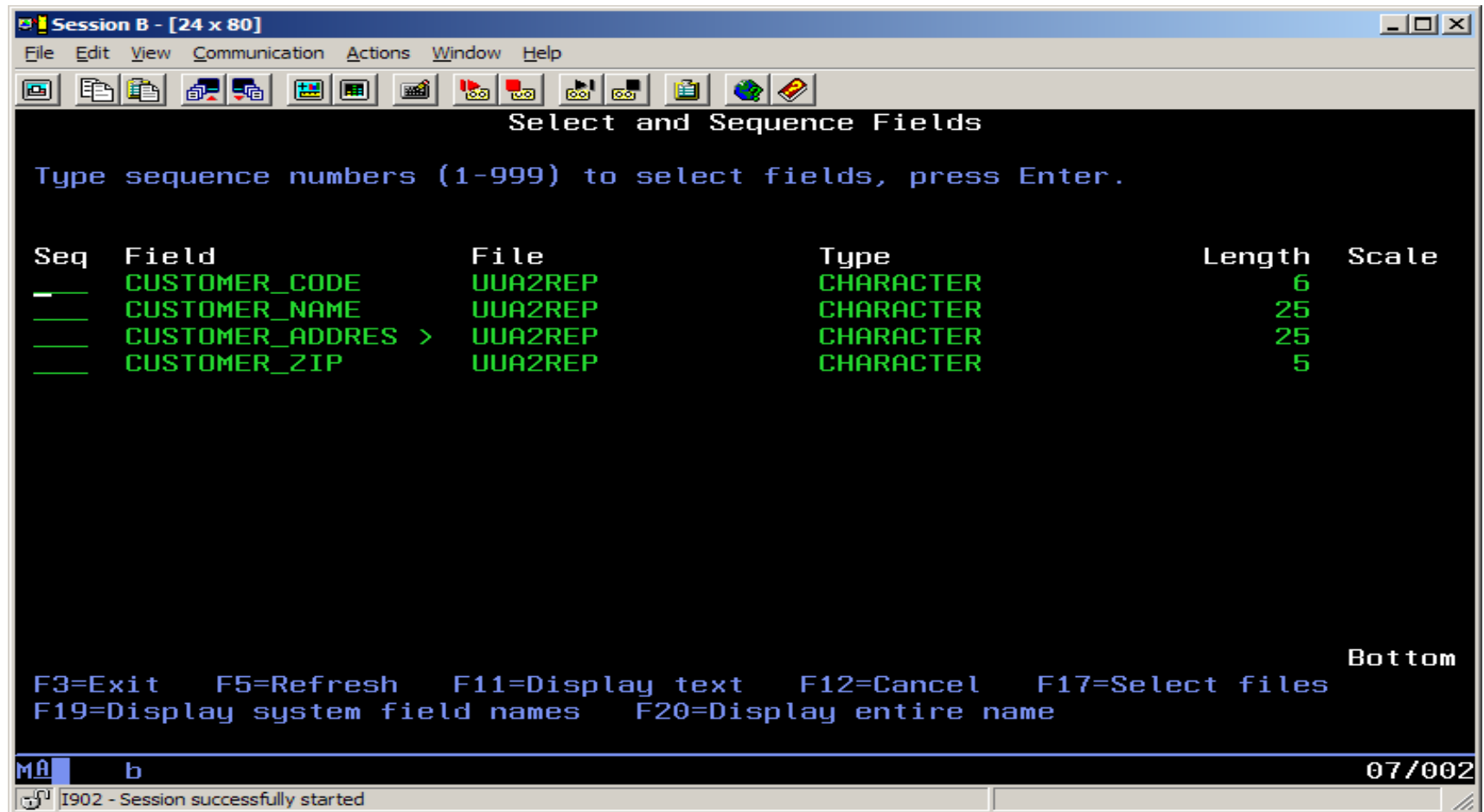
  Field text . . . . . : Customer Name
  Alternative name . . . . . :
    CUSTOMER_NAME
  Default value . . . . . :

More...

F3=Exit  F12=Cancel  F19=Left  F20=Right  F24=More keys

MA  b                                     03/022
I902 - Session successfully started
  
```

# Fields in SQL



# Summary

- So, we now have DDL TABLE and INDEX source being generated from our 2E Model
- The resulting objects have the same format level identifier that the DDS versions had, and require no recompiles
- We have long field names visible to SQL that are based on the Model field names
- Our DB2 Web Query users also see long field names

# Questions?