

```

// TYPESET.WCM — Replace ASCII quotation, apostrophe, and dash marks
// with typesetter-style marks,
// using characters from WP charsets 1 and 4, and arrange
// for correct hyphenation at dashes.
//
// "foo" becomes "foo"
// 'foo' becomes 'foo'
// dad's becomes dad's
// foo--bar becomes foo—bar
//
// For simplicity and speed—so that we needn't step through the doc
// character by character but can
// instead use Search—we do the job on quote marks in two passes: one for
// ASCII doublequotes and
// one for ASCII singlequotes/apostrophes.
//
// WARNING: this screws up on nested quotation marks of the same kind (e.g.
// doublequotes
// within doublequotes). This could be fixed, but the speed penalty makes it
// probably not worthwhile.
//
// Author: Richard Reiner <rreiner@nexus.yorku.ca>
//
// Revision history
// 12 Jun 1992 RjR Initial version
// 16 Jun 1992 RjR Logic to distinguish between
// singlequotes and apostrophes;
// loop unrolling for speed.
// 25 Jun 1992 RjR Added conversion of dashes; changed
// name to TYPESET.WCM.
// 16 Oct 1992 RjR Added extra passes to insert [HyphSRt]
// codes after Emdashes and Endashes.
//

```

APPLICATION(WP;WPWP;Default)

```

// *****
// Main
//
CALL(Init@)
CALL(ProcessDq@)
CALL(ProcessSq@)
CALL(ProcessDash@)
CALL(ProcessHyphSRt@)
CALL(End@)
// NOTREACHED

```

QUIT

```
// *****
// Init@
//
// Initialization
//
LABEL(Init@)
ONERROR(End@)
ONCANCEL(End@)
GetWPData(MacroVariable:RevCodes;SystemVariable:RevealCodesActive!)
SelectMode(Off!)
RevealCodes(Off!)
MacroStatusPrompt(Off!)
Display(Off!)
RETURN

// *****
// ProcessDq@
//
// Convert all doublequotes.
//
LABEL(ProcessDq@)
PosDocTop()
ONNOTFOUND(DqDone@)
// Unwind the following loop by one, so we can use SearchNext() in the body.
// This cuts almost 50%
// off the macro's execution time.
SearchText(SearchString:"""";SearchDirection:Forward!;SearchScope:Extended!)
DeleteCharPrevious()
TypeChar(CharacterSet:4;CharacterOffset:30)
InQuote:=True
// The only way out of this loop is the ONNOTFOUND trap
WHILE(True)
    SearchNext()
    DeleteCharPrevious()
    IF (InQuote)
        TypeChar(CharacterSet:4;CharacterOffset:31)
        InQuote:=False
    ELSE
        TypeChar(CharacterSet:4;CharacterOffset:30)
        InQuote:=True
    ENDIF
ENDWHILE
```

```

LABEL(DqDone@)
    RETURN

    // *****
    // ProcessSq@
    //
    // Convert singlequotes and apostrophes.
    //
    LABEL(ProcessSq@)
PosDocTop()
ONNOTFOUND(SqDone@)
// Unwind the following loop by one, so we can use SearchNext() in the body.
    This cuts almost 50%
// off the macro's execution time.
SearchText(SearchString:"";SearchDirection:Forward!;SearchScope:Extended!)
// If the chars to the left and right of the singlequote are both alpha, this must
    be an apostrophe.
PosCharPrevious()
GetWPData(MacroVariable:LeftCh;SystemVariable:LeftChar!)
PosCharNext()
GetWPData(MacroVariable:RightCh;SystemVariable:RightChar!)
IF (((LeftCh >= "a") and (LeftCh <= "z")) or ((LeftCh >= "A") and (LeftCh <=
    "Z"))) and
(((RightCh >= "a") and (RightCh <= "z")) or ((RightCh >= "A") and (RightCh
    <= "Z")))
    // It's an apostrophe
    DeleteCharPrevious()
    TypeChar(CharacterSet:1;CharacterOffset:9)
    InQuote:=False
ELSE
    // It's a singlequote
    DeleteCharPrevious()
    TypeChar(CharacterSet:4;CharacterOffset:27)
    InQuote:=True
ENDIF
// The only way out of this loop is the ONNOTFOUND trap
WHILE(True)
    SearchNext()
    // If the chars to the left and right of the singlequote are both alpha, this
    must be an apostrophe.
    PosCharPrevious()
    GetWPData(MacroVariable:LeftCh;SystemVariable:LeftChar!)
    PosCharNext()
    GetWPData(MacroVariable:RightCh;SystemVariable:RightChar!)
    IF (((LeftCh >= "a") and (LeftCh <= "z")) or ((LeftCh >= "A") and

```

```

(LeftCh <= "Z")) and
  (((RightCh >= "a") and (RightCh <= "z")) or ((RightCh >= "A") and
(RightCh <= "Z"))))
  // It's an apostrophe
  DeleteCharPrevious()
  TypeChar(CharacterSet:1;CharacterOffset:9)
ELSE
  // It's a singlequote
  DeleteCharPrevious()
  IF (InQuote)
    TypeChar(CharacterSet:4;CharacterOffset:28)
    InQuote:=False
  ELSE
    TypeChar(CharacterSet:4;CharacterOffset:27)
    InQuote:=True
  ENDIF
ENDIF
ENDWHILE
LABEL(SqDone@)
RETURN

```

```

// *****
// ProcessDash@
//
// Convert dashes. This one's easy.
//
LABEL(ProcessDash@)
PosDocTop()
SearchReplace(SearchString:"";
  SearchDirection:Forward!;
  ReplacementScope:Extended!;
  ReplacementString:"—";
  ReplacementAction:ReplaceAll!)
RETURN

```

```

// *****
// ProcessHyphSRt@
//
// Insert [HyphSRt] codes after Emdashes and Endashes. This one's easy too.
//
LABEL(ProcessHyphSRt@)
// Do Emdashes
PosDocTop()
SearchReplace(SearchString:"—";

```

```
        SearchDirection:Forward!;
        ReplacementScope:Extended!;
        ReplacementString:"—";
        ReplacementAction:ReplaceAll!)
// Do Endashes
PosDocTop()
SearchReplace(SearchString:"-";
        SearchDirection:Forward!;
        ReplacementScope:Extended!;
        ReplacementString:"-";
        ReplacementAction:ReplaceAll!)
RETURN

// *****
// End@
//
// Wrap things up and terminate
//
LABEL(End@)
IF(RevCodes=True)
    RevealCodes(On!)
ENDIF
PosDocTop()
QUIT
```