Table of Contents

Home * Chess * Position * **Extended Position Description**

Like FEN, **Extended Position Description** (**EPD**) describes a chess position. Unlike FEN, **EPD** is designed to be expandable by the addition of new operations. **EPD** was developed by John Stanback and Steven Edwards. Its first implementation is in Stanback's chessplaying program Zarkov. Steven Edwards specified the **EPD** standard for computer chess applications as part of the Portable Game Notation [1].

EPD Syntax

One EPD string or record consists of one text line of variable length composed of four fields separated by a space character followed by zero or more operations. The four data fields, which describe the position, are common with the FEN-Specification.

Terminal and none terminal symbols *of a variant of* BNF *below are embedded in ' ' resp. .*

```
<EPD> ::=  <Piece Placement>
       ' ' <Side to move>
       ' ' <Castling ability>
       ' ' <En passant target square>
      {' ' <operation>}
```

Piece Placement

The Piece Placement is determined rank by rank in big-endian *order, that is starting at the 8th rank down to the first rank. Each rank is separated by the terminal symbol '/' (slash). One rank, scans piece placement in* little-endian *file-order from the A to H.*
A decimal digit counts consecutive empty squares, the pieces are identified by a single letter from standard English names for chess pieces as used in the Algebraic chess notation*. Uppercase letters are for white pieces, lowercase letters for black pieces.*

```
<Piece Placement> ::= <rank8>'/'<rank7>'/'<rank6>'/'<rank5>'/'<rank4>'
/'<rank3>'/'<rank2>'/'<rank1>
<ranki>       ::= [<digit17>]<piece> {[<digit17>]<piece>} [<digit17>]
| '8'
<piece>       ::= <white Piece> | <black Piece>
<digit17>     ::= '1' | '2' | '3' | '4' | '5' | '6' | '7'
<white Piece> ::= 'P' | 'N' | 'B' | 'R' | 'Q' | 'K'
<black Piece> ::= 'p' | 'n' | 'b' | 'r' | 'q' | 'k'
```

Side to move

Side to move is one lowercase letter for either White ('w') or Black ('b').

```
<Side to move> ::= {'w' | 'b'}
```

Castling ability

If neither side can castle, the symbol '-' is used, otherwise each of four individual castling rights *for king and queen castling for both sides are indicated by a sequence of one to four letters.*

```
<Castling ability> ::= '-' | ['K'] ['Q'] ['k'] ['q'] (1..4)
```

En passant target square

The en passant *target square is specified after a double push of a pawn, no matter whether an en passant capture is really possible or not. Other moves than double pawn pushes imply the symbol '-' for this FEN field.*

```
<En passant target square> ::= '-' | <epsquare>
<epsquare>   ::= <fileLetter> <eprank>
<fileLetter> ::= 'a' | 'b' | 'c' | 'd' | 'e' | 'f' | 'g' | 'h'
<eprank>     ::= '3' | '6'
```

Operations

```
<operation> ::= <opcode> {' '<operand>} ';'
<opcode>    ::= <letter> {<letter> | <digit> | '_'} (up to 14)
<operand>   ::= <stringOperand>
              | <sanMove>
              | <unsignedOperand>
              | <integerOperand>
              | <floatOperand>

<stringOperand>  ::= '"' {<char>} '"'

<sanMove>        ::= <PieceCode> [<Disambiguation>] <targetSquare> [<p
romotion>] ['+'|'#']
                   | <castles>
<castles>        ::= 'O-O' | 'O-O-O' (upper case O, not zero)
<PieceCode>      ::= '' | 'N' | 'B' | 'R' | 'Q' | 'K'
<Disambiguation> ::= <fileLetter> | <digit18>
<targetSquare>   ::= <fileLetter> <digit18>
<fileLetter> ::= 'a' | 'b' | 'c' | 'd' | 'e' | 'f' | 'g' | 'h'
<promotion>      ::=  '=' <PiecePromotion>
<PiecePromotion> ::= 'N' | 'B' | 'R' | 'Q'

<unsignedOperand>::= <digit19> { <digit> } | '0'
<integerOperand> ::= ['-' | '+'] <unsignedIntegerOperand>
<floatOperand>   ::= <integerOperand> '.' <digit> {<digit>}
<digit18> ::= '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8'
<digit19> ::= '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'
<digit>   ::= '0' | <digit19>
```

Opcode mnemonics

- ***acd*** *analysis count* depth [2]
- ***acn*** *analysis count* nodes
- ***acs*** *analysis count seconds*
- ***am*** *avoid move(s)*
- ***bm*** *best move(s)*
- ***c0*** *comment (primary, also* ***c1*** *though* ***c9****)*
- ***ce*** centipawn *evaluation*
- ***dm*** *direct mate fullmove count*
- ***draw_accept*** *accept a draw offer*
- ***draw_claim*** *claim a draw*
- ***draw_offer*** *offer a draw*
- ***draw_reject*** *reject a draw offer*
- ***eco*** Encyclopedia of Chess Openings *opening code*
- ***fmvn*** *fullmove number*
- ***hmvc*** halfmove clock
- ***id*** *position identification*
- ***nic*** New In Chess *opening code*
- ***noop*** *no operation*
- ***pm*** *predicted move*
- ***pv*** *predicted variation*
- ***rc*** *repetition count*
- ***resign*** *game resignation*
- ***sm*** *supplied move*
- ***tcgs*** *telecommunication game selector*
- ***tcri*** *telecommunication receiver identification*
- ***tcsi*** *telecommunication sender identification*
- ***v0*** *variation name (primary, also* ***v1*** *though* ***v9****)*

See also

- Chess Artist *by* Ferdinand Mosca
- Forsyth-Edwards Notation *(FEN)*
- Portable Game Notation *(PGN)*

Forum Posts

- EPD examples: Bratko-Kopec test suite *by* Steven J. Edwards, CCC, *June 15, 1998 »* Bratko-Kopec Test
- EPD format *by* Stefan Meyer-Kahlen, CCC, *November 07, 2000*
- Question about EPD *by* Aaron Tay, CCC, *February 20, 2001*

- XBoard and epd tournament *by* Vlad Stamate, CCC, *January 31, 2010* » Chess Engine Communication Protocol, Engine Testing
- What's wrong with this EPD? *by* Jouni Uski, CCC, *March 20, 2011*
- epd multipv *by* J. Wesley Cleveland, CCC, *July 28, 2015* » Principal Variation
- Test epd for Linux ? *by Jean Arthuin,* CCC, *March 25, 2016* » Linux, STS, XBoard
- FEN - Flipper for Windows *by* Matthias Gemuh, CCC, *May 17, 2017* » Color Flipping, FEN
- how to create a labeled epd from pgn? *by* Erin Dame, CCC, *December 02, 2017* » Texel's Tuning Method, PGN

External Links

- Standard: Portable Game Notation Specification and Implementation Guide *16.2: EPD by* Steven Edwards
- EPD to HTML/ASCII Diagram Converter *by* Manfred Rosenboom [3]
- 40H Chess Tools and Utilities *by* Norm Pollock » Portable Game Notation [4] [5] [6]
- Chess-Tools/epd2uci.py at master · Mk-Chan/Chess-Tools · GitHub *by* Manik Charan » Python, python-chess

References

1. ^ Standard: Portable Game Notation Specification and Implementation Guide *16.2: EPD by* Steven Edwards
2. ^ EPD format *by* Stefan Meyer-Kahlen, CCC, *November 07, 2000*
3. ^ EPD2diag *hosted by* Ed Schröder
4. ^ An important message to users of 40H utility tools *by* Norm Pollock, CCC, *December 13, 2015*
5. ^ Re: 40H chess downloads has moved *by* Norm Pollock, CCC, *March 12, 2016*
6. ^ Moved "40H" tools/utilities to a new URL *by*Norm Pollock, CCC, *December 21, 2016*

What links here?

Page	*Date Edited*
More Links	
Aaron Tay	*Mar 24, 2014*
Aquarium	*Dec 4, 2016*
Arena	*Mar 16, 2018*
Belofte	*Jun 29, 2017*
Board Representation	*Dec 11, 2017*
Bratko-Kopec Test	*Jun 1, 2015*
CCR One Hour Test	*Jun 14, 2016*
Chess Academy	*Jan 15, 2014*
Chess Position	*Sep 10, 2017*
Color Flipping	*May 17, 2017*
Data	*Nov 26, 2017*
David Forsyth	*May 2, 2011*

Page	*Date Edited*
Diagrams	*Sep 10, 2017*
Dictionary	*Aug 24, 2017*
Eigenmann Endgame Test	*Jun 1, 2017*
Erin Dame	*Dec 2, 2017*
Extended Position Description	*Dec 2, 2017*
FCP	*Jun 15, 2013*
Ferdinand Mosca	*Jul 28, 2017*
Forsyth-Edwards Notation	*Sep 10, 2017*
Graphics Programming	*Dec 22, 2017*
Kaufman Test	*Jan 25, 2013*
Kvetka	*Mar 28, 2016*
LCT II	*Feb 25, 2018*
Linux	*Jan 21, 2018*
LittleBlitzer	*Jan 25, 2017*
Manik Charan	*Mar 10, 2018*
Matthias Gemuh	*Jan 21, 2018*
Patzer	*Apr 5, 2017*
Perl	*Oct 25, 2016*
Piece Recognition	*Sep 8, 2017*
Portable Game Notation	*Jan 11, 2018*
Principal variation	*Dec 4, 2017*
Programming	*Dec 16, 2017*
Python	*Jan 31, 2018*
python-chess	*Nov 4, 2017*
Scid	*Apr 15, 2017*
Silent but deadly	*Jun 17, 2015*
Stefan Meyer-Kahlen	*Jun 19, 2017*
Steven Edwards	*Aug 26, 2017*

Up one Level