



OpenOffice.org's Documentation of the

Microsoft Excel File Format

Excel Versions 2, 3, 4, 5, 95, 97, 2000, XP, 2003

Author	Daniel Rentz ✉ mailto:dr@openoffice.org 🌐 http://sc.openoffice.org
License	Public Documentation License
Contributors	Andrew C. Oliver ✉ mailto:acoliver@apache.org 🌐 http://jakarta.apache.org/poi/index.html – POIFS, additional records, fixes Yves Hiltpold, James J. Keene, Sami Kuhmonen, John Marmion, Alexander Mavrin, Mike Salter, Charles Wyble – additional records, fixes
Other sources	Hyperlinks to Wikipedia (🌐 http://www.wikipedia.org) for various extended information
Mailing list	✉ mailto:dev@sc.openoffice.org Subscription ✉ mailto:dev-subscribe@sc.openoffice.org
Download	PDF 🌐 http://sc.openoffice.org/excelfileformat.pdf XML 🌐 http://sc.openoffice.org/excelfileformat.sxw
Project started	2001-Jun-29
Last change	2005-Mar-24
Revision	1.39

Contents

1	Introduction	6
1.1	License Notices	6
1.2	Abstract	7
1.3	Byte Order	9
2	Document Structure	10
2.1	File Format Versions	10
2.2	Structure of the Document File	11
3	Common Record Substructures	12
3.1	Structure of a Record	12
3.2	Formatting Runs	13
3.3	Byte Strings (BIFF2-BIFF7)	13
3.4	Unicode Strings (BIFF8)	14
3.5	RGB Colours	17
3.6	RK Values	17
3.7	Error Codes	17
3.8	Constant Cached Value Lists	18
3.9	URL Encoding	20
3.10	Line Styles for Cell Borders (BIFF3-BIFF8)	22
3.11	Patterns for Cell Background Area (BIFF3-BIFF8)	22
3.12	Cell Attributes (BIFF2)	23
3.13	Cell Range Addresses and Lists	24
4	Formulas	25
4.1	Common Formula Structure	25
4.2	Token Classes	28
4.3	Cell Addresses in Tokens	32
4.4	Token Overview	36
4.5	Unary Operator Tokens	40
4.6	Binary Operator Tokens	41
4.7	Function Operator Tokens	45
4.8	Constant Operand Tokens	47
4.9	Operand Tokens	50
4.10	Control Tokens	61
4.11	Built-In Sheet Functions	69

5	Worksheet/Workbook Structure	73
5.1	Worksheet/Workbook Stream	74
5.2	Record Order in Substreams	76
5.3	Worksheet Calculation Settings Block	81
5.4	Page Settings Block	81
5.5	Worksheet View Settings Block	82
5.6	Cell Formatting	83
5.7	Cell Table	84
5.8	Array Formulas, Shared Formulas	90
5.9	Multiple Operation Tables	91
5.10	Internal and External References	92
5.11	Shared String Table (BIFF8)	104
5.12	Conditional Formatting Table (BIFF8)	106
5.13	Hyperlink Table (BIFF8)	106
5.14	Data Validity Table (BIFF8)	106
5.15	AutoFilter, Advanced Filter	107
5.16	Scenarios	107
5.17	Web Queries (BIFF8)	107
5.18	Worksheet/Workbook Protection	108
5.19	File Protection	110
6	Worksheet/Workbook Records	113
6.1	Overview, Ordered by Record Identifier	113
6.2	Overview, Ordered by Record Names	117
6.3	Overview, Ordered by BIFF Versions	120
6.4	ARRAY	124
6.5	BACKUP	125
6.6	BITMAP	125
6.7	BLANK	126
6.8	BOF – Beginning of File	127
6.9	BOOKBOOL	129
6.10	BOOLERR	129
6.11	BOTTOMMARGIN	130
6.12	BOUNDSHEET	130
6.13	BULTINFMTCOUNT	131
6.14	CALCCOUNT	131
6.15	CALCMODE	131
6.16	CF – Conditional Formatting	132
6.17	CODEPAGE	136
6.18	COLINFO	137
6.19	COLUMNDEFAULT	138
6.20	COLWIDTH	138
6.21	CONDFMT	139
6.22	CONTINUE	140
6.23	COUNTRY	141
6.24	CRN	142
6.25	DATEMODE	142
6.26	DBCELL	143
6.27	DCONREF – Data Consolidation Reference	143
6.28	DEFAULTROWHEIGHT	143
6.29	DEFCOLWIDTH	144
6.30	DELTA	144
6.31	DIMENSIONS	145
6.32	DSF – Double Stream File	145
6.33	DV – Data Validity Settings	146
6.34	DVAL – Data Validity List	147
6.35	EFONT	147
6.36	EOF – End of File	148
6.37	EXTERNCOUNT	148

6.38	EXTERNNAME	149
6.39	EXTERNSHEET	152
6.40	EXTSST – Extended SST	153
6.41	FILEPASS	154
6.42	FILESHARING	156
6.43	FONT	157
6.44	FOOTER	159
6.45	FORMAT	160
6.46	FORMULA	162
6.47	GCW – Global Column Width	164
6.48	GRIDSET	165
6.49	GUTS	165
6.50	HCENTER	165
6.51	HEADER	166
6.52	HIDEOBJ	167
6.53	HLINK – Hyperlink	168
6.54	HORIZONTALPAGEBREAKS	171
6.55	INDEX	172
6.56	INTEGER	174
6.57	ITERATION	174
6.58	IXFE – Index to XF	174
6.59	LABEL	175
6.60	LABELRANGES	175
6.61	LABELSST	176
6.62	LEFTMARGIN	176
6.63	MERGEDCELLS	176
6.64	MULBLANK – Multiple BLANK	177
6.65	MULRK – Multiple RK	177
6.66	NAME	178
6.67	NOTE	181
6.68	NUMBER	182
6.69	OBJECTPROTECT	182
6.70	PALETTE	183
6.71	PANE	184
6.72	PASSWORD	184
6.73	PHONETIC	185
6.74	PRECISION	185
6.75	PRINTGRIDLINES	186
6.76	PRINTHEADERS	186
6.77	PROTECT	187
6.78	QUICKTIP	187
6.79	RANGEPROTECTION	187
6.80	REFMODE	188
6.81	RIGHTMARGIN	188
6.82	RK	188
6.83	ROW	189
6.84	RSTRING	191
6.85	SAVERECALC	191
6.86	SCENPROTECT	192
6.87	SCL	192
6.88	SELECTION	192
6.89	SETUP	193
6.90	SHEETHDR	196
6.91	SHEETLAYOUT	196
6.92	SHEETPROTECTION	197
6.93	SHEETSOFFSET	197
6.94	SHRFMLA – Shared Formula	198
6.95	SORT	199
6.96	SST – Shared String Table	200
6.97	STANDARDWIDTH	200

6.98	STRING	201
6.99	STYLE	202
6.100	SUPBOOK – External Workbook	203
6.101	TABLEOP – Multiple Operation Table	205
6.102	TABLEOP2 – Multiple Operation Table	206
6.103	TOPMARGIN	206
6.104	UNCALCED	207
6.105	USESELFS	207
6.106	VCENTER	208
6.107	VERTICALPAGEBREAKS	208
6.108	WINDOW1	209
6.109	WINDOW2	210
6.110	WINDOWPROTECT	212
6.111	WRITEACCESS	212
6.112	WRITEPROT	212
6.113	WSBOOL	213
6.114	XCT – CRN Count	213
6.115	XF – Extended Format	214
7	Drawing Objects, Escher Layer	219
8	Charts	220
9	PivotTables	221
10	Change Tracking	222

1 Introduction

1.1 License Notices

1.1.1 Public Documentation License Notice

The contents of this Documentation are subject to the Public Documentation License Version 1.0 (the "License"); you may only use this Documentation if you comply with the terms of this License. A copy of the License is available at <http://www.openoffice.org/licenses/PDL.html>.

The Original Documentation is "OpenOffice.org's Documentation of the Microsoft Excel File Format".

The Initial Writer of the Original Documentation is Sun Microsystems, Inc., Copyright © 2003. All Rights Reserved.

See title page for Author contact and Contributors.

All Trademarks are properties of their respective owners.

1.1.2 Wikipedia

Wikipedia Disclaimer: http://en.wikipedia.org/wiki/Wikipedia:General_disclaimer

1.2 Abstract

This document contains a description of the binary file format of Microsoft Excel, including all available Excel versions (for Windows) from Excel 2.1 to the current Excel 2003.

This project has been started in June 2001 and is still in progress. At several places the remark “2do” indicates an incomplete section of the documentation.

1.2.1 Project Status

Chapter	Contents	Status	
1	Introduction	Common information	Done
2	Document Structure	Raw document structure overview	Done
3	Common Record Substructures	Parts of records for common use	Done
4	Formulas	- Structure of RPN token arrays - Detailed description of all tokens	Done In progress
5	Worksheet/Workbook Structure	Abstract description of complex features represented by several records	In progress
6	Worksheet/Workbook Records	Detailed description of all records of the worksheet/workbook stream	In progress
7	Drawing Objects, Escher Layer		Not started
8	Charts		Not started
9	PivotTables		Not started
10	Change Tracking		Not started

1.2.2 Used Terms, Symbols, and Formatting

• References

A reference to another chapter is symbolised by a little arrow: →1.1.

• Examples

An example is indented and marked with a light-grey border.

This is an example.

• Important Passages

Text passages with important information contain a leading exclamation mark.

! This is an important passage.

• Numbers and Strings

Numerical values are shown in several number systems:

Number system	Marking	Example
Decimal	None	1234
Hexadecimal	Trailing “H”	1234 _H
Binary	Trailing “2”	1001 ₂

Constant strings are enclosed in quotation marks. They may contain specific values (control characters, unprintable characters). These values are enclosed in angle brackets.

Example of a string containing a control character: “abcdef<01_H>ghij”.

• Record Listings

A record listing shows a bundle of records in the required order. A dark grey rectangle stands for a single record, a light grey rectangle stands for a group of records. Either this is a group representing a specific feature and is referred with the *Record Group Name*, or it is a group of various unspecified records that do not matter in this context.

RECORD NAME	Comments
<i>Record Group Name</i>	Comments

• Record Content Listings

- Data offsets enclosed in square brackets indicate record content that may be omitted (the remark “optional” may point out this).
- The term “*Not used*” means: Ignore the data on import and write zero bytes on export. The same applies for unmentioned bits in bit fields.
- The term “*Unknown*” describes data fields with fixed but unknown contents. On export these fields have to be written as shown.
- At several places a variable is introduced, which represents the value of this field for later use. In most common cases this is a field containing a size value, which is used later in the “Size” column of the record content listing. An example can be found in →3.1.

• Algorithm Listings

Algorithms given in pseudo-code are shown with a light-grey background.

```
ALGORITHM Example_Algorithm
```

- 1) Command 1
- 2) Command 2

Notation conventions used in algorithms:

Notation	Description
<code>command1 ; command2</code>	Two commands in one line, first execute <code>command1</code> , then <code>command2</code>
<code>var ← value</code>	The value <code>value</code> is assigned to the variable <code>var</code>
<code>JUMP x)</code>	Continue with line <code>x)</code> in the algorithm
<code>RETURN [value]</code>	Returns value <code>value</code> if specified, otherwise returns without a return value
<code>IF cond THEN command</code>	Execute <code>command</code> only, if condition <code>cond</code> evaluates to <code>true</code>
<code>AND</code>	Binary AND operation
<code>OR</code>	Binary OR operation
<code>XOR</code>	Binary XOR (exclusive or) operation
<code>= <> < > <= >=</code>	Comparison operators for conditional execution
<code>array[]</code>	An array consisting of equal typed elements
<code>array[0]</code>	The first element of the array <code>array[]</code> (arrays are used zero-based)

1.3 Byte Order

All data items containing more than one byte are stored using the Little-Endian method¹. That means the least significant byte is stored first and the most significant byte last. This applies for all data types like 16-bit integers, 32-bit integers, floating-point values and Unicode characters.

Example: The 32-bit integer value `13579BDFH` is converted into the byte sequence `DFH 9BH 57H 13H`.

¹ For more information see <http://en.wikipedia.org/wiki/Endianness>.

2 Document Structure

2.1 File Format Versions

The Excel file format is named BIFF (Binary Interchange File Format). The following table shows which Excel version writes which file format.

Excel version	BIFF version	Document type	File type
Excel 2.1	BIFF2	Worksheet	Stream
Excel 3.0	BIFF3	Worksheet	Stream
Excel 4.0	BIFF4S	Worksheet	Stream
	BIFF4W	Workbook	Stream
Excel 5.0	BIFF5	Workbook	Compound Document
Excel 7.0 (Excel 95)	BIFF7	Workbook	Compound Document
Excel 8.0 (Excel 97)	BIFF8	Workbook	Compound Document
Excel 9.0 (Excel 2000)	BIFF8	Workbook	Compound Document
Excel 10.0 (Excel XP)	BIFF8X	Workbook	Compound Document
Excel 11.0 (Excel 2003)	BIFF8X	Workbook	Compound Document

The oldest file format BIFF2 has of course the most restrictions. Excel 4.0 supports single worksheet files (BIFF4S) or files containing a bundle of sheets, called a workbook (BIFF4W). In this document the name “BIFF4” refers to both BIFF4S and BIFF4W.

The current format BIFF8 contains major changes towards older BIFF versions, for instance the handling of Unicode strings. Excel XP writes BIFF8X, an extension of BIFF8. Excel 97 and Excel 2000 can read BIFF8X, except the new features added with Excel XP. In this document the name “BIFF8” refers to BIFF8X too, if it is not mentioned explicitly.

2.2 Structure of the Document File

2.2.1 Structure of a Stream File (BIFF2-BIFF4)

Files stored in the BIFF versions BIFF2 to BIFF4 contain all records for a sheet or a BIFF4W workbook in one simple stream. The records are arranged sequential, they are never embedded in other records.

2.2.2 Structure of a Compound Document File (BIFF5-BIFF8)

An Excel workbook with several sheets (from BIFF5 on) is stored using the compound document file format (also known as “OLE2 storage file format” or “Microsoft Office compatible storage file format”). It contains several streams for different types of data. A complete documentation of the format of compound document files can be found at <http://sc.openoffice.org/compdocfileformat.pdf>.

The following table lists names of possible streams.

Stream name	Contents
Book	BIFF5/BIFF7 workbook stream (→5.1.3)
Workbook	BIFF8 workbook stream (→5.1.3)
<05 _H >SummaryInformation	Document settings
<05 _H >DocumentSummaryInformation	Document settings
Ctls	Formatting of form controls
User Names	User names in shared workbooks (→10)
Revision Log	Change tracking log stream (→10)

It is possible to create substorages like subdirectories in a file system, for instance for the PivotTable streams. These storages contain substreams itself.

Storage name	Contents
LNKxxxxxxxx	Storage for a linked OLE object (→7)
MBDxxxxxxxx	Storage for an embedded OLE object (→7)
_SX_DB_CUR	Pivot cache storage. The streams contain cached values for PivotTables (→9).
_VBA_PROJECT_CUR	Visual BASIC project storage

In all streams the records are arranged sequential, they are never embedded in other records².

² Exception in BIFF8: The Escher object stream is split and embedded in several MSODRAWING records (→7).

3 Common Record Substructures

This chapter contains information about the record structure itself and about basic substructures which are part of several records, for instance strings or error codes.

3.1 Structure of a Record

In an Excel data stream the data is divided into several records. Each record contains specific data for the various features of Excel. The common structure of a record is described in the following table.

Offset	Size	Contents	
0	2	Identifier	} Record header
2	2	Size of the following data (<u>s.z</u>)	
4	<u>s.z</u>	Data	

The maximum size of the record data is limited. If the size of the record data exceeds the given limits, one or more CONTINUE records (→6.22) will be added. Inside of a CONTINUE record the data of the previous record continues as usual.

In the following descriptions only the record data without the headers is shown. All offsets are relative to the beginning of the record data and not to the entire record. The contents of most of the records differ from version to version. This will be described in separate tables. A few older records are replaced in newer BIFF versions. Excel does not write these old records in new BIFF versions anymore.

3.2 Formatting Runs

Formatting runs describe the character formatting of strings. A formatting run contains the index of a character and the index of a font in the font buffer. The font is used to format the indexed character and the following characters, until the string ends or another formatting run follows.

Formatting run, BIFF2-BIFF7:

Offset	Size	Contents
0	1	First formatted character (zero-based)
1	1	Index to FONT record (→6.43)

Formatting run, BIFF8:

Offset	Size	Contents
0	2	First formatted character (zero-based)
2	2	Index to FONT record (→6.43)

3.3 Byte Strings (BIFF2-BIFF7)

All Excel file formats up to BIFF7 contain simple byte strings. The byte string consists of the length of the string followed by the character array. The length is stored either as 8-bit value or as 16-bit value, depending on the current record. The string is not zero-terminated. The character array is encoded with the character encoding specified in the CODEPAGE record (→6.17).

Offset	Size	Contents
0	1 or 2	Length of the string (character count, <u>l</u> <u>n</u>)
1 or 2	<u>l</u> <u>n</u>	Character array (8-bit characters)

3.4 Unicode Strings (BIFF8)

From BIFF8 on, strings are always stored using UTF-16LE³ text encoding. The character array is a sequence of 16-bit values⁴. Additionally it is possible to use a compressed format, which omits the high bytes of all characters, if they are all zero.

The following tables describe the standard format of the entire string, but in many records the strings differ from this format. This will be mentioned separately. It is possible (but not required) to store Rich-Text formatting information and Asian phonetic information inside a Unicode string. This results in four different ways to store a string. The character array is not zero-terminated.

3.4.1 Contents of a Unicode String

The string consists of the character count (as usual an 8-bit value or a 16-bit value), option flags, the character array and optional formatting information. If the string is empty, sometimes the option flags field will not occur. This is mentioned at the respective place.

Offset	Size	Contents												
0	1 or 2	Length of the string (character count, <u>ln</u>)												
1 or 2	1	Option flags: <table border="1" data-bbox="416 904 1402 1243"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>01_H</td> <td>Character compression (<u>ccomp.r</u>): 0₂ = Compressed (8-bit characters) 1₂ = Uncompressed (16-bit characters)</td> </tr> <tr> <td>2</td> <td>04_H</td> <td>Asian phonetic settings (<u>phonetic</u>): 0₂ = Does not contain Asian phonetic settings 1₂ = Contains Asian phonetic settings</td> </tr> <tr> <td>3</td> <td>08_H</td> <td>Rich-Text settings (<u>richtext</u>): 0₂ = Does not contain Rich-Text settings 1₂ = Contains Rich-Text settings</td> </tr> </tbody> </table>	Bit	Mask	Contents	0	01 _H	Character compression (<u>ccomp.r</u>): 0 ₂ = Compressed (8-bit characters) 1 ₂ = Uncompressed (16-bit characters)	2	04 _H	Asian phonetic settings (<u>phonetic</u>): 0 ₂ = Does not contain Asian phonetic settings 1 ₂ = Contains Asian phonetic settings	3	08 _H	Rich-Text settings (<u>richtext</u>): 0 ₂ = Does not contain Rich-Text settings 1 ₂ = Contains Rich-Text settings
Bit	Mask	Contents												
0	01 _H	Character compression (<u>ccomp.r</u>): 0 ₂ = Compressed (8-bit characters) 1 ₂ = Uncompressed (16-bit characters)												
2	04 _H	Asian phonetic settings (<u>phonetic</u>): 0 ₂ = Does not contain Asian phonetic settings 1 ₂ = Contains Asian phonetic settings												
3	08 _H	Rich-Text settings (<u>richtext</u>): 0 ₂ = Does not contain Rich-Text settings 1 ₂ = Contains Rich-Text settings												
[2 or 3]	2	(optional, only if <u>richtext</u> =1) Number of Rich-Text formatting runs (<u>rt</u>)												
[var.]	4	(optional, only if <u>phonetic</u> =1) Size of Asian phonetic settings block (in bytes, <u>sz</u>)												
var.	<u>ln</u> or 2· <u>ln</u>	Character array (8-bit characters or 16-bit characters, dependent on <u>ccomp.r</u>)												
[var.]	4· <u>rt</u>	(optional, only if <u>richtext</u> =1) List of <u>rt</u> formatting runs (→3.2)												
[var.]	<u>sz</u>	(optional, only if <u>phonetic</u> =1) <i>Asian Phonetic Settings Block</i> (→3.4.2)												

³ For more information see <http://en.wikipedia.org/wiki/UTF-16>.

⁴ In most cases each value corresponds to a Unicode character. Only the Unicode characters above U+FFFF are encoded with a “surrogate pair”, that are two 16-bit code values in UTF-16 (see footnote 3).

3.4.2 Asian Phonetic Settings Block

Asian phonetic text⁵ (Ruby) can be used to provide extended phonetic information for specific characters or words. It appears above the regular text (or to the right of vertical text), and can refer to single characters, groups of characters, or entire words.

Offset	Size	Contents												
0	2	Unknown identifier 0001 _H												
2	2	Size of the following data ($10 + 2 \cdot \underline{ln} + 6 \cdot \underline{np}$)												
4	2	Index to FONT record (→6.43) used for the Asian phonetic text												
6	2	Additional settings for the Asian phonetic text:												
		<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>1-0</td> <td>0003_H</td> <td>Type of Japanese phonetic text (<u>t</u>.<u>type</u>): 00₂ = Katakana (narrow) 10₂ = Hiragana 01₂ = Katakana (wide)</td> </tr> <tr> <td>3-2</td> <td>000C_H</td> <td>Alignment of all portions of the Asian phonetic text (<u>a</u>.<u>lign</u>): 00₂ = Not specified (Japanese only) 10₂ = Centered 01₂ = Left (Top for vertical text) 11₂ = Distributed</td> </tr> <tr> <td>5-4</td> <td>0030_H</td> <td>11₂ (always set)</td> </tr> </tbody> </table>	Bit	Mask	Contents	1-0	0003 _H	Type of Japanese phonetic text (<u>t</u> . <u>type</u>): 00 ₂ = Katakana (narrow) 10 ₂ = Hiragana 01 ₂ = Katakana (wide)	3-2	000C _H	Alignment of all portions of the Asian phonetic text (<u>a</u> . <u>lign</u>): 00 ₂ = Not specified (Japanese only) 10 ₂ = Centered 01 ₂ = Left (Top for vertical text) 11 ₂ = Distributed	5-4	0030 _H	11 ₂ (always set)
Bit	Mask	Contents												
1-0	0003 _H	Type of Japanese phonetic text (<u>t</u> . <u>type</u>): 00 ₂ = Katakana (narrow) 10 ₂ = Hiragana 01 ₂ = Katakana (wide)												
3-2	000C _H	Alignment of all portions of the Asian phonetic text (<u>a</u> . <u>lign</u>): 00 ₂ = Not specified (Japanese only) 10 ₂ = Centered 01 ₂ = Left (Top for vertical text) 11 ₂ = Distributed												
5-4	0030 _H	11 ₂ (always set)												
8	2	Number of portions the Asian phonetic text is broken into (<u>np</u>). If <u>np</u> = 0, the Asian phonetic text refers to the entire cell text.												
10	2	Total length of the following Asian phonetic text (number of characters, <u>ln</u>)												
12	2	Repeated total length of the text												
14	2· <u>ln</u>	Character array of Asian phonetic text, no Unicode string header, always 16-bit characters. Note: If <u>ln</u> = 0, this field is <i>not</i> empty but contains 0000 _H .												
14+2· <u>ln</u>	6· <u>np</u>	List of <u>np</u> structures that describe the position of each portion in the main text. Each structure contains the following fields:												
		<table border="1"> <thead> <tr> <th>Offset</th> <th>Size</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>2</td> <td>First character in the Asian phonetic text of this portion (<u>c</u>.<u>pa</u>)</td> </tr> <tr> <td>2</td> <td>2</td> <td>First character of the main text belonging to this portion (<u>c</u>.<u>pm</u>)</td> </tr> <tr> <td>4</td> <td>2</td> <td>Number of characters in main text belonging to this portion (<u>c</u>.<u>cm</u>)</td> </tr> </tbody> </table>	Offset	Size	Contents	0	2	First character in the Asian phonetic text of this portion (<u>c</u> . <u>pa</u>)	2	2	First character of the main text belonging to this portion (<u>c</u> . <u>pm</u>)	4	2	Number of characters in main text belonging to this portion (<u>c</u> . <u>cm</u>)
Offset	Size	Contents												
0	2	First character in the Asian phonetic text of this portion (<u>c</u> . <u>pa</u>)												
2	2	First character of the main text belonging to this portion (<u>c</u> . <u>pm</u>)												
4	2	Number of characters in main text belonging to this portion (<u>c</u> . <u>cm</u>)												

⁵ For more information see http://en.wikipedia.org/wiki/Ruby_characters.

Example: Japanese word Tokyo (東京) with added hiragana (とうきょう)⁶. The following examples show the contents of the important fields of the *Asian Phonetic Settings Block*.

Example 1: Hiragana centered over the entire word:

とうきょう
東京

`type` = 10₂ (hiragana)
`align` = 10₂ (centered)
`np` = 0 (no portions, hiragana refers to entire text)
`ln` = 5 (length of entire hiragana text)
No portion structures

Example 2: Hiragana left-aligned per character:

とう きょう
東京

`type` = 10₂ (hiragana)
`align` = 01₂ (left-aligned)
`np` = 2 (hiragana split into 2 portions)
`ln` = 5 (length of entire hiragana text)
Portion #1: `cpa` = 0 (start with 1st hiragana character); `cpm` = 0; `ccm` = 1 (attach to 1st character in main text)
Portion #2: `cpa` = 2 (start with 3rd hiragana character); `cpm` = 1; `ccm` = 1 (attach to 2nd character in main text)

⁶ Example taken from http://en.wikipedia.org/wiki/Ruby_characters.

3.5 RGB Colours

Colours are represented in RGB mode (red/green/blue).

Offset	Size	Contents
0	1	Red component
1	1	Green component
2	1	Blue component
3	1	Not used

3.6 RK Values

An RK value is an encoded integer or floating-point value. RK values have a size of 4 bytes and are used to decrease file size for floating-point values.

Structure of an RK value (32-bit value), BIFF3-BIFF8:

Bit	Mask	Contents
0	00000001 _H	0 = Value not changed 1 = Value is multiplied by 100
1	00000002 _H	0 = Floating-point value 1 = Signed integer value
31-2	FFFFFFFC _H	Encoded value

If bit 1 is cleared, the encoded value represents the 30 most significant bits of an IEEE 754 floating-point value (64-bit double precision). The 34 least significant bits must be set to zero. If bit 1 is set, the encoded value represents a signed 30-bit integer value. To get the correct integer, the encoded value has to be shifted right arithmetically by 2 bits. If bit 0 is set, the decoded value (both integer and floating-point) must be divided by 100 to get the final result.

Examples:

RK value	Type	Div 100	Encoded value	Decoded value	Result
3FF00000 _H	float	no	3FF00000 _H	3FF0000000000000 _H = 1.0	1.0
3FF00001 _H	float	yes	3FF00000 _H	3FF0000000000000 _H = 1.0	0.01
004B5646 _H	integer	no	004B5644 _H	0012D591 _H = 1234321	1234321
004B5647 _H	integer	yes	004B5644 _H	0012D591 _H = 1234321	12343.21

3.7 Error Codes

If the calculation of a formula results in an error or any other action fails, Excel sets a specific error code. These error codes are used for instance in cell records and formulas.

Error code	Error value	Description
00 _H	#NULL!	Intersection of two cell ranges is empty
07 _H	#DIV/0!	Division by zero
0F _H	#VALUE!	Wrong type of operand
17 _H	#REF!	Illegal or deleted cell reference
1D _H	#NAME?	Wrong function or range name
24 _H	#NUM!	Value range overflow
2A _H	#N/A!	Argument or function not available

3.8 Constant Cached Value Lists

Sometimes it is needed to store a list of constant values, for instance for cached values of external cell references (record CRN, →6.24), cached DDE link results (record EXTERNNAME, →6.38), or for constant arrays in formulas (token tArray, →4.8.7). The value list may contain floating-point values, strings, Boolean values, or error codes. The number of values the list contains is stored before in the respective record or token. The list may be linear (record CRN), or an 2-dimensional array (record EXTERNNAME and token tArray, see 3.8.2).

3.8.1 Constant Cached Values

• Empty Value

Offset	Size	Contents
0	1	00 _H (Identifier for an empty constant)
1	8	Not used

• Number

Offset	Size	Contents
0	1	01 _H (Identifier for a numerical constant)
1	8	IEEE 754 floating-point value (64-bit double precision)

• String Value

Offset	Size	Contents
0	1	02 _H (Identifier for a string constant)
1	var.	BIFF2-BIFF7: Byte string, 8-bit string length (→3.3) BIFF8: Unicode string, 16-bit string length, option flags occur always (→3.4)

• Boolean value

Offset	Size	Contents
0	1	04 _H (Identifier for a Boolean constant)
1	1	0 = FALSE, 1 = TRUE
2	7	Not used

• Error Value

Offset	Size	Contents
0	1	10 _H (Identifier for an error constant)
1	1	Error code (→3.7)
2	7	Not used

3.8.2 Constant Cached Value Array

The array contains the dimensions (width and height), followed by the list of all constant cached values. It is used in the record EXTERNNAME (→6.38) and the token tArray (→4.8.7).

2-dimensional cached value array, BIFF2-BIFF7:

Offset	Size	Contents
0	1	Number of columns (<u>nc</u>). The value 0 represents 256 columns.
1	2	Number of rows (<u>nr</u>)
3	var.	List of <u>nc</u> · <u>nr</u> cached values (→3.8.1)

2-dimensional cached value array, BIFF8:

Offset	Size	Contents
0	1	Number of columns decreased by 1 (<u>nc</u>)
1	2	Number of rows decreased by 1 (<u>nr</u>)
3	var.	List of (<u>nc</u> +1)·(<u>nr</u> +1) cached values (→3.8.1)

3.9 URL Encoding

3.9.1 Encoded File URLs

The intention of encoding file URLs is to make them more platform independent. Encoded URLs occur in the records EXTERNSHEET (BIFF2-BIFF7, →6.39) or SUPBOOK (BIFF8, →6.100), and DCONREF (→6.27).

The first character of the URL is used to determine the type of encoding. In Unicode strings (BIFF8) this could be a 16-bit value.

First character	BIFF2-BIFF4	BIFF5/BIFF7	BIFF8
01 _H	Encoded URL follows	Encoded URL follows	Encoded URL follows
02 _H	Reference to the current sheet (nothing will follow)	Reference to the current sheet (nothing will follow)	Reference to a sheet in the own document (sheet name follows)
03 _H	Not used	Reference to a sheet in the own document (sheet name follows)	Not used
04 _H	Not used	Reference to the own workbook, sheet is unspecified (nothing will follow)	Not used
others	Not encoded. This is already the first character of the URL.		

Inside of the encoded URL there can occur several control characters.

Control character	Meaning
01 _H	An MS-DOS drive letter will follow, or “@” and the server name of a UNC path
02 _H	Start path name on same drive as own document
03 _H	End of subdirectory name
04 _H	Start path name in parent directory of own document (may occur repeatedly)
05 _H	Unencoded URL. Followed by the length of the URL (1 byte), and the URL itself.
06 _H	Start path name in installation directory of Excel
08 _H	Macro template directory in installation directory of Excel
09 _H	Sheet in the same workbook (BIFF4W)

If a sheet name follows the file name in the encoded URL (BIFF4W-BIFF8), the file name (but not the file path) will be enclosed in brackets. Note that in SUPBOOK records (BIFF8) table names do not occur and therefore the file names are not enclosed in brackets.

Examples for BIFF2-BIFF4S (own document is saved as “C:\path\own.xls”):

Formula	Encoded filename
=own.xls!A1	<02 _H >
=ext.xls!A1	<01 _H >ext.xls
= 'sub\ext.xls' !A1	<01 _H >sub<03 _H >ext.xls
= '\ext.xls' !A1	<01 _H ><02 _H >ext.xls
= '\sub\ext.xls' !A1	<01 _H ><02 _H >sub<03 _H >ext.xls
= '\sub\sub2\ext.xls' !A1	<01 _H ><02 _H >sub<03 _H >sub2<03 _H >ext.xls
= 'D:\sub\ext.xls' !A1	<01 _H ><01 _H >Dsub<03 _H >ext.xls
= '..\sub\ext.xls' !A1	<01 _H ><04 _H >sub<03 _H >ext.xls
= '\\pc\sub\ext.xls' !A1	<01 _H ><01 _H >@pc<03 _H >sub<03 _H >ext.xls
= 'http://www.example.org/ext.xls' !A1	<01 _H ><05 _H ><1E _H >http://www.example.org/ext.xls (the length of the URL (30 = 1E _H) follows the 05 _H byte)

Examples for BIFF4W internal references (all formulas are located on “Sheet1”):

Formula	Encoded filename
=Sheet1!A1	<02 _H >
=Sheet2!A1	<01 _H ><09 _H >Sheet2

Examples for BIFF5/BIFF7 internal references (all formulas are located on “Sheet1”):

Formula	Encoded filename
=Sheet1!A1	<02 _H >
=Sheet2!A1	<03 _H >Sheet2
=NonExistentSheet!A1	<04 _H >

Examples for BIFF8 internal references (for example in record DCONREF):

Formula	Encoded filename
=Sheet2!A1	<02 _H >Sheet2

Examples for BIFF4W-BIFF8 external references (own document is saved as “C:\path\own.xls”):

Formula	Encoded filename
= [ext.xls]Sheet1!A1	<01 _H >[ext.xls]Sheet1
= 'sub\[ext.xls]Sheet1'!A1	<01 _H >sub<03 _H >[ext.xls]Sheet1
= '\[ext.xls]Sheet1'!A1	<01 _H ><02 _H >[ext.xls]Sheet1
= '\sub\[ext.xls]Sheet1'!A1	<01 _H ><02 _H >sub<03 _H >[ext.xls]Sheet1
= '\sub\sub2\[ext.xls]Sheet1'!A1	<01 _H ><02 _H >sub<03 _H >sub2<03 _H >[ext.xls]Sheet1
= 'D:\sub\[ext.xls]Sheet1'!A1	<01 _H ><01 _H >Dsub<03 _H >[ext.xls]Sheet1
= '..\sub\[ext.xls]Sheet1'!A1	<01 _H ><04 _H >sub<03 _H >[ext.xls]Sheet1
= '\\pc\sub\[ext.xls]Sheet1'!A1	<01 _H ><01 _H >@pc<03 _H >sub<03 _H >[ext.xls]Sheet1
= 'http://www.example.org/[ext.xls]Sheet1'!A1	<01 _H ><05 _H ><26 _H >http://www.example.org/[ext.xls]Sheet1 (the length of the URL (38 = 26 _H) follows the 05 _H byte)

3.9.2 Encoded DDE and OLE Object Links














A DDE link contains the name of the server application and the DDE topic (usually the URL of the document). An OLE object link contains a class name and the URL of the document. In both cases the names are stored in one string, separated by the control character 03_H. The URLs are *not* encoded.

Example: A document contains a DDE link to the SO/OOo Calc document “sub\example.sxc” and an OLE object link to the bitmap file “sub\example.bmp”.

Link	Encoded document name
DDE	soffice<03 _H >sub\example.sxc
OLE object	Package<03 _H >sub\example.bmp





































3.10 Line Styles for Cell Borders (BIFF3-BIFF8)

These line styles are used to define cell borders. The styles 08_H to 0D_H are available in BIFF8 only.

Index	Style	Sample	Index	Style	Sample
00 _H	No line		The following for BIFF8 only:		
01 _H	Thin		08 _H	Medium dashed	
02 _H	Medium		09 _H	Thin dash-dotted	
03 _H	Dashed		0A _H	Medium dash-dotted	
04 _H	Dotted		0B _H	Thin dash-dot-dotted	
05 _H	Thick		0C _H	Medium dash-dot-dotted	
06 _H	Double		0D _H	Slanted medium dash-dotted	
07 _H	Hair				

3.11 Patterns for Cell Background Area (BIFF3-BIFF8)

From BIFF3 on, the cell background area may contain a pattern. Pattern colour and pattern background colour are defined separately. In the following table black is used as pattern colour and white as pattern background colour.

Index	Pattern	Sample	Index	Pattern	Sample
00 _H		No pattern	0A _H		
01 _H			0B _H		
02 _H			0C _H		
03 _H			0D _H		
04 _H			0E _H		
05 _H			0F _H		
06 _H			10 _H		
07 _H			11 _H		
08 _H			12 _H		
09 _H					

The following table shows how a pattern is used with the correct colour indexes.

Cell format	Pattern	Pattern colour index	Background colour index
No background (shows system window background)	00 _H	System window text (not used)	System window background
Red background (solid)	01 _H	Red	System window text (not used)
Red background with thin horizontal blue lines	0B _H	Blue	Red
Red background with thin horizontal lines, automatic colour	0B _H	System window text	Red
No background, thin horizontal blue lines	0B _H	Blue	System window background
No background, thin horizontal lines, automatic colour	0B _H	System window text	System window background

The description of the PALETTE record (→6.70) contains how the special system colours are used. Note the behaviour of solid coloured backgrounds, where pattern 01_H is used in conjunction with the pattern colour.

3.12 Cell Attributes (BIFF2)

All cell records in BIFF2 contain a cell attribute field with a size of 3 bytes. They contain an index to an XF record (→6.115) and some repeated contents of the referenced XF record. The XF index field has a size of only 6 bits, so the index range is 0...63. If a real XF index greater than 62 is used, the XF index field always contains the value 63, and an IXFE record (→6.58) occurs in front of a cell record. The IXFE record contains the correct index of the XF record. In a ROW record (→6.83) this field is not used, because there will always occur a real XF index field.

Cell attributes field (3 bytes), BIFF2:

Offset	Size	Contents																					
0	1	Cell protection and XF index:																					
		<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>5-0</td> <td>3F_H</td> <td>Index to XF record (→6.115). The value 3F_H = 63 indicates a preceding IXFE record (→6.58). Not used in ROW records (→6.83).</td> </tr> <tr> <td>6</td> <td>40_H</td> <td>1 = Cell is locked</td> </tr> <tr> <td>7</td> <td>80_H</td> <td>1 = Formula is hidden</td> </tr> </tbody> </table>	Bit	Mask	Contents	5-0	3F _H	Index to XF record (→6.115). The value 3F _H = 63 indicates a preceding IXFE record (→6.58). Not used in ROW records (→6.83).	6	40 _H	1 = Cell is locked	7	80 _H	1 = Formula is hidden									
Bit	Mask	Contents																					
5-0	3F _H	Index to XF record (→6.115). The value 3F _H = 63 indicates a preceding IXFE record (→6.58). Not used in ROW records (→6.83).																					
6	40 _H	1 = Cell is locked																					
7	80 _H	1 = Formula is hidden																					
1	1	Indexes to FORMAT and FONT records:																					
		<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>5-0</td> <td>3F_H</td> <td>Index to FORMAT record (→6.45)</td> </tr> <tr> <td>7-6</td> <td>C0_H</td> <td>Index to FONT record (→6.43)</td> </tr> </tbody> </table>	Bit	Mask	Contents	5-0	3F _H	Index to FORMAT record (→6.45)	7-6	C0 _H	Index to FONT record (→6.43)												
Bit	Mask	Contents																					
5-0	3F _H	Index to FORMAT record (→6.45)																					
7-6	C0 _H	Index to FONT record (→6.43)																					
2	1	Cell style:																					
		<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>2-0</td> <td>07_H</td> <td>XF_HOR_ALIGN – Horizontal alignment (→6.115.1)</td> </tr> <tr> <td>3</td> <td>08_H</td> <td>1 = Cell has left black border</td> </tr> <tr> <td>4</td> <td>10_H</td> <td>1 = Cell has right black border</td> </tr> <tr> <td>5</td> <td>20_H</td> <td>1 = Cell has top black border</td> </tr> <tr> <td>6</td> <td>40_H</td> <td>1 = Cell has bottom black border</td> </tr> <tr> <td>7</td> <td>80_H</td> <td>1 = Cell has shaded background</td> </tr> </tbody> </table>	Bit	Mask	Contents	2-0	07 _H	XF_HOR_ALIGN – Horizontal alignment (→6.115.1)	3	08 _H	1 = Cell has left black border	4	10 _H	1 = Cell has right black border	5	20 _H	1 = Cell has top black border	6	40 _H	1 = Cell has bottom black border	7	80 _H	1 = Cell has shaded background
Bit	Mask	Contents																					
2-0	07 _H	XF_HOR_ALIGN – Horizontal alignment (→6.115.1)																					
3	08 _H	1 = Cell has left black border																					
4	10 _H	1 = Cell has right black border																					
5	20 _H	1 = Cell has top black border																					
6	40 _H	1 = Cell has bottom black border																					
7	80 _H	1 = Cell has shaded background																					

3.13 Cell Range Addresses and Lists

In several records and formula tokens there is a common way to store absolute cell range addresses and address lists.

3.13.1 Cell Range Address

A cell range address specifies a fixed cell range in the current sheet.

Cell range address, BIFF2-BIFF7:

Offset	Size	Contents
0	2	Index to first row
2	2	Index to last row
4	1	Index to first column
5	1	Index to last column

Cell range address, BIFF8:

Offset	Size	Contents
0	2	Index to first row
2	2	Index to last row
4	2	Index to first column
6	2	Index to last column

In several cases, BIFF8 still writes the BIFF2-BIFF7 format of a cell range address (using 8-bit values for the column indexes). This will be mentioned at the respective place.

3.13.2 Cell Range Address List

A cell range address list consists of a field with the number of ranges and the list of the range addresses.

Cell range address list, BIFF2-BIFF8:

Offset	Size	Contents
0	2	Number of following cell range addresses (<u>nm</u>)
2	6· <u>nm</u> or 8· <u>nm</u>	List of <u>nm</u> cell range addresses (→3.13.1)

In several cases, BIFF8 still writes the BIFF2-BIFF7 format of a cell range address (using 8-bit values for the column indexes). This will be mentioned at the respective place.

4 Formulas

4.1 Common Formula Structure

4.1.1 Common Structure

Formulas are stored as part of a record, for instance inside of a FORMULA record or a NAME record. The common format of a formula is as follows:

Formula in BIFF2:

Offset	Size	Contents
0	1	Size of the following formula data (RPN token array, <u>s.z.</u>)
1	<u>s.z.</u>	Formula data (RPN token array)
[1+ <u>s.z.</u>]	var.	(optional) Additional data for specific tokens (→4.1.6, for example tArray token, →4.8.7)

Formula in BIFF3-BIFF8:

Offset	Size	Contents
0	2	Size of the following formula data (<u>s.z.</u>)
2	<u>s.z.</u>	Formula data (RPN token array)
[2+ <u>s.z.</u>]	var.	(optional) Additional data for specific tokens (→4.1.6, for example tArray token, →4.8.7)

Sometimes the size field is not stored directly before the RPN token array, but somewhere else. If this happens, it will be mentioned at the respective place.

4.1.2 Tokens

Tokens are the indivisible particles of a formula. There might be operators, numerical or string constants or function names. Each token contains a token identifier. Several tokens contain additional information. A token does not contain any size information, so importing and exporting tokens must be done carefully.

Common structure of a formula token, BIFF2-BIFF8:

Offset	Size	Contents
0	1	Token identifier
[1]	var.	(optional) Additional data for the token

4.1.3 Token Notation

Tokens are referred to by their name. All token names start with a small “t” (for “Token”). If a token contains additional information, it is appended to the token name in parentheses.

Examples:

The addition operator + is represented by the token tAdd. It does not contain any additional data.

The integer constant 1 is represented by the token tInt(1).

An absolute reference to cell \$A\$1 is represented by the token tRef(\$A\$1).

4.1.4 Operators

There are 3 types of operators:

- Unary operators like the minus sign that negates a value. These operators pop the topmost operand from the stack.
- Binary operators like addition or multiplication. These operators pop the two topmost operands from the stack.
- Function operators represent the sheet functions of Excel. They operate on different numbers of topmost operands on the stack. Either the function expects a fixed number of operands (for instance SIN expects one operand), or a variable number of operands given in the function token (for instance SUM is able to process 0 to 30 operands).

All operators push the (single) result of their operation back onto the stack.

4.1.5 Token Arrays

The tokens of a formula are stored in the Reverse-Polish Notation (RPN). This means, first occur all operands of an operation, followed by the respective operator. For instance the simple term 1+2 consists of the 3 tokens “1”, “+” and “2”. Written in RPN, the formula is converted to the token list “1”, “2”, “+”. During parsing such an expression, operands are pushed onto a stack. An operator pops the needed number of operands from stack, performs the operation and pushes the result back onto the stack.

Other examples for RPN token arrays:

Formula	Token array	Excel notation	Parsing result
2*4+5	2, 4, *, 5, +	tInt(2), tInt(4), tMul, tInt(5), tAdd	First, the integer constants 2 and 4 are pushed onto the stack. The * operator pops them from the stack and pushes 8. Then the constant 5 is pushed. The + operator pops 5 and 8 and pushes 13 (the final result).
2+4*5	2, 4, 5, *, +	tInt(2), tInt(4), tInt(5), tMul, tAdd	First, the integer constants 2, 4, and 5 are pushed onto the stack. The * operator pops 5 and 4 and pushes 20, the + operator pops 20 and 2 and pushes 22 (the final result).
ABS(2*-A1)	2, A1, -, *, ABS	tInt(2), tRefV(A1), tUminus, tMul, tFunc(ABS)	First, the integer constant 2 and the value from cell A1 (for example 3) are pushed onto the stack. The unary - operator (tUminus) pops the topmost value 3 from stack and pushes the negated value -3. The * operator pops -3 and 2 and pushes -6. The ABS function needs 1 parameter. It pops -6 and pushes 6 (the final result).

Example of the complete byte representation of the formula $2 * 4 + 5$.

- The RPN representation of the formula is: 2, 4, 5, *, +.
- Written in Excel token notation, this is: tInt(2), tInt(4), tInt(5), tMul, tAdd.

Offset	Size	Contents	Token name	Description
0	2	000B _H		Size of the following formula data (<u>s.z.</u>)
2	1	1E _H	tInt	} Integer value token for 2
3	2	0002 _H		
5	1	1E _H	tInt	} Integer value token for 4
6	2	0004 _H		
8	1	05 _H	tMul	Multiplication operator
9	1	1E _H	tInt	} Integer value token for 5
10	2	0005 _H		
12	1	03 _H	tAdd	Addition operator

4.1.6 Additional Token Data

A few tokens contain additional data that does not follow the token identifier, but is appended to the token array. Its size is *not* contained in the leading field containing the token array size. Affected tokens are the tArray token (→4.8.7), the tMemArea token (→4.9.4), and a few subtypes of the the tExtended token (→4.10.4). The additional data of each token is appended in the same order the tokens are located in the token array.

Example: The formula $=\{1\}+A1:A2 A2:A3+\{2\}$ contains 3 tokens with additional data: 2 tArray tokens representing the constant arrays, and a tMemArea token containing the result of the intersection operator (the cell reference A2). The detailed example below applies for BIFF8.

- The RPN representation of the formula is: {1}, A1:A2, A2:A3, “ ”, +, {2}, +.
- Written in Excel token notation, this is (the term A1:A2, A2:A3, “ ” is led by a tMemArea token): tArrayV({1}), tMemAreaV(A2), tAreaR(A1:A2), tAreaR(A2:A3), tIsect, tAdd, tArrayV({2}), tAdd.

Offset	Size	Token name	Description
0	2		Size of the following formula data (44 bytes), <i>without</i> the additional data following the last tAdd token
2	8	tArrayV	Placeholder for first constant array {1}
10	7	tMemAreaV	Constant reference subexpression follows in next 19 bytes (until and including the tIsect token)
17	9	tAreaR	Cell reference A1:A2
26	9	tAreaR	Cell reference A2:A3
35	1	tIsect	Intersection operator
36	1	tAdd	Addition operator
37	8	tArrayV	Placeholder for second constant array {2}
45	1	tAdd	Addition operator
46	12		Constant cached value array (→3.8.2) containing the values of the constant array {1}: contains width 1, height 1, value 1
58	10		Cell range address list (→3.13.2) containing the result of the reference subexpression A1:A2 A2:A3: a one-element list with the cell address A2
68	12		Constant cached value array (→3.8.2) containing the values of the constant array {2}: contains width 1, height 1, value 2

4.2 Token Classes

4.2.1 Classified Tokens

All function operators and most operand tokens exist in 3 different versions: as *reference class token*, *value class token*, and *array class token*. The token class depends on which type of data the involved operator expects. Sometimes only 1 or 2 token classes are valid for a token (for example, *array class tokens* cannot exist as *reference class tokens*, but they can exist as *value class tokens*).

- *Reference class token*: The reference address itself, independent of the cell contents.
- *Value class token*: A value (a constant, a function result, or one specific value from a dereferenced cell range).
- *Array class token*: An array of values (array of constant values, an array function result, or all values of a cell range).

The structure of the 8-bit identifier of a classified token is described in the following table.

Bit	Mask	Contents
4-0	1F _H	Basic token identifier
6-5	60 _H	01 ₂ = Reference class token (token range 20 _H -3F _H) 10 ₂ = Value class token (token range 40 _H -5F _H) 11 ₂ = Array class token (token range 60 _H -7F _H)
7	80 _H	0 ₂ (zero)

The token class is marked in the names of the tokens. The names of *reference class tokens* contain a trailing “R”, *value class tokens* contain a trailing “V”, and the names of *array class tokens* a trailing “A”.

Example: The tArea token (→4.9.3) is no specific token, but refers to the three tokens tAreaR (25_H), tAreaV (45_H), and tAreaA (65_H).

- The tAreaR token represents the cell range address itself.
- The tAreaV token represents one value in the cell range (for example the value in the current row or column).
- The tAreaA token represents the entire array of values in the cell range.

4.2.2 Operand Classes

Each operator and operand has a default token class, called operand class. For operands itself, the operand class is dependent on the data the token represents. The operand class of operators is determined from its return value. For functions (tokens tFunc, →4.7.1, and tFuncVar, →4.7.2), the operand class is dependent on the value the function returns. Classified tokens can represent their operand class, other tokens cannot.

Examples for operand classes:

Term	Operand class	Token identifier	Description
A1	Reference	tRefR	Reference to cell A1
A1 A1	Reference	tIsect	Return value of the intersection operator
INDEX (A1 , 1 , 1)	Reference	tFuncVarR	Return value of the INDEX function
2	Value	tInt	The constant 2
2+3	Value	tAdd	Return value of the addition operator
SUM (2 , 3)	Value	tFuncVarV	Return value of the SUM function
{2 , 3}	Array	tArrayA	The constant array {2 , 3}
TREND ({2 , 3})	Array	tFuncVarA	Return value of the TREND function

4.2.3 Expected Parameter Classes

Function parameters expect operands of specific operand classes. The expected token classes of all function parameters are specified in the list of built-in functions (→4.11). The result of the whole formula is handled as a parameter, called the *root level* parameter (the parameter of the equality sign). The *root level* also expects a specific token class.

Examples for function parameter classes:

- *Reference class*:
ROW (A1) (the first parameter of the function ROW expects *reference class*, ROW (1) would produce an error).
- *Value class*:
ABS (A1) (the first parameter of the function ABS expects *value class*, it dereferences the cell reference to a value).
- *Array class*:
MDETERM (A1 : C3) (the first parameter of the function MDETERM expects *array class*, it dereferences the cell range to an array of values).
- *Root level*:
=A1 (parameter is on formula root level, expected class is dependent on formula type).

4.2.4 Token Class Transformation

The final class of a token depends on the combination of the operand class of the token and the expected parameter class. Furthermore it is dependent on the type of the formula. There are 3 different types of formulas:

Formula type	Examples
Cell type formula	Cell formulas (→5.7), shared formulas (→5.8)
Array type formula	Array formulas (→5.8), conditional formatting (→5.12), data validity (→5.14)
Name type formula	Defined names (→5.10), reference lists (for example chart source range, form control links)

Token class transformation is done in several steps. All steps have to be performed for each classified token in a formula.

• Step 1: Value Operators

If the token has *reference operand class*, and if it is a direct operand of a unary or binary value operator (tAdd, tSub, tMul, tDiv, tPower, tConcat, tLT, tLE, tEQ, tGE, tGT, tNE, tUplus, tUminus, or tPercent – the binary reference operators tRange, tList, and tIsect are *not* included here), its class will be changed to *value class*, and further handling is done regarding this new class (for example it may be changed to *array class* later).

Example: In the formula =SUM (A1 , B1+1) , the cell addresses A1 and B1 are represented by tRefR tokens. Because the second tRefR token containing B1 has *reference operand class*, and it is an operand of the addition operator, it will be changed to *value class* (tRefV). As described below, in *array type formulas* this token will be changed to *array class* then, while the *reference class* of the first tRefR(A1) token is retained.

• Step 2: Forced Array Class

If the token is part of a function parameter (directly or indirectly, for example nested in another function), transformation of the token class is dependent on the expected token classes of all involved functions. For this purpose, a Boolean state “*forced array class*” is carried, and it will be updated whenever processing of a new function parameter is started. If this state is set to *true* already, nothing will be changed (the state is still *true*). If the state is *false*, it will be changed to *true*, if one of the following conditions is met:

- In *cell type formulas*: if the parameter expects *array class*, or
- In *array type formulas*: if the parameter expects *reference class* or *array class*, or
- In *name type formulas*: if the parameter expects *reference class* or *array class*, or if the function returns a value with *value class* or *array class*.

The *forced array class* state will be restored, when the entire function parameter is processed. The state starts with *false* at root level of all formula types.

As described below, the cell formula =ABS (A1) changes its tRefR token to value class (tRefV). The formula =MDETERM (ABS (A1)) behaves differently: The first parameter of the MDETERM function expects *array class*. Therefore the “*forced array class*” state is set to *true*, when the parameter (the term SUM (A1)) is started. This state is the reason that the tRefR token will be changed to *array class* (tRefA), regardless that the ABS function does not indicate this. When the parameter is completed, the *forced array class* state is restored to its old value (*false*).

• Step 3: Token Class Transformation

Token class transformation is dependent on the expected token class of the current position in the formula.

- Current position is a function parameter expecting a *reference class token* (for example the parameters of the SUM function), or it is the *root level* of a *name type formula* (for example the return value of the outer function in a defined name):
 - *Reference class tokens* are not modified.
 - *Value class tokens* will be changed dependent on further conditions. In *array type functions* and *name type functions*, or if the *forced array class* state is set, it is changed to *array class*. In all other cases (*cell type formula* without *forced array class*), *value class* is retained.
 - *Array class tokens* are not modified.

Examples for *reference class tokens*, when *reference class* is expected:

- In the formula =SUM (A1) , the tRefR token is not modified (function parameter expects *reference class*).
- In the defined name =Sheet1!A1, the tRef3dR token is not modified (root level of a *name type formula*).

Examples for *value class tokens*, when *reference class* is expected:

- In the cell formula =SUM (PI ()) , the tFuncV(PI) token is not modified (cell formula, no *forced array class*).
- In the array formula {=SUM (PI ())} , the tFuncV(PI) token is set to *array class* (*array type formula*).
- In the cell formula =MDETERM (SUM (PI ())) , the tFuncV(PI) token is set to *array class* (*forced array class* in MDETERM function parameter).

Examples for *array class tokens*, when *reference class* is expected:

- In the formula =SUM ({1}) , the tArrayA token is not modified.
- In the defined name = {1} , the tArrayA token is not modified (root level of a *name type formula*).

- Current position is a function parameter expecting a *value class token* (for example the parameter of the ABS function), or it is the *root level* of a *cell type formula* or an *array type formula* (for example the return value of the outer function in a cell formula, a shared formula, or an array formula):
 - If the *forced array class* state is set, all tokens will be changed to *array class*, otherwise to *value class*.

Examples for token transformation, when *value class* is expected without *forced array class* state:

- In the cell formula =ABS (A1), the tRefR token is changed to *value class* (tRefV), and the tFuncV(ABS) token keeps its *value class* (*root level* of a cell function).
- In the cell formula =ABS (PI ()), the tFuncV(PI) token keeps unchanged.
- In the cell formula =ABS ({ 1 }), the tArrayA token is changed to *value class* (tArrayV).

Examples for token transformation, when *value class* is expected with *forced array class* state:

- In the name formula =ABS (A1), the tRefR token is changed to *array class* (tRefA, a *name type formula* sets the *forced array class* state for all parameters of functions returning a value).
- In the cell formula =MDETERM (ABS (A1)), the tRefR token is set to *array class* (tRefA, *forced array class* in MDETERM function parameter).

- Current position is a function parameter expecting an *array class token* (for example the parameter of the MDETERM function):
 - All tokens are changed to *array class*.

Examples for token transformation, when *array class* is expected:

- In the formula =MDETERM (A1), the tRefR token is changed to *array class* (tRefA).
- In the formula =MDETERM (PI ()), the tFuncV(PI) token is changed to *array class* (tFuncA).
- In the formula =MDETERM ({ 1 }), the tArrayA token keeps unchanged.

4.2.5 Binary Reference Operators

The binary reference operators tIsect (→4.6.13), tList (→4.6.14), and tRange (→4.6.15) are always encapsulated into reference subexpression tokens (for instance tMemArea, →4.9.4, or tMemFunc, →4.9.7). These tokens will change their token class, but the operands of the reference operators will not be changed. Because these operators require *reference operands* (constant references or functions returning a reference), the actual operands of these operators always have *reference class*.

Example: The ABS function in the cell formula =ABS (A1 A1) expects *value class* for its parameter. The intersection operator tIsect and its operands are encapsulated into a tMemAreaR token, which is changed to *value class* in cell formulas. The tRefR tokens of the intersection operator are not changed.

The resulting token array of this formula is:

tMemAreaV, tRefR(A1), tRefR(A1), tIsect, tFuncV(ABS).

4.3 Cell Addresses in Tokens

All tokens containing cell addresses (with the two components row and column) store them in the same format. There are differences in storing relative components of an address.

4.3.1 Absolute and Relative Components of a Cell Address

An absolute component of an address (for example the row in B\$6) points always to the same fixed row or column. It is stored with its zero-based index (the value 0 refers to row 1 or column A).

To store a relative component (for example the column in B\$6), there are two different ways used in an Excel file:

- In method [A], the reference component is stored equally to absolute components (with its zero-based index), together with the information, that the component is relative.
- In method [B], a signed offset is stored for the reference component. The resulting address can be calculated later from the position in the current context (for example the cell position in which a defined name is used).

The following table shows which formula type in Excel uses which method to store relative components:

Method	Formula type
[A]	Cell formulas (→5.7), array formulas (→5.8)
[B]	Shared formulas (→5.8), conditional formatting (→5.12), data validity (→5.14), defined names (→5.10)

Example for method [A]: The reference in the cell formula =C4 is stored in a tRef token (→4.9.2). Regardless of the position of the formula, the token always contains the row index 3 (zero-based index for row 4) and the column index 2 (zero-based column index for column C), together with two flags, that row and column indexes are relative. These flags are used for example when the cell is copied to another position (to adjust the relative reference components), but they do not influence the evaluation of the formula.

Example for method [B]: The reference in the condition =C4 in a conditional formatting, entered in cell A7, is stored in a tRefN token (→4.9.10). The token does not contain the address C4 itself, but the difference to the base cell A7, therefore it contains the row offset -3 (three rows up) and the column offset +2 (2 columns to the right). If the same conditional formatting is copied, the token will not be modified, because it only contains offset values, and not an explicit cell address. If it is copied for example to cell B8, the condition will refer to cell D5 (three rows up and 2 columns to the right from cell B8).

4.3.2 Reference Token Replacement

In formulas using method [B], several tokens are replaced by similar tokens that have the letter “N” appended to their name. Other tokens do not have such a counterpart and are used in both types of formulas (for example, tRef always uses method [A], tRefN always uses method [B], and tRef3d uses either method, dependent on the formula type). The following tables show the details.

Reference token replacement, BIFF2-BIFF4:

Cell formulas, array formulas [A]	Defined names [B]
tRef/tArea	tRefN/tAreaN
tRefErr/tAreaErr	tRefErr/tAreaErr
tMemArea	tMemAreaN
tMemNoMem	tMemNoMemN
tMemFunc	tMemFunc
tMemErr	tMemErr

Reference token replacement, BIFF5-BIFF8:

Cell formulas, array formulas [A]	Defined names [B]	Shared formulas [B]	Conditional formatting, data validity [B] (BIFF8)
tRef/tArea	tRef3d/tArea3d	tRefN/tAreaN	tRefN/tAreaN
tRefErr/tAreaErr	tRefErr3d/tAreaErr3d	tRefErr/tAreaErr	tRefErr/tAreaErr
tRef3d/tArea3d	tRef3d/tArea3d	—	—
tRefErr3d/tAreaErr3d	tRefErr3d/tAreaErr3d	—	—
tMemArea	tMemFunc	—	—
tMemNoMem	tMemFunc	—	—
tMemFunc	tMemFunc	—	—
tMemErr	tMemErr	—	—

In BIFF5-BIFF8, the tRefN and tAreaN tokens do *not* occur, if all components of the contained cell address or cell range address are absolute. In this case, the respective tRef or tArea tokens are used.

Example: The shared formulas =A1, =\$A1, and =A\$1 (shared formulas use method [B]) use a tRefNV token. But the shared formula =\$A\$1 uses a tRefV token instead.

4.3.3 Cell Addresses in BIFF2-BIFF7

In the file format versions up to BIFF7, it is possible to use 16384 (2^{14}) rows. A cell address contains the row index as a 14-bit value, the column index as an 8-bit value, and two flags. The flags, encoded into the row index, specify whether the row or column index is absolute or relative. The differences in interpreting relative indexes are described in 4.3.1.

Index to row or row offset, with embedded relative flags (16-bit value), BIFF2-BIFF7:

Bit	Mask	Contents
13-0	3FFF _H	Index to row (0...16383) or row offset (method [B], -8192...8191)
14	4000 _H	0 = Absolute column index 1 = Relative column index, or column offset
15	8000 _H	0 = Absolute row index 1 = Relative row index, or row offset

Cell address, BIFF2-BIFF7:

Offset	Size	Contents
0	2	Index to row or row offset, with relative flags (see table above)
2	1	Index to column (0...255) or column offset (method [B], -128...127)

Cell range address, BIFF2-BIFF7:

Offset	Size	Contents
0	2	Index to first row or offset of first row, with relative flags (see table above)
2	2	Index to last row or relative offset of last row, with relative flags (see table above)
4	1	Index to first column (0...255) or offset of first column (method [B], -128...127)
5	1	Index to last column (0...255) or offset of last column (method [B], -128...127)

Example for a reference in a cell formula (method [A], token tRef): The reference B\$6 consists of the absolute row index 5 and the relative column index 1. The value of the encoded row index is 4005_H (row 6, column is relative). The value of the column index is 01_H (column B). Though the column is relative, the fixed index to column B is stored in cell formulas.

Example for a relative reference in a shared formula (method [B], token tRefN): The reference “decrease column by 1, absolute row 6” (which would evaluate to the address B6, if used in cell C1) consists of the absolute row index 5, and (in shared formulas) the relative column offset -1. The value of the encoded row index is 4005_H (row 6, column is relative). The value of the column index is FF_H (the signed offset -1).

4.3.4 Cell Addresses in BIFF8

From BIFF8 on, 65536 (2^{16}) rows are available. A cell address contains the row index as a 16-bit value, the column index as an 8-bit value, and two flags. The flags are encoded into the column index (which therefore needs 2 bytes), and specify whether the row or column index is absolute or relative. The differences in interpreting relative indexes are described in 4.3.1.

Index to column or column offset, with embedded relative flags (16-bit value), BIFF8:

Bit	Mask	Contents
7-0	00FF _H	Index to column (0...255) or column offset (method [B], -128...127)
14	4000 _H	0 = Absolute column index 1 = Relative column index, or column offset
15	8000 _H	0 = Absolute row index 1 = Relative row index, or row offset

Cell address, BIFF8:

Offset	Size	Contents
0	2	Index to row (0...65535) or row offset (method [B], -32768...32767)
2	2	Index to column or column offset, with relative flags (see table above)

Cell range address, BIFF8:

Offset	Size	Contents
0	2	Index to first row (0...65535) or offset of first row (method [B], -32768...32767)
2	2	Index to last row (0...65535) or offset of last row (method [B], -32768...32767)
4	2	Index to first column or offset of first column, with relative flags (see table above)
6	2	Index to last column or offset of last column, with relative flags (see table above)

Example for a reference in a cell formula (method [A], token tRef): The reference B\$6 consists of the absolute row index 5 and the relative column index 1. The value of the row index is 0005_H (row 6). The value of the encoded column index is 4001_H (column B, column is relative). Though the column is relative, the fixed index to column B is stored in cell formulas.

Example for a relative reference in a shared formula (method [B], token tRefN): The reference “decrease column by 1, absolute row 6” (which would evaluate to the address B6, if used in cell C1) consists of the absolute row index 5, and (in shared formulas) the relative column offset -1. The value of the row index is 0005_H (row 6). The value of the encoded column index is 7FFF_H (the signed 14-bit offset 3FFF_H = -1, and the column relative flag).

4.4 Token Overview

Following a list of all tokens, separated into the several token types and ordered by token identifier.

4.4.1 Unary Operator Tokens

Token ID	Token name	Description
12 _H	tUplus	Unary plus
13 _H	tUminus	Unary minus
14 _H	tPercent	Percent sign

4.4.2 Binary Operator Tokens

Token ID	Token name	Description
03 _H	tAdd	Addition
04 _H	tSub	Subtraction
05 _H	tMul	Multiplication
06 _H	tDiv	Division
07 _H	tPower	Exponentiation
08 _H	tConcat	Concatenation
09 _H	tLT	Less than
0A _H	tLE	Less than or equal
0B _H	tEQ	Equal
0C _H	tGE	Greater than or equal
0D _H	tGT	Greater than
0E _H	tNE	Not equal
0F _H	tIsect	Cell range intersection
10 _H	tList	Cell range list
11 _H	tRange	Cell range

4.4.3 Function Operator Tokens

Token ID	Token name	Description
21 _H 41 _H 61 _H	tFunc	Function with fixed number of arguments
22 _H 42 _H 62 _H	tFuncVar	Function or macro command with variable number of arguments
38 _H 58 _H 78 _H	tFuncCE	Macro command with variable number of arguments (BIFF2-BIFF3)

4.4.4 Constant Operand Tokens

Token ID	Token name	Description
16 _H	tMissArg	Missing argument
17 _H	tStr	String constant
1C _H	tErr	Error constant
1D _H	tBool	Boolean constant
1E _H	tInt	Integer constant
1F _H	tNum	Floating-point constant
20 _H 40 _H 60 _H	tArray	Array constant

4.4.5 Operand Tokens

Token ID	Token name	Description
23 _H 43 _H 63 _H	tName	Internal defined name
24 _H 44 _H 64 _H	tRef	2D cell reference
25 _H 45 _H 65 _H	tArea	2D area reference
26 _H 46 _H 66 _H	tMemArea	Constant reference subexpression
27 _H 47 _H 67 _H	tMemErr	Deleted constant reference subexpression
28 _H 48 _H 68 _H	tMemNoMem	Incomplete constant reference subexpression
29 _H 49 _H 69 _H	tMemFunc	Variable reference subexpression
2A _H 4A _H 6A _H	tRefErr	Deleted 2D cell reference
2B _H 4B _H 6B _H	tAreaErr	Deleted 2D area reference
2C _H 4C _H 6C _H	tRefN	Relative 2D cell reference
2D _H 4D _H 6D _H	tAreaN	Relative 2D area reference
2E _H 4E _H 6E _H	tMemAreaN	Relative constant reference subexpression
2F _H 4F _H 6F _H	tMemNoMemN	Incomplete relative constant reference subexpression
39 _H 59 _H 79 _H	tNameX	External name (BIFF5-BIFF8)
3A _H 5A _H 7A _H	tRef3d	3D cell reference (BIFF5-BIFF8)
3B _H 5B _H 7B _H	tArea3d	3D area reference (BIFF5-BIFF8)
3C _H 5C _H 7C _H	tRefErr3d	Deleted 3D cell reference (BIFF5-BIFF8)
3D _H 5D _H 7D _H	tAreaErr3d	Deleted 3D area reference (BIFF5-BIFF8)

4.4.6 Control Tokens, Special Tokens

Token ID	Token name	Description
01 _H	tExp	Matrix formula or shared formula
02 _H	tTbl	Multiple operation table
15 _H	tParen	Parentheses
18 _H	tExtended	Natural language reference (BIFF8)
19 _H	tAttr	Special attribute
1A _H	tSheet	Start of external sheet reference (BIFF2-BIFF4)
1B _H	tEndSheet	End of external sheet reference (BIFF2-BIFF4)

4.4.7 Overview, Ordered by Token Identifiers

The following table shows all tokens, ordered by their identifiers, together with their sizes in bytes. The sizes include the token identifier. Note that the tStr token, the tExtended token, and the tAttr token do not have a fixed size.

• Base Tokens (00_H–1F_H)

Token ID	Token name	BIFF2	BIFF3	BIFF4	BIFF5/7	BIFF8
00 _H	<i>Not used</i>	—	—	—	—	—
01 _H	tExp	4	5	5	5	5
02 _H	tTbl	4	5	5	5	5
03 _H	tAdd	1	1	1	1	1
04 _H	tSub	1	1	1	1	1
05 _H	tMul	1	1	1	1	1
06 _H	tDiv	1	1	1	1	1
07 _H	tPower	1	1	1	1	1
08 _H	tConcat	1	1	1	1	1
09 _H	tLT	1	1	1	1	1
0A _H	tLE	1	1	1	1	1
0B _H	tEQ	1	1	1	1	1
0C _H	tGE	1	1	1	1	1
0D _H	tGT	1	1	1	1	1
0E _H	tNE	1	1	1	1	1
0F _H	tIsect	1	1	1	1	1
10 _H	tList	1	1	1	1	1
11 _H	tRange	1	1	1	1	1
12 _H	tUplus	1	1	1	1	1
13 _H	tUminus	1	1	1	1	1
14 _H	tPercent	1	1	1	1	1
15 _H	tParen	1	1	1	1	1
16 _H	tMissArg	1	1	1	1	1
17 _H	tStr	var.	var.	var.	var.	var.
18 _H	tExtended	—	—	—	—	var.
19 _H	tAttr	var.	var.	var.	var.	var.
1A _H	tSheet	8	11	11	—	—
1B _H	tEndSheet	4	5	5	—	—
1C _H	tErr	2	2	2	2	2
1D _H	tBool	2	2	2	2	2
1E _H	tInt	3	3	3	3	3
1F _H	tNum	9	9	9	9	9

• **Classified Tokens (20_H–7F_H)**

Token ID	Token name	BIFF2	BIFF3	BIFF4	BIFF5/7	BIFF8
20 _H 40 _H 60 _H	tArray	8	9	9	9	9
21 _H 41 _H 61 _H	tFunc	3	3	4	4	4
22 _H 42 _H 62 _H	tFuncVar	4	4	5	5	5
23 _H 43 _H 63 _H	tName	8	11	11	15	5
24 _H 44 _H 64 _H	tRef	4	4	4	4	5
25 _H 45 _H 65 _H	tArea	7	7	7	7	9
26 _H 46 _H 66 _H	tMemArea	5	7	7	7	7
27 _H 47 _H 67 _H	tMemErr	5	7	7	7	7
28 _H 48 _H 68 _H	tMemNoMem	5	7	7	7	7
29 _H 49 _H 69 _H	tMemFunc	2	3	3	3	3
2A _H 4A _H 6A _H	tRefErr	4	4	4	4	5
2B _H 4B _H 6B _H	tAreaErr	7	7	7	7	9
2C _H 4C _H 6C _H	tRefN	4	4	4	4	5
2D _H 4D _H 6D _H	tAreaN	7	7	7	7	9
2E _H 4E _H 6E _H	tMemAreaN	2	3	3	3	3
2F _H 4F _H 6F _H	tMemNoMemN	2	3	3	3	3
30 _H 50 _H 70 _H	<i>Not used</i>	—	—	—	—	—
31 _H 51 _H 71 _H	<i>Not used</i>	—	—	—	—	—
32 _H 52 _H 72 _H	<i>Not used</i>	—	—	—	—	—
33 _H 53 _H 73 _H	<i>Not used</i>	—	—	—	—	—
34 _H 54 _H 74 _H	<i>Not used</i>	—	—	—	—	—
35 _H 55 _H 75 _H	<i>Not used</i>	—	—	—	—	—
36 _H 56 _H 76 _H	<i>Not used</i>	—	—	—	—	—
37 _H 57 _H 77 _H	<i>Not used</i>	—	—	—	—	—
38 _H 58 _H 78 _H	tFuncCE	3	3	—	—	—
39 _H 59 _H 79 _H	tNameX	—	—	—	25	7
3A _H 5A _H 7A _H	tRef3d	—	—	—	18	7
3B _H 5B _H 7B _H	tArea3d	—	—	—	21	11
3C _H 5C _H 7C _H	tRefErr3d	—	—	—	18	7
3D _H 5D _H 7D _H	tAreaErr3d	—	—	—	21	11
3E _H 5E _H 7E _H	<i>Not used</i>	—	—	—	—	—
3F _H 5F _H 7F _H	<i>Not used</i>	—	—	—	—	—

4.5 Unary Operator Tokens

Unary operators perform an operation with the topmost operand from stack. The tokens do not contain any additional data.

4.5.1 tUplus (12_H)

Unary plus operator. This operator has no effect on the operand. Operand class: *value*.

Token tUplus, BIFF2-BIFF8:

Offset	Size	Contents
0	1	12 _H

Example: +A1 returns the value of cell A1.

4.5.2 tUminus (13_H)

Unary minus operator. Negates the operand. Operand class: *value*.

Token tUminus, BIFF2-BIFF8:

Offset	Size	Contents
0	1	13 _H

Example: -A1 returns the negated value of cell A1.

4.5.3 tPercent (14_H)

Percent sign. Divides the operand by 100. Operand class: *value*.

Token tPercent, BIFF2-BIFF8:

Offset	Size	Contents
0	1	14 _H

Example: 1% returns 0.01.

4.6 Binary Operator Tokens

Binary operators perform an operation with the two topmost operands from stack. The tokens do not contain any additional data.

4.6.1 tAdd (03_H)

Addition operator. Adds the operands. Operand class: *value*.

Token tAdd, BIFF2-BIFF8:

Offset	Size	Contents
0	1	03 _H

Example: 3+2 returns 5.

4.6.2 tSub (04_H)

Subtraction operator. Subtracts the top operand from the second-to-top operand. Operand class: *value*.

Token tSub, BIFF2-BIFF8:

Offset	Size	Contents
0	1	04 _H

Example: 3-2 returns 1.

4.6.3 tMul (05_H)

Multiplication operator. Multiplies the operands. Operand class: *value*.

Token tMul, BIFF2-BIFF8:

Offset	Size	Contents
0	1	05 _H

Example: 3*2 returns 6.

4.6.4 tDiv (06_H)

Division operator. Divides the second-to-top operand by the top operand. Operand class: *value*.

Token tDiv, BIFF2-BIFF8:

Offset	Size	Contents
0	1	06 _H

Example: 3/2 returns 1.5.

4.6.5 tPower (07_H)

Exponentiation operator. Raises the second-to-top operand to the power of the top operand. Operand class: *value*.

Token tPower, BIFF2-BIFF8:

Offset	Size	Contents
0	1	07 _H

Example: 3^2 returns 9.

4.6.6 tConcat (08_H)

Concatenation operator. Appends the top operand to the second-to-top operand. Operand class: *value*.

Token tConcat, BIFF2-BIFF8:

Offset	Size	Contents
0	1	08 _H

Example: "ABC"&"DEF" returns "ABCDEF".

4.6.7 tLT (09_H)

Less than operator. Returns TRUE if the second-to-top operand is less than the top operand. Operand class: *value*.

Token tLT, BIFF2-BIFF8:

Offset	Size	Contents
0	1	09 _H

Example: $3 < 2$ returns FALSE.

4.6.8 tLE (0A_H)

Less than or equal operator. Returns TRUE if the second-to-top operand is less than or equal to the top operand. Operand class: *value*.

Token tLE, BIFF2-BIFF8:

Offset	Size	Contents
0	1	0A _H

Example: $3 \leq 2$ returns FALSE.

4.6.9 tEQ (0B_H)

Equality operator. Returns TRUE if the operands are equal. Operand class: *value*.

Token tEQ, BIFF2-BIFF8:

Offset	Size	Contents
0	1	0B _H

Example: 3=2 returns FALSE.

4.6.10 tGE (0C_H)

Greater than or equal operator. Returns TRUE if the second-to-top operand is greater than or equal to the top operand. Operand class: *value*.

Token tGE, BIFF2-BIFF8:

Offset	Size	Contents
0	1	0C _H

Example: 3>=2 returns TRUE.

4.6.11 tGT (0D_H)

Greater than operator. Returns TRUE if the second-to-top operand is greater than the top operand. Operand class: *value*.

Token tGT, BIFF2-BIFF8:

Offset	Size	Contents
0	1	0D _H

Example: 3>2 returns TRUE.

4.6.12 tNE (0E_H)

Inequality operator. Returns TRUE if the operands are not equal. Operand class: *value*.

Token tNE, BIFF2-BIFF8:

Offset	Size	Contents
0	1	0E _H

Example: 3<>2 returns TRUE.

4.6.13 tIsect (0F_H)

Intersection operator, represented by the space sign. Returns the intersected range of two ranges. If the resulting cell range is empty, the formula will return the error code “#NULL!” (for instance A1 : A2 B3). This token is not allowed in conditions of conditional formattings and in data validity. If this token is used in cell formulas, it prevents the creation of shared formulas. Operand class: *reference*.

Token tIsect, BIFF2-BIFF8:

Offset	Size	Contents
0	1	0F _H

Example: A1 : B3 B2 : C3 returns B2 : B3.

4.6.14 tList (10_H)

Range list operator, represented by the system's list separator sign (for example comma sign). Treats two ranges as one operator. This might be useful for function parameters. Note: This is *not* a union operator. This token is not allowed in conditions of conditional formattings and in data validity. If this token is used in cell formulas, it prevents the creation of shared formulas. Operand class: *reference*.

Token tList, BIFF2-BIFF8:

Offset	Size	Contents
0	1	10 _H

Example: (A1 : A2 , A2 : A3) will be handled as one operand. Cell A2 will be handled twice, therefore this is a range list operator, *no* union operator.

4.6.15 tRange (11_H)

Range operator, represented by the colon sign. Returns the minimal rectangular range that contains both parameters. This token occurs for instance by using defined names. This token is not allowed in conditions of conditional formattings and in data validity. If this token is used in cell formulas, it prevents the creation of shared formulas. Operand class: *reference*.

Token tRange, BIFF2-BIFF8:

Offset	Size	Contents
0	1	11 _H

Example: namedcell : D5 returns A1 : D5, if the defined name namedcell contains A1 : B2.

4.7 Function Operator Tokens

The reference class of function operator tokens depend on the function itself and on the position of the function inside of the formula (for instance the expected class for the current parameter of an enclosing function).

4.7.1 tFunc (21_H, 41_H, 61_H)

This token contains the index to a built-in function with fixed number of arguments. The operand class is dependent on the return type of the function.

Token tFunc, BIFF2-BIFF3:

Offset	Size	Contents
0	1	21 _H (tFuncR), 41 _H (tFuncV), 61 _H (tFuncA)
1	1	Index to a built-in sheet function (→4.11)

Token tFunc, BIFF4-BIFF8:

Offset	Size	Contents
0	1	21 _H (tFuncR), 41 _H (tFuncV), 61 _H (tFuncA)
1	2	Index to a built-in sheet function (→4.11)

4.7.2 tFuncVar (22_H, 42_H, 62_H)

This token contains the index to a built-in function or a macro command with variable number of arguments. The operand class is dependent on the return type of the function.

Token tFuncVar, BIFF2-BIFF3:

Offset	Size	Contents
0	1	22 _H (tFuncVarR), 42 _H (tFuncVarV), 62 _H (tFuncVarA)
1	1	Number of arguments
2	1	Index to a built-in sheet function (→4.11)

Token tFuncVar, BIFF4-BIFF8:

Offset	Size	Contents									
0	1	22 _H (tFuncVarR), 42 _H (tFuncVarV), 62 _H (tFuncVarA)									
1	1	Number of arguments									
		<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>6-0</td> <td>7F_H</td> <td>Number of arguments</td> </tr> <tr> <td>7</td> <td>80_H</td> <td>1 = User prompt for macro commands (shown by a question mark following the command name)</td> </tr> </tbody> </table>	Bit	Mask	Contents	6-0	7F _H	Number of arguments	7	80 _H	1 = User prompt for macro commands (shown by a question mark following the command name)
Bit	Mask	Contents									
6-0	7F _H	Number of arguments									
7	80 _H	1 = User prompt for macro commands (shown by a question mark following the command name)									
2	2	Index to a sheet function									
		<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>14-0</td> <td>7FFF_H</td> <td>Index to a built-in sheet function (→4.11) or a macro command</td> </tr> <tr> <td>15</td> <td>8000_H</td> <td>0 = Built-in function; 1 = Macro command</td> </tr> </tbody> </table>	Bit	Mask	Contents	14-0	7FFF _H	Index to a built-in sheet function (→4.11) or a macro command	15	8000 _H	0 = Built-in function; 1 = Macro command
Bit	Mask	Contents									
14-0	7FFF _H	Index to a built-in sheet function (→4.11) or a macro command									
15	8000 _H	0 = Built-in function; 1 = Macro command									

4.7.3 tFuncCE (38_H, 58_H, 78_H)

This token contains the index to a macro sheet command. From BIFF4 on, macro commands are represented by tFuncVar tokens (→4.7.2). Operand class: *reference*.

Token tFuncCE, BIFF2-BIFF3:

Offset	Size	Contents
0	1	38 _H (tFuncCER), 58 _H (tFuncCEV), 78 _H (tFuncCEA)
1	1	Number of arguments
2	1	Index to a macro sheet command

4.8 Constant Operand Tokens

4.8.1 tMissArg (16_H)

A missing argument in a function argument list is stored as a tMissArg token. Operand class: *value*.

Token tMissArg, BIFF2-BIFF8:

Offset	Size	Contents
0	1	16 _H

Example: SUM (1, , 3) – Second argument is missing and represented by a tMissArg token.

4.8.2 tStr (17_H)

This token contains a string constant. The maximum length of the string is 253 characters in BIFF2 (due to the limitation of 255 bytes per formula) and 255 characters in BIFF3-BIFF8. Operand class: *value*.

Token tStr, BIFF2-BIFF7:

Offset	Size	Contents
0	1	17 _H
1	var.	Byte string, 8-bit string length (→3.3)

Token tStr, BIFF8:

Offset	Size	Contents
0	1	17 _H
1	var.	Unicode string, 8-bit string length, option flags occur always (→3.4)

Example: LEN ("ABC") – The string constant "ABC" is represented by a tStr token.

4.8.3 tErr (1C_H)

This token contains an error code. Operand class: *value*.

Token tErr, BIFF2-BIFF8:

Offset	Size	Contents
0	1	1C _H
1	1	Error code (→3.7)

Example: ERROR . TYPE (#NV) – The constant error code #NV is represented by a tErr token.

4.8.4 tBool (1D_H)

This token contains a Boolean value (TRUE or FALSE). Operand class: *value*.

Token tBool, BIFF2-BIFF8:

Offset	Size	Contents
0	1	1D _H
1	1	0 = FALSE, 1 = TRUE

Example: IF (A1<0, FALSE, TRUE) – The constants FALSE and TRUE are represented by tBool tokens.

4.8.5 tInt (1E_H)

This token contains an unsigned 16-bit integer value in the range from 0 to 65535. Operand class: *value*.

Token tInt, BIFF2-BIFF8:

Offset	Size	Contents
0	1	1E _H
1	2	Unsigned integer value

Example: 3+2.2222 – The constant 3 is represented by a tInt token.

4.8.6 tNumber (1F_H)

This token contains a floating-point number. Operand class: *value*.

Token tNumber, BIFF2-BIFF8:

Offset	Size	Contents
0	1	1F _H
1	8	IEEE 754 floating-point value (64-bit double precision) ⁷

Example: 3+2.2222 – The constant 2.2222 is represented by a tNumber token.

⁷ For details about the internal structure of floating-point values see http://en.wikipedia.org/wiki/IEEE_floating-point_standard.

4.8.7 tArray (20_H, 40_H, 60_H)

This token contains an array constant. For instance the 2x1 matrix { 1 ; 2 } is an array constant. The values of the array constant do not follow the token identifier but are stored behind the complete token array (→4.1.6) in a constant value array(→3.8.2). This token is not allowed in conditions of conditional formattings and in data validity. If this token is used in cell formulas, it prevents the creation of shared formulas. Operand class: *array*.

Token tArray, BIFF2:

Offset	Size	Contents
0	1	20 _H (tArrayR), 40 _H (tArrayV), 60 _H (tArrayA)
1	6	Not used

Token tArray, BIFF3-BIFF8:

Offset	Size	Contents
0	1	20 _H (tArrayR), 40 _H (tArrayV), 60 _H (tArrayA)
1	7	Not used

4.9 Operand Tokens

4.9.1 tName (23_H, 43_H, 63_H)

This token contains the *one-based* index to a NAME record (→6.66). In BIFF2-BIFF4 this could be the index to an EXTERNNAME record (→6.38) too. From BIFF5 on an external name or an internal name with explicit sheet name (for example “Sheet1!localname”) is represented by the token tNameX (→4.9.14). If this token is used in cell formulas, it prevents the creation of shared formulas. Operand class: *reference*.

Token tName, BIFF2:

Offset	Size	Contents
0	1	23 _H (tNameR), 43 _H (tNameV), 63 _H (tNameA)
1	2	<i>One-based</i> index to NAME record (→6.66) or EXTERNNAME record (→6.38)
3	5	Not used

Token tName, BIFF3-BIFF4:

Offset	Size	Contents
0	1	23 _H (tNameR), 43 _H (tNameV), 63 _H (tNameA)
1	2	<i>One-based</i> index to NAME record (→6.66) or EXTERNNAME record (→6.38)
3	8	Not used

Token tName, BIFF5/BIFF7:

Offset	Size	Contents
0	1	23 _H (tNameR), 43 _H (tNameV), 63 _H (tNameA)
1	2	<i>One-based</i> index to NAME record (→6.66) in the <i>Global Link Table</i> (→5.10.2)
3	12	Not used

Token tName, BIFF8:

Offset	Size	Contents
0	1	23 _H (tNameR), 43 _H (tNameV), 63 _H (tNameA)
1	2	<i>One-based</i> index to NAME record (→6.66) in the <i>Link Table</i> (→5.10.3)
3	2	Not used

4.9.2 tRef (24_H, 44_H, 64_H)

This token contains the reference to a cell in the same sheet. Operand class: *reference*.

Token tRef, BIFF2-BIFF7:

Offset	Size	Contents
0	1	24 _H (tRefR), 44 _H (tRefV), 64 _H (tRefA)
1	3	Encoded cell address (→4.3.3)

Token tRef, BIFF8:

Offset	Size	Contents
0	1	24 _H (tRefR), 44 _H (tRefV), 64 _H (tRefA)
1	4	Encoded cell address (→4.3.4)

4.9.3 tArea (25_H, 45_H, 65_H)

This token contains the reference to a cell range in the same sheet. Operand class: *reference*.

Token tArea, BIFF2-BIFF7:

Offset	Size	Contents
0	1	25 _H (tAreaR), 45 _H (tAreaV), 65 _H (tAreaA)
1	6	Encoded cell range address (→4.3.3)

Token tArea, BIFF8:

Offset	Size	Contents
0	1	25 _H (tAreaR), 45 _H (tAreaV), 65 _H (tAreaA)
1	8	Encoded cell range address (→4.3.4)

4.9.4 tMemArea (26_H, 46_H, 66_H)

This token encapsulates a reference subexpression (→4.2.5) that results in a constant cell address, cell range address, or cell range list on the same sheet. The token provides the result of the reference subexpression in a cell range address list (→3.13.2). This list does not follow the token identifier, but is stored behind the complete token array (→4.1.6).

The tMemArea token is not allowed in conditions of conditional formattings and in data validity. If this token is used in cell formulas, it prevents the creation of shared formulas. Operand class: *reference*.

Token tMemArea, BIFF2:

Offset	Size	Contents
0	1	26 _H (tMemAreaR), 46 _H (tMemAreaV), 66 _H (tMemAreaA)
1	3	Not used
4	1	Size of the following reference subexpression (all reference operators with their operands)

Token tMemArea, BIFF3-BIFF8:

Offset	Size	Contents
0	1	26 _H (tMemAreaR), 46 _H (tMemAreaV), 66 _H (tMemAreaA)
1	4	Not used
5	2	Size of the following reference subexpression (all reference operators with their operands)

Examples for constant reference subexpressions represented by a tMemArea token:

Reference subexpression	Resulting cell range address list
A1:B2 B2:C3	B2
A1:B2:B2:C3	A1:C3
(A1:B2, B2:C3)	A1:B2, B2:C3
(A1:B2, B2:C3) B1:B3	B1:B2, B2:B3

4.9.5 tMemErr (27_H, 47_H, 67_H)

This token encapsulates a reference subexpression (→4.2.5) that results in an erroneous cell range list on the same sheet. This token is not allowed in conditions of conditional formattings and in data validity. If this token is used in cell formulas, it prevents the creation of shared formulas. Operand class: *reference*.

Token tMemErr, BIFF2:

Offset	Size	Contents
0	1	27 _H (tMemErrR), 47 _H (tMemErrV), 67 _H (tMemErrA)
1	3	Not used
4	1	Size of the following reference subexpression (all reference operators with their operands)

Token tMemArea, BIFF3-BIFF8:

Offset	Size	Contents
0	1	27 _H (tMemErrR), 47 _H (tMemErrV), 67 _H (tMemErrA)
1	4	Not used
5	2	Size of the following reference subexpression (all reference operators with their operands)

Examples for erroneous reference subexpressions represented by a tMemErr token:

Reference subexpression	Reason for the error
A1:B2 C3:D4	Intersection results in an empty cell range list (#NULL! error)
(A1:B2, #REF!)	One of the references is invalid (for example after deleting a row)

4.9.6 tMemNoMem (28_H, 48_H, 68_H)

Whenever a tMemArea token (→4.9.4) should be created (containing the result cell range list of a reference subexpression, →4.2.5), but there was not enough memory to calculate that resulting cell range list, a tMemNoMem token is used instead. Its contents are equal to the tMemArea token but it does not append a cell range list to the token array.

This token is not allowed in conditions of conditional formattings and in data validity. If this token is used in cell formulas, it prevents the creation of shared formulas. Operand class: *reference*.

Token tMemNoMem, BIFF2:

Offset	Size	Contents
0	1	28 _H (tMemNoMemR), 48 _H (tMemNoMemV), 68 _H (tMemNoMemA)
1	3	Not used
4	1	Size of the following reference subexpression (all reference operators with their operands)

Token tMemNoMem, BIFF3-BIFF8:

Offset	Size	Contents
0	1	28 _H (tMemNoMemR), 48 _H (tMemNoMemV), 68 _H (tMemNoMemA)
1	4	Not used
5	2	Size of the following reference subexpression (all reference operators with their operands)

4.9.7 tMemFunc (29_H, 49_H, 69_H)

This token encapsulates a reference subexpression (→4.2.5) that results in a non-constant cell address, cell range address, or cell range list. Whenever one operand of the reference subexpression is a function, a defined name, a 3D reference, or an external reference (and no error occurs), a tMemFunc token is used.

This token is not allowed in conditions of conditional formattings and in data validity. If this token is used in cell formulas, it prevents the creation of shared formulas. Operand class: *reference*.

Token tMemFunc, BIFF2:

Offset	Size	Contents
0	1	29 _H (tMemFuncR), 49 _H (tMemFuncV), 69 _H (tMemFuncA)
1	1	Size of the following reference subexpression (all reference operators with their operands)

Token tMemFunc, BIFF3-BIFF8:

Offset	Size	Contents
0	1	29 _H (tMemFuncR), 49 _H (tMemFuncV), 69 _H (tMemFuncA)
1	2	Size of the following reference subexpression (all reference operators with their operands)

Examples for non-constant reference subexpressions represented by a tMemFunc token:

Reference subexpression	Description
A1:B2 myname	Defined name used
A1:INDEX(A2;1;1)	Function used
(A1:B2,Sheet2!C3)	3D reference used
(A1:B2,extfile.xls!C3)	External reference used

4.9.8 tRefErr (2A_H, 4A_H, 6A_H)

This token contains the last reference to a deleted cell in the same sheet. Operand class: *reference*.

Token tRefErr, BIFF2-BIFF7:

Offset	Size	Contents
0	1	2A _H (tRefErrR), 4A _H (tRefErrV), 6A _H (tRefErrA)
1	3	Not used

Token tRefErr, BIFF8:

Offset	Size	Contents
0	1	2A _H (tRefErrR), 4A _H (tRefErrV), 6A _H (tRefErrA)
1	4	Not used

4.9.9 tAreaErr (2B_H, 4B_H, 6B_H)

This token contains the last reference to a deleted cell range in the same sheet. Operand class: *reference*.

Token tAreaErr, BIFF2-BIFF7:

Offset	Size	Contents
0	1	2B _H (tAreaErrR), 4B _H (tAreaErrV), 6B _H (tAreaErrA)
1	6	Not used

Token tAreaErr, BIFF8:

Offset	Size	Contents
0	1	2B _H (tAreaErrR), 4B _H (tAreaErrV), 6B _H (tAreaErrA)
1	8	Not used

4.9.10 tRefN (2C_H, 4C_H, 6C_H)

This token contains the relative reference to a cell in the same sheet. It stores relative components as signed offsets and is used in defined names (BIFF2-BIFF4), shared formulas, conditional formatting, and data validity. If all components of the cell address are absolute, a tRef token (→4.9.2) is used instead. For more details see →4.3.1 and →4.3.2. Operand class: *reference*.

Token tRefN, BIFF2-BIFF7:

Offset	Size	Contents
0	1	2C _H (tRefNR), 4C _H (tRefNV), 6C _H (tRefNA)
1	3	Encoded cell address (→4.3.3)

Token tRefN, BIFF8:

Offset	Size	Contents
0	1	2C _H (tRefNR), 4C _H (tRefNV), 6C _H (tRefNA)
1	4	Encoded cell address (→4.3.4)

4.9.11 tAreaN (2D_H, 4D_H, 6D_H)

This token contains the relative reference to a cell range in the same sheet. It stores relative components as signed offsets and is used in defined names (BIFF2-BIFF4), shared formulas, conditional formatting, and data validity. If all components of the cell range address are absolute, a tArea token (→4.9.3) is used instead. For more details see →4.3.1 and →4.3.2. Operand class: *reference*.

Token tAreaN, BIFF2-BIFF7:

Offset	Size	Contents
0	1	2D _H (tAreaNR), 4D _H (tAreaNV), 6D _H (tAreaNA)
1	6	Encoded cell range address (→4.3.3)

Token tAreaN, BIFF8:

Offset	Size	Contents
0	1	2D _H (tAreaNR), 4D _H (tAreaNV), 6D _H (tAreaNA)
1	8	Encoded cell range address (→4.3.4)

4.9.12 tMemAreaN (2E_H, 4E_H, 6E_H)

This token is closely related to the tMemArea token (→4.9.4). It is used in defined names in BIFF2-BIFF4, and encapsulates a reference subexpression (→4.2.5) that results in a constant cell address, cell range address, or cell range list on the same sheet. It does *not* append a cell range address list to the token array. The token is not used anymore in BIFF5-BIFF8, but import filters should be prepared for its existence. Operand class: *reference*.

Token tMemAreaN, BIFF2:

Offset	Size	Contents
0	1	2E _H (tMemAreaNR), 4E _H (tMemAreaNV), 6E _H (tMemAreaNA)
1	1	Size of the following reference subexpression (all reference operators with their operands)

Token tMemAreaN, BIFF3-BIFF8:

Offset	Size	Contents
0	1	2E _H (tMemAreaNR), 4E _H (tMemAreaNV), 6E _H (tMemAreaNA)
1	2	Size of the following reference subexpression (all reference operators with their operands)

4.9.13 tMemNoMemN (2F_H, 4F_H, 6F_H)

This token is closely related to the tMemAreaN token (→4.9.12). It is used in reference subexpressions (→4.2.5) in defined names in BIFF2-BIFF4. Whenever a tMemAreaN token should be created, but there was not enough memory for any reason, a tMemNoMemN token is used instead. Its contents are equal to the tMemAreaN token. The token is not used anymore in BIFF5-BIFF8, but import filters should be prepared for its existence. Operand class: *reference*.

Token tMemNoMemN, BIFF2:

Offset	Size	Contents
0	1	2F _H (tMemNoMemNR), 4F _H (tMemNoMemNV), 6F _H (tMemNoMemNA)
1	1	Size of the following reference subexpression (all reference operators with their operands)

Token tMemNoMemN, BIFF3-BIFF8:

Offset	Size	Contents
0	1	2F _H (tMemNoMemNR), 4F _H (tMemNoMemNV), 6F _H (tMemNoMemNA)
1	2	Size of the following reference subexpression (all reference operators with their operands)

4.9.14 tNameX (39_H, 59_H, 79_H) (BIFF5-BIFF8)

This token contains the index to a NAME or EXTERNNAME record. It occurs by using external names or internal names with explicit sheet name (for example “Sheet1!localname”), add-in functions, DDE links, or linked OLE objects. In BIFF5/BIFF7 the contents of this token differs for internal respectively external names. This token is not allowed in conditions of conditional formattings and in data validity. If this token is used in cell formulas, it prevents the creation of shared formulas. Operand class: *reference*.

Token tNameX for *internal names*, BIFF5/BIFF7:

Offset	Size	Contents
0	1	39 _H (tNameXR), 59 _H (tNameXV), 79 _H (tNameXA)
1	2	This is always a negative value to indicate an internal name. The absolute value is the <i>one-based</i> index to EXTERNSHEET record (→6.39) in the <i>Local Link Table</i> (→5.10.2).
3	8	Not used
11	2	<i>One-based</i> index to NAME record (→6.66) in the <i>Global Link Table</i> (→5.10.2)
13	12	Not used

Token tNameX for *external names*, BIFF5/BIFF7:

Offset	Size	Contents
0	1	39 _H (tNameXR), 59 _H (tNameXV), 79 _H (tNameXA)
1	2	This is always a positive value to indicate an external name. <i>One-based</i> index to EXTERNSHEET record (→6.39) in the <i>Local Link Table</i> (→5.10.2).
3	8	Not used
11	2	<i>One-based</i> index to EXTERNNAME record (→6.38)
13	12	Not used

Token tNameX, BIFF8:

Offset	Size	Contents
0	1	39 _H (tNameXR), 59 _H (tNameXV), 79 _H (tNameXA)
1	2	Index to REF entry in EXTERNSHEET record (→6.39) in the <i>Link Table</i> (→5.10.3)
3	2	<i>One-based</i> index to NAME record (→6.66) or EXTERNNAME record (→6.38)
5	2	Not used

4.9.15 tRef3d (3A_H, 5A_H, 7A_H) (BIFF5-BIFF8)

This token contains a 3D reference or an external reference to a cell. In BIFF5/BIFF7 the contents of this token differs for 3D respectively external references. This token is not allowed in conditions of conditional formattings and in data validity. If this token is used in cell formulas, it prevents the creation of shared formulas. Operand class: *reference*.

Token tRef3d for *3D references*, BIFF5/BIFF7:

Offset	Size	Contents
0	1	3A _H (tRef3dR), 5A _H (tRef3dV), 7A _H (tRef3dA)
1	2	This is always a negative value to indicate a 3D reference. The absolute value is the <i>one-based</i> index to EXTERNSHEET record (→6.39) in the <i>Local Link Table</i> (→5.10.2) containing the name of the <i>first</i> referenced sheet.
3	8	Not used
11	2	Zero-based index to first referenced sheet (FFFF _H = deleted sheet)
13	2	Zero-based index to last referenced sheet (FFFF _H = deleted sheet)
15	3	Encoded cell address (→4.3.3)

Token tRef3d for *external references*, BIFF5/BIFF7:

Offset	Size	Contents
0	1	3A _H (tRef3dR), 5A _H (tRef3dV), 7A _H (tRef3dA)
1	2	This is always a positive value to indicate an external reference. <i>One-based</i> index to EXTERNSHEET record (→6.39) in the <i>Local Link Table</i> (→5.10.2).
3	12	Not used
15	3	Encoded cell address (→4.3.3)

Token tRef3d, BIFF8:

Offset	Size	Contents
0	1	3A _H (tRef3dR), 5A _H (tRef3dV), 7A _H (tRef3dA)
1	2	Index to REF entry in EXTERNSHEET record (→6.39) in the <i>Link Table</i> (→5.10.3)
3	4	Encoded cell address (→4.3.4)

4.9.16 tArea3d (3B_H, 5B_H, 7B_H) (BIFF5-BIFF8)

This token contains a 3D reference or an external reference to a cell range. In BIFF5/BIFF7 the contents of this token differs for 3D respectively external references. This token is not allowed in conditions of conditional formattings and in data validity. If this token is used in cell formulas, it prevents the creation of shared formulas. Operand class: *reference*.

Token tArea3d for *3D references*, BIFF5/BIFF7:

Offset	Size	Contents
0	1	3B _H (tArea3dR), 5B _H (tArea3dV), 7B _H (tArea3dA)
1	2	This is always a negative value to indicate a 3D reference. The absolute value is the <i>one-based</i> index to EXTERNSHEET record (→6.39) in the <i>Local Link Table</i> (→5.10.2) containing the name of the <i>first</i> referenced sheet.
3	8	Not used
11	2	Zero-based index to first referenced sheet (FFFF _H = deleted sheet)
13	2	Zero-based index to last referenced sheet (FFFF _H = deleted sheet)
15	6	Encoded cell range address (→4.3.3)

Token tArea3d for *external references*, BIFF5/BIFF7:

Offset	Size	Contents
0	1	3B _H (tArea3dR), 5B _H (tArea3dV), 7B _H (tArea3dA)
1	2	This is always a positive value to indicate an external reference. <i>One-based</i> index to EXTERNSHEET record (→6.39) in the <i>Local Link Table</i> (→5.10.2).
3	12	Not used
15	6	Encoded cell range address (→4.3.3)

Token tArea3d, BIFF8:

Offset	Size	Contents
0	1	3B _H (tArea3dR), 5B _H (tArea3dV), 7B _H (tArea3dA)
1	2	Index to REF entry in EXTERNSHEET record (→6.39) in the <i>Link Table</i> (→5.10.3)
3	8	Encoded cell range address (→4.3.4)

4.9.17 tRefErr3d (3C_H, 5C_H, 7C_H) (BIFF5-BIFF8)

This token contains the last 3D reference or external reference to a cell in a deleted row or column. In BIFF5/BIFF7 the contents of this token differs for 3D respectively external references. This token is not allowed in conditions of conditional formattings and in data validity. If this token is used in cell formulas, it prevents the creation of shared formulas. Operand class: *reference*.

Token tRefErr3d for *3D references*, BIFF5/BIFF7:

Offset	Size	Contents
0	1	3C _H (tRefErr3dR), 5C _H (tRefErr3dV), 7C _H (tRefErr3dA)
1	2	This is always a negative value to indicate a 3D reference. The absolute value is the <i>one-based</i> index to EXTERNSHEET record (→6.39) in the <i>Local Link Table</i> (→5.10.2).
3	8	Not used
11	2	Index to first referenced sheet (FFFF _H = deleted sheet)
13	2	Index to last referenced sheet (FFFF _H = deleted sheet)
15	3	Not used

Token tRefErr3d for *external references*, BIFF5/BIFF7:

Offset	Size	Contents
0	1	3C _H (tRefErr3dR), 5C _H (tRefErr3dV), 7C _H (tRefErr3dA)
1	2	This is always a positive value to indicate an external reference. <i>One-based</i> index to EXTERNSHEET record (→6.39) in the <i>Local Link Table</i> (→5.10.2).
3	15	Not used

Token tRefErr3d, BIFF8:

Offset	Size	Contents
0	1	3C _H (tRefErr3dR), 5C _H (tRefErr3dV), 7C _H (tRefErr3dA)
1	2	Index to REF entry in EXTERNSHEET record (→6.39) in the <i>Link Table</i> (→5.10.3)
3	4	Not used

4.9.18 tAreaErr3d (3D_H, 5D_H, 7D_H) (BIFF5-BIFF8)

This token contains the last 3D reference or external reference to a cell range in deleted columns or rows. In BIFF5/BIFF7 the contents of this token differs for 3D respectively external references. This token is not allowed in conditions of conditional formattings and in data validity. If this token is used in cell formulas, it prevents the creation of shared formulas. Operand class: *reference*.

Token tAreaErr3d for *3D references*, BIFF5/BIFF7:

Offset	Size	Contents
0	1	3D _H (tAreaErr3dR), 5D _H (tAreaErr3dV), 7D _H (tAreaErr3dA)
1	2	This is always a negative value to indicate a 3D reference. The absolute value is the <i>one-based</i> index to EXTERNSHEET record (→6.39) in the <i>Local Link Table</i> (→5.10.2).
3	8	Not used
11	2	Index to first referenced sheet (FFFF _H = deleted sheet)
13	2	Index to last referenced sheet (FFFF _H = deleted sheet)
15	6	Not used

Token tAreaErr3d for *external references*, BIFF5/BIFF7:

Offset	Size	Contents
0	1	3D _H (tAreaErr3dR), 5D _H (tAreaErr3dV), 7D _H (tAreaErr3dA)
1	2	This is always a positive value to indicate an external reference. <i>One-based</i> index to EXTERNSHEET record (→6.39) in the <i>Local Link Table</i> (→5.10.2).
3	18	Not used

Token tAreaErr3d, BIFF8:

Offset	Size	Contents
0	1	3D _H (tAreaErr3dR), 5D _H (tAreaErr3dV), 7D _H (tAreaErr3dA)
1	2	Index to REF entry in EXTERNSHEET record (→6.39) in the <i>Link Table</i> (→5.10.3)
3	8	Not used

4.10 Control Tokens

4.10.1 tExp (01_H)

This token is used to mark that a formula cell is part of an array formula or shared formula (→5.8). It is always the only token in a token array and is only allowed in the FORMULA record (→6.46). It contains the address of the base FORMULA record for this formula range. Note that in shared formulas this address may be different to the top-left cell of the formula range (if the top left cell is not part of the shared formula).

Token tExp, BIFF2:

Offset	Size	Contents
0	1	01 _H
1	2	Index to row of first FORMULA record in the formula range
3	1	Index to column of first FORMULA record in the formula range

Token tExp, BIFF3-BIFF8:

Offset	Size	Contents
0	1	01 _H
1	2	Index to row of first FORMULA record in the formula range
3	2	Index to column of first FORMULA record in the formula range

4.10.2 tTbl (02_H)

This token is used to mark that a formula cell is part of a multiple operation table (→5.9). It is always the only token in a token array and is only allowed in the FORMULA record (→6.46).

Token tTbl, BIFF2:

Offset	Size	Contents
0	1	02 _H
1	2	Index to first row of the table range
3	1	Index to first column of the table range

Token tTbl, BIFF3-BIFF8:

Offset	Size	Contents
0	1	02 _H
1	2	Index to first row of the table range
3	2	Index to first column of the table range

4.10.3 tParen (15_H)

Parentheses. This token is for display purposes only, it does not affect the result of the token array. If it follows an operator, the parentheses will enclose the operator and its operand(s), which in fact is the result of the enclosed operation.

Token tParen, BIFF2-BIFF8:

Offset	Size	Contents
0	1	15 _H

Examples for enclosed operands:

=3+(2) is represented by tInt(3), tInt(2), tParen, tAdd;

=(3)% is represented by tInt(3), tParen, tPercent;

=SUM(3,(2)) is represented by tInt(3), tInt(2), tParen, tFuncV(SUM).

Examples for enclosed operators:

=(3+2) is represented by tInt(3), tInt(2), tAdd, tParen;

=(3%) is represented by tInt(3), tPercent, tParen;

=(SUM(3,2)) is represented by tInt(3), tInt(2), tFuncV(SUM), tParen.

4.10.4 tExtended (18_H) (BIFF8)

2do

Operand class: *reference*.

4.10.5 tAttr (19_H)

Special attribute. This token is able to represent a variety of information, described in the following chapters.

Common structure of the tAttr token, BIFF2-BIFF8:

Offset	Size	Contents
0	1	19 _H
1	1	Attribute type flags: 01 _H = This is a <i>tAttrVolatile</i> token (volatile function) 02 _H = This is a <i>tAttrIf</i> token (IF function control) 04 _H = This is a <i>tAttrChoose</i> token (CHOOSE function control) 08 _H = This is a <i>tAttrSkip</i> token (skip part of token array) 10 _H = This is a <i>tAttrSum</i> token (SUM function with one parameter) 20 _H = This is a <i>tAttrAssign</i> token (assignment-style formula in a macro sheet) 40 _H = This is a <i>tAttrSpace</i> token (spaces and carriage returns, BIFF3-BIFF8) 41 _H = This is a <i>tAttrSpaceVolatile</i> token (BIFF3-BIFF8, see below)
2	var.	Additional information dependent on the attribute type

Theoretically, the attribute type flags may be combined to represent several attributes in one tAttr token. In practise, this is only done with the tAttrVolatile token and a leading tAttrSpace token. This “tAttrSpaceVolatile” token contains the attribute type 41_H (flags of the tAttrVolatile token and the tAttrSpace token). The data of the token is equivalent to a regular tAttrSpace token. This is possible, because the tAttrVolatile token does not contain additional valid data.

• Volatile Formula – The tAttrVolatile Token

If the formula contains a volatile function (a function that needs to be recalculated always, for example the NOW function, →4.11), or a defined name that directly or indirectly includes a volatile function, this token will lead the token array (except in assignment-style formulas in macro sheets, see tAttrAssign token below).

Token tAttrVolatile, BIFF2:

Offset	Size	Contents
0	1	19 _H
1	1	01 _H (identifier for the tAttrVolatile token)
2	1	Not used

Token tAttrVolatile, BIFF3-BIFF8:

Offset	Size	Contents
0	1	19 _H
1	1	01 _H (identifier for the tAttrVolatile token)
2	2	Not used

• IF Function Control – The tAttrIf Token

This token always follows the first parameter of the IF function (containing the condition). It contains a relative offset that allows to skip the *true* parameter of the function (second parameter), if the condition evaluates to *false*.

The token promises that the topmost operand on the formula stack (the last processed subexpression) is the condition of an IF function. This operand is popped from the stack, and processing continues depending on its value.

- If the condition is *true*, the following tokens are processed, and at the end of the parameter, a tAttrSkip token (see below) will occur that jumps right behind the tFuncVar(IF) token (this has to be assured during creation of the formula). Thus, the *false* parameter is not evaluated at all, and cannot cause an error code as final formula result, if it contains an error.
- If the condition is *false*, the following tokens are skipped (the *true* parameter), and evaluation continues with the *false* parameter. Again, at the end of the parameter, a tAttrSkip token will occur that jumps behind the tFuncVar(IF) token. If the *false* parameter does not exist, the tAttrIf token causes to jump to the tFuncVar(IF) token, and a simple *false* constant is pushed onto the formula stack.

If the IF function is processed, the result of either the *true* parameter or the *false* parameter will be on top of the formula stack.

Token tAttrIf, BIFF2:

Offset	Size	Contents
0	1	19 _H
1	1	02 _H (identifier for the tAttrIf token)
2	1	Distance (number of bytes) from start of next token (first token of second parameter) to start of first token of the <i>false</i> parameter (if extant); otherwise to start of the respective tFuncVar(IF) token (skipping all leading tAttrSpace tokens). See example below for more details.

Token tAttrIf, BIFF3-BIFF8:

Offset	Size	Contents
0	1	19 _H
1	1	02 _H (identifier for the tAttrIf token)
2	2	Distance (number of bytes) from start of next token (first token of second parameter) to start of first token of the <i>false</i> parameter (if extant); otherwise to start of the respective tFuncVar(IF) token (skipping all leading tAttrSpace tokens). See example below for more details.

Example of the formula = IF (TRUE , 1 , 2) . The formula contains spaces before the function identifier and before the third parameter.

Offset	Size	Token name	Description
0	2	tBool	The condition of the IF function: the constant <i>true</i>
2	4	tAttrIf	Distance to start of <i>false</i> parameter: 7 bytes
6	3	tInt	<i>True</i> parameter of the IF function: the constant 1
9	4	tAttrSkip	Distance to position behind tFuncVar(IF) token (minus 1): 11+8-1 = 18 bytes
13	4	tAttrSpace	Leading spaces for the next token (the <i>false</i> parameter)
17	3	tInt	<i>False</i> parameter of the IF function: the constant 2
20	4	tAttrSkip	Distance to position behind tFuncVar(IF) token (minus 1): 8-1 = 7 bytes
24	4	tAttrSpace	Leading spaces for the next token (the function identifier)
28	4	tFuncVar(IF)	Identifier of the IF function, function with 3 parameters

Example of the formula = IF (TRUE , 1) . The formula contains spaces before the function identifier.

Offset	Size	Token name	Description
0	2	tBool	The condition of the IF function: the constant <i>true</i>
2	4	tAttrIf	Distance to start of tFuncVar(IF) token: 7+4 = 11 bytes
6	3	tInt	<i>True</i> parameter of the IF function: the constant 1
9	4	tAttrSkip	Distance to position behind tFuncVar(IF) token (minus 1): 8-1 = 7 bytes
24	4	tAttrSpace	Leading spaces for the next token (the function identifier)
28	4	tFuncVar(IF)	Identifier of the IF function with 3 parameters

• **CHOOSE Function Control – The tAttrChoose Token**

This token always follows the first parameter of the CHOOSE function (containing the value which parameter to choose). It contains a relative offset that allows to skip all unused choices and to jump directly to the correct parameter.

The token promises that the topmost operand on the formula stack (the last processed subexpression) is the first parameter of a CHOOSE function. This operand is popped from the stack, and processing continues at the specified position in the token array. At the end of the parameter, a tAttrSkip token (see below) will occur that jumps right behind the tFuncVar(CHOOSE) token (this has to be assured during creation of the formula). Thus, only one more parameter of the function is evaluated at all, all other parameters cannot cause an error code as final formula result, if they contain an error. If the choice from the first parameter is wrong, the tAttrChoose token causes to jump to the tFuncVar(CHOOSE) token, and an error is generated.

Token tAttrChoose, BIFF2:

Offset	Size	Contents
0	1	19 _H
1	1	04 _H (identifier for the tAttrChoose token)
2	1	Number of choices in the CHOOSE function (<u>n_C</u> , number of parameters decreased by 1)
3	<u>n_C</u>	Jump table containing <u>n_C</u> distances (8-bit values specifying numbers of bytes) from start of <i>this jump table</i> to start of first token of the respective parameter of the CHOOSE function.
3+ <u>n_C</u>	1	Distance (numbers of bytes) from start of the <i>jump table</i> of this token to start of the tFuncVar(CHOOSE) token (skipping all leading tAttrSpace tokens). Used in case of an error (wrong choice in first parameter).

Token tAttrChoose, BIFF3-BIFF8:

Offset	Size	Contents
0	1	19 _H
1	1	04 _H (identifier for the tAttrChoose token)
2	2	Number of choices in the CHOOSE function ($\underline{n.c.}$, number of parameters decreased by 1)
4	$2 \cdot \underline{n.c.}$	Jump table containing $\underline{n.c.}$ distances (16-bit values specifying numbers of bytes) from start of <i>this jump table</i> to start of first token of the respective parameter of the CHOOSE function.
$4 + 2 \cdot \underline{n.c.}$	2	Distance (numbers of bytes) from start of the <i>jump table</i> of this token to start of the tFuncVar(CHOOSE) token (skipping all leading tAttrSpace tokens). Used in case of an error (wrong choice in first parameter).

Example of the formula = CHOOSE (2 , 1 , 2 , 3) . The formula contains spaces before the function identifier and before the third and fourth parameter.

Offset	Size	Token name	Description
0	2	tInt	The choice: the constant 2 (take second choice)
2	12	tAttrChoose	Function contains 3 choices ($\underline{n.c.} = 3$)
			<ul style="list-style-type: none"> • Jump table: distances to start positions of the next $\underline{n.c.} = 3$ parameters (including the size of the jump table of this token: $\underline{n.c.} \cdot 2 + 2 = 3 \cdot 2 + 2 = 8$ bytes): <ul style="list-style-type: none"> - First choice: 8 bytes (skip the jump table only) - Second choice: $8 + 7 = 15$ bytes - Third choice: $8 + 7 + 11 = 26$ bytes Distance to start of tFuncVar(CHOOSE) token (including jump table size): $8 + 7 + 11 + 11 + 4 = 41$ bytes
14	3	tInt	First choice parameter: the constant 1
17	4	tAttrSkip	Distance to position behind tFuncVar(CHOOSE) token (minus 1): $11 + 11 + 8 - 1 = 29$ bytes
21	4	tAttrSpace	Leading spaces for the next token
25	3	tInt	Second choice parameter: the constant 2
28	4	tAttrSkip	Distance to position behind tFuncVar(CHOOSE) token (minus 1): $11 + 8 - 1 = 18$ bytes
32	4	tAttrSpace	Leading spaces for the next token
36	3	tInt	Third choice parameter: the constant 3
39	4	tAttrSkip	Distance to position behind tFuncVar(CHOOSE) token (minus 1): $8 - 1 = 7$ bytes
43	4	tAttrSpace	Leading spaces for the next token (the function identifier)
47	4	tFuncVar(CHOOSE)	Identifier of the CHOOSE function with 4 parameters

• Jump to Position – The tAttrSkip Token

This token is used to let formula evaluation continue at another position in the token array. It contains the number of bytes to skip from the current position. For no obvious reason, this value is always one less than the actual number. This token is only used in conjunction with the tAttrIf token and tAttrChoose token (see tAttrIf and tAttrChosse tokens above for examples).

Token tAttrSkip, BIFF2:

Offset	Size	Contents
0	1	19 _H
1	1	08 _H (identifier for the tAttrSkip token)
2	1	Distance (number of bytes) from start of next token to destination position, decreased by 1

Token tAttrSkip, BIFF3-BIFF8:

Offset	Size	Contents
0	1	19 _H
1	1	08 _H (identifier for the tAttrSkip token)
2	2	Distance (number of bytes) from start of next token to destination position, decreased by 1

• Sum with One Parameter – The tAttrSum Token

This token replaces the tFuncVar(SUM) token, if the SUM function takes one parameter only. This is the only tAttr token that behaves like a real operator.

Token tAttrSum, BIFF2:

Offset	Size	Contents
0	1	19 _H
1	1	10 _H (identifier for the tAttrSum token)
2	1	Not used

Token tAttrSum, BIFF3-BIFF8:

Offset	Size	Contents
0	1	19 _H
1	1	10 _H (identifier for the tAttrSum token)
2	2	Not used

Example: The formula =SUM(1) is represented with the token array tInt(1), tAttrSum; instead of the token array tInt(1), tFuncVarV(SUM).

• Assignment in Macro Sheet – The tAttrAssign Token

In macro sheets it is possible to replace the SET.NAME function with an assignment-style formula. The usual syntax =SET.NAME(name,value) will be replaced with the command name=value then.

Example: The macro sheet function =SET.NAME(myname,1) is represented by the token array tNameV(myname), tInt(1), tFuncVarV(SET.NAME).

The used name is contained in a tName token.

The same can be done with the assignment-style formula myname=1. The token array of this function is tAttrAssign, tAttrStr("myname"), tAttrInt(1), tFuncVarV(SET.NAME).

Now, the used name is given as string in a tStr token, not in a tName token anymore.

Token tAttrAssign, BIFF2:

Offset	Size	Contents
0	1	19 _H
1	1	20 _H (identifier for the tAttrAssign token)
2	1	Not used

Token tAttrAssign, BIFF3-BIFF8:

Offset	Size	Contents
0	1	19 _H
1	1	20 _H (identifier for the tAttrAssign token)
2	2	Not used

In an assignment-style formulas the tAttrVolatile token will not be the first token in the formula, but it will follow the tStr token containing the name used in the assignment (as usual, this can also be a tAttrSpaceVolatile token).

Example: The assignment-style formula `myname=NOW()` is represented by the following token array: tAttrAssign, tStr("myname"), tAttrVolatile, tFuncV(NOW), tFuncVarV(SET.NAME).

See also the examples for the tAttrSpace token below.

• Spaces and Carriage Returns – The tAttrSpace Token (BIFF3-BIFF8)

This token is a placeholder for space characters and carriage returns inserted into the formula representation. It specifies the type and number of inserted characters preceding the next token. It does not affect the evaluation of the token array. In BIFF3, only spaces are allowed, and they can only occur in macro sheets and only in front of the entire formula, following the equality sign.

Several tAttrSpace tokens may occur before a token, for example mixed spaces and carriage returns. Parentheses and function calls can contain spaces at different positions:

- Parentheses: before the opening parenthesis and before the closing parenthesis.
- Function call: before the function name and before the closing parenthesis. It is not possible to add a space between function name and opening parenthesis.

The position of the added characters is specified inside the tAttrSpace token.

Token tAttrSpace, BIFF3:

Offset	Size	Contents
0	1	19 _H
1	1	40 _H (identifier for the tAttrSpace token), or 41 _H (identifier for the tAttrSpaceVolatile token)
2	2	Number of spaces following the equality sign (1...255)

Token tAttrSpace, BIFF4-BIFF8:

Offset	Size	Contents
0	1	19 _H
1	1	40 _H (identifier for the tAttrSpace token), or 41 _H (identifier for the tAttrSpaceVolatile token)
2	1	Type and position of the inserted character(s): 00 _H = Spaces before the next token (not allowed before tParen token) 01 _H = Carriage returns before the next token (not allowed before tParen token) 02 _H = Spaces before opening parenthesis (only allowed before tParen token) 03 _H = Carriage returns before opening parenthesis (only allowed before tParen token) 04 _H = Spaces before closing parenthesis (only allowed before tParen, tFunc, and tFuncVar tokens) 05 _H = Carriage returns before closing parenthesis (only allowed before tParen, tFunc, and tFuncVar tokens) 06 _H = Spaces following the equality sign (only in macro sheets)
3	1	Number of inserted spaces or carriage returns

In the following examples, the space characters are replaced by “°” characters, and a carriage return is shown as “<cr>”. The tAttrSpace tokens are shown using the following notation: tAttrSpace(type,count).

Formula	Token array
=° 1	tAttrSpace(00 _H ,1), tInt(1)
=° 1° ° +° ° ° ° 2	tAttrSpace(00 _H ,1), tInt(1), tAttrSpace(00 _H ,3), tInt(2), tAttrSpace(00 _H ,2), tAdd
=1° <cr> ° ° +2	tInt(1), tInt(2), tAttrSpace(00 _H ,1), tAttrSpace(01 _H ,1), tAttrSpace(00 _H ,2), tAdd
=1+° (2° °)	tInt(1), tInt(2), tAttrSpace(02 _H ,1), tAttrSpace(04 _H ,2), tParen, tAdd
=° SUM (1° °)	tInt(1), tAttrSpace(00 _H ,1), tAttrSpace(04 _H ,2), tAttrSum
=° NOW ()	tAttrSpaceVolatile(00 _H ,1), tFuncV(NOW)
Macro sheet formula	Token array
=° 1	tAttrSpace(06 _H ,1), tInt(1)
=° NOW ()	tAttrSpaceVolatile(06 _H ,1), tFuncV(NOW)
° myname=° ° 1	tAttrAssign, tAttrSpace(06 _H ,1), tStr("myname"), tAttrSpace(00 _H ,2), tInt(1), tFuncVarV(SET.NAME)
° myname=° ° NOW ()	tAttrAssign, tAttrSpace(06 _H ,1), tStr("myname"), tAttrSpaceVolatile(00 _H ,2), tFuncV(NOW), tFuncVarV(SET.NAME)

4.11 Built-In Sheet Functions

Meaning of the table column headers:

Column header	Meaning
Func ID	Function identifier used in tFunc (→4.7.1) and tFuncVar (→4.7.2) tokens
Function name	English name of the function
Min par	Minimum number of parameters this function requires
Max par	Maximum number of parameters this function allows. If equal to “Min par”, a tFunc token is used for this function, otherwise a tFuncVar token.
Ret class	Default token class of the return value (→4.2.2). R = <i>reference class</i> ; V = <i>value class</i> ; A = <i>array class</i> .
Parameter classes	Token classes expected by the respective parameters of the function (→4.2.3). The ellipsis “...” repeats the last specified token class for all following parameters.
Volatile	If marked, the function result is volatile (needs to be recalculated always). Existence of such a function causes the “Recalculate always” flag to be set in the ARRAY (→6.4), FORMULA (→6.46), TABLEOP (→6.101), and TABLEOP2 (→6.102) records.

4.11.1 Built-In Sheet Functions in BIFF2

Func ID	Function name	Min par	Max par	Ret class	Parameter classes	Volatile	Func ID	Function name	Min par	Max par	Ret class	Parameter classes	Volatile
0	COUNT	0	30	V	R ...		27	ROUND	2	2	V	V V	
1	IF	2	3	R	V R R		28	LOOKUP	2	3	V	V R R	
2	ISNA	1	1	V	V		29	INDEX	2	4	R	R V V V	
3	ISERROR	1	1	V	V		30	REPT	2	2	V	V V	
4	SUM	0	30	V	R ...		31	MID	3	3	V	V V V	
5	AVERAGE	1	30	V	R ...		32	LEN	1	1	V	V	
6	MIN	1	30	V	R ...		33	VALUE	1	1	V	V	
7	MAX	1	30	V	R ...		34	TRUE	0	0	V	–	
8	ROW	0	1	V	R		35	FALSE	0	0	V	–	
9	COLUMN	0	1	V	R		36	AND	1	30	V	R ...	
10	NA	0	0	V	–		37	OR	1	30	V	R ...	
11	NPV	2	30	V	V R ...		38	NOT	1	1	V	V	
12	STDEV	1	30	V	R ...		39	MOD	2	2	V	V V	
13	DOLLAR	1	2	V	V V		40	DCOUNT	3	3	V	R R R	
14	FIXED ⁸	2	2	V	V V		41	DSUM	3	3	V	R R R	
15	SIN	1	1	V	V		42	DAVERAGE	3	3	V	R R R	
16	COS	1	1	V	V		43	DMIN	3	3	V	R R R	
17	TAN	1	1	V	V		44	DMAX	3	3	V	R R R	
18	ARCTAN	1	1	V	V		45	DSTDEV	3	3	V	R R R	
19	PI	0	0	V	–		46	VAR	1	30	V	R ...	
20	SQRT	1	1	V	V		47	DVAR	3	3	V	R R R	
21	EXP	1	1	V	V		48	TEXT	2	2	V	V V	
22	LN	1	1	V	V		49	LINEST ⁹	1	2	A	R R	
23	LOG10	1	1	V	V		50	TREND ⁹	1	3	A	R R R	
24	ABS	1	1	V	V		51	LOGEST ⁹	1	2	A	R R	
25	INT	1	1	V	V		52	GROWTH ⁹	1	3	A	R R R	
26	SIGN	1	1	V	V		56	PV	3	5	V	V V V V V	

⁸ Parameter count changes in BIFF4.

⁹ Parameter count changes in BIFF3.

4 Formulas

Func ID	Function name	Min par	Max par	Ret class	Parameter classes	Volatile
57	FV	3	5	V	V V V V V	
58	NPER	3	5	V	V V V V V	
59	PMT	3	5	V	V V V V V	
60	RATE	3	6	V	V V V V V V	
61	MIRR	3	3	V	R V V	
62	IRR	1	2	V	R V	
63	RAND	0	0	V	-	•
64	MATCH	2	3	V	V R R	
65	DATE	3	3	V	V V V	
66	TIME	3	3	V	V V V	
67	DAY	1	1	V	V	
68	MONTH	1	1	V	V	
69	YEAR	1	1	V	V	
70	WEEKDAY ¹⁰	1	1	V	V	
71	HOUR	1	1	V	V	
72	MINUTE	1	1	V	V	
73	SECOND	1	1	V	V	
74	NOW	0	0	V	-	•
75	AREAS	1	1	V	R	
76	ROWS	1	1	V	R	
77	COLUMNS	1	1	V	R	
78	OFFSET	3	5	R	R V V V V	•
82	SEARCH	2	3	V	V V V	
83	TRANSPOSE	1	1	A	A	
86	TYPE	1	1	V	V	
97	ATAN2	2	2	V	V V	
98	ASIN	1	1	V	V	
99	ACOS	1	1	V	V	
100	CHOOSE	2	30	R	V R ...	
101	HLOOKUP ¹⁰	3	3	V	V R R	
102	VLOOKUP ¹⁰	3	3	V	V R R	
105	ISREF	1	1	V	R	
109	LOG	1	2	V	V V	
111	CHAR	1	1	V	V	
112	LOWER	1	1	V	V	
113	UPPER	1	1	V	V	
114	PROPER	1	1	V	V	
115	LEFT	1	2	V	V V	
116	RIGHT	1	2	V	V V	
117	EXACT	2	2	V	V V	
118	TRIM	1	1	V	V	
119	REPLACE	4	4	V	V V V V	
120	SUBSTITUTE	3	4	V	V V V V	
121	CODE	1	1	V	V	
124	FIND	2	3	V	V V V	
125	CELL	1	2	V	V R	•
126	ISERR	1	1	V	V	
127	ISTEXT	1	1	V	V	
128	ISNUMBER	1	1	V	V	
129	ISBLANK	1	1	V	V	
130	T	1	1	V	R	
131	N	1	1	V	R	
140	DATEVALUE	1	1	V	V	
141	TIMEVALUE	1	1	V	V	
142	SLN	3	3	V	V V V	
143	SYD	4	4	V	V V V V	
144	DDB	4	5	V	V V V V V	
148	INDIRECT	1	2	R	V V	•
162	CLEAN	1	1	V	V	
163	MDETERM	1	1	V	A	
164	MINVERSE	1	1	A	A	
165	MMULT	2	2	A	A A	
167	IPMT	4	6	V	V V V V V V	
168	PPMT	4	6	V	V V V V V V	
169	COUNTA	0	30	V	R ...	
183	PRODUCT	0	30	V	R ...	
184	FACT	1	1	V	V	
191	DPRODUCT	3	3	V	R R R	
192	ISNONTEXT	1	1	V	V	
193	STDEVP	1	30	V	R ...	
194	VARP	1	30	V	R ...	
195	DSTDEVP	3	3	V	R R R	
196	DVARP	3	3	V	R R R	
197	TRUNC ¹¹	1	1	V	V	
198	ISLOGICAL	1	1	V	V	
199	DCOUNTA	3	3	V	R R R	
204	USDOLLAR	1	2	V	V V	
205	FINDB	2	3	V	V V V	
206	SEARCHB	2	3	V	V V V	
207	REPLACEB	4	4	V	V V V V	
208	LEFTB	1	2	V	V V	
209	RIGHTB	1	2	V	V V	
210	MIDB	3	3	V	V V V	
211	LENB	1	1	V	V	
212	ROUNDUP	2	2	V	V V	
213	ROUNDDOWN	2	2	V	V V	
214	ASC	1	1	V	V	
215	DBSC	1	1	V	V	

¹⁰ Parameter count changes in BIFF5.

¹¹ Parameter count changes in BIFF3.

4.11.2 New Built-In Sheet Functions in BIFF3

Func ID	Function name	Min par	Max par	Ret class	Parameter classes	Vola tile	Func ID	Function name	Min par	Max par	Ret class	Parameter classes	Vola tile
49	LINEST ¹²	1	4	A	R R V V		228	SUMPRODUCT	1	30	V	A ...	
50	TREND ¹²	1	4	A	R R R V		229	SINH	1	1	V	V	
51	LOGEST ¹²	1	4	A	R R V V		230	COSH	1	1	V	V	
52	GROWTH ¹²	1	4	A	R R R V		231	TANH	1	1	V	V	
197	TRUNC ¹²	1	2	V	V V		232	ASINH	1	1	V	V	
219	ADDRESS	2	5	V	V V V V V		233	ACOSH	1	1	V	V	
220	DAYS360 ¹⁰	2	2	V	V V		234	ATANH	1	1	V	V	
221	TODAY	0	0	V	-	•	235	DGET	3	3	V	R R R	
222	VDB	5	7	V	V V V V V V V		244	INFO	1	1	V	V	
227	MEDIAN	1	30	V	R ...								

4.11.3 New Built-In Sheet Functions in BIFF4

Func ID	Function name	Min par	Max par	Ret class	Parameter classes	Vola tile	Func ID	Function name	Min par	Max par	Ret class	Parameter classes	Vola tile
14	FIXED ¹³	2	3	V	V V V		299	PERMUT	2	2	V	V V	
216	RANK	2	3	V	V R V		300	POISSON	3	3	V	V V V	
247	DB	4	5	V	V V V V V		301	TDIST	3	3	V	V V V	
252	FREQUENCY	2	2	A	R R		302	WEIBULL	4	4	V	V V V V	
261	ERROR.TYPE	1	1	V	V		303	SUMXMY2	2	2	V	A A	
269	AVEDEV	1	30	V	R ...		304	SUMX2MY2	2	2	V	A A	
270	BETADIST	3	5	V	V V V V V		305	SUMX2PY2	2	2	V	A A	
271	GAMMALN	1	1	V	V		306	CHITEST	2	2	V	A A	
272	BETAINV	3	5	V	V V V V V		307	CORREL	2	2	V	A A	
273	BINOMDIST	4	4	V	V V V V		308	COVAR	2	2	V	A A	
274	CHIDIST	2	2	V	V V		309	FORECAST	3	3	V	V A A	
275	CHIINV	2	2	V	V V		310	FTEST	2	2	V	A A	
276	COMBIN	2	2	V	V V		311	INTERCEPT	2	2	V	A A	
277	CONFIDENCE	3	3	V	V V V		312	PEARSON	2	2	V	A A	
278	CRITBINOM	3	3	V	V V V		313	RSQ	2	2	V	A A	
279	EVEN	1	1	V	V		314	STEYX	2	2	V	A A	
280	EXPONDIS	3	3	V	V V V		315	SLOPE	2	2	V	A A	
281	FDIST	3	3	V	V V V		316	TTEST	4	4	V	A A V V	
282	FINV	3	3	V	V V V		317	PROB	3	4	V	A A V V	
283	FISHER	1	1	V	V		318	DEVSQ	1	30	V	R ...	
284	FISHERINV	1	1	V	V		319	GEOMEAN	1	30	V	R ...	
285	FLOOR	2	2	V	V V		320	HARMEAN	1	30	V	R ...	
286	GAMMADIST	4	4	V	V V V V		321	SUMSQ	0	30	V	R ...	
287	GAMMAINV	3	3	V	V V V		322	KURT	1	30	V	R ...	
288	CEILING	2	2	V	V V		323	SKEW	1	30	V	R ...	
289	HYPGEOMVERT	4	4	V	V V V V		324	ZTEST	2	3	V	R V V	
290	LOGNORMDIST	3	3	V	V V V		325	LARGE	2	2	V	R V	
291	LOGINV	3	3	V	V V V		326	SMALL	2	2	V	R V	
292	NEGBINOMDIST	3	3	V	V V V		327	QUARTILE	2	2	V	R V	
293	NORMDIST	4	4	V	V V V V		328	PERCENTILE	2	2	V	R V	
294	NORMSDIST	1	1	V	V		329	PERCENTRANK	2	3	V	R V V	
295	NORMINV	3	3	V	V V V		330	MODE	1	30	V	A ...	
296	MNORMSINV	1	1	V	V		331	TRIMMEAN	2	2	V	R V	
297	STANDARDIZE	3	3	V	V V V		332	TINV	2	2	V	V V	
298	ODD	1	1	V	V								

¹² Function exists in BIFF2, but parameter count has been changed.

¹³ Function exists in BIFF2-BIFF3, but parameter count has been changed.

4.11.4 New Built-In Sheet Functions in BIFF5/BIFF7

Func ID	Function name	Min par	Max par	Ret class	Parameter classes	Volatile
70	WEEKDAY ¹⁴	1	2	V	V V	
101	HLOOKUP ¹⁴	3	4	V	V R R V	
102	VLOOKUP ¹⁴	3	4	V	V R R V	
220	DAYS360 ¹⁵	2	3	V	V V V	
336	CONCATENATE	0	30	V	V ...	
337	POWER	2	2	V	V V	
342	RADIANS	1	1	V	V	
343	DEGREES	1	1	V	V	
344	SUBTOTAL	2	30	V	V R ...	

Func ID	Function name	Min par	Max par	Ret class	Parameter classes	Volatile
345	SUMIF	2	3	V	R V R	
346	COUNTIF	2	2	V	R V	
347	COUNTBLANK	1	1	V	R	
350	ISPMT	4	4	V	V V V V	
351	DATEDIF	3	3	V	V V V	
352	DATESTRING	1	1	V	V	
353	NUMBERSTRING	2	2	V	V V	
354	ROMAN	1	2	V	V V	

4.11.5 New Built-In Sheet Functions in BIFF8/BIFF8X

Func ID	Function name	Min par	Max par	Ret class	Parameter classes	Volatile
358	GETPIVOTDATA	2	30			
359	HYPERLINK	1	2	V	V V	
360	PHONETIC	1	1	V	R	
361	AVERAGEA	1	30	V	R ...	
362	MAXA	1	30	V	R ...	

Func ID	Function name	Min par	Max par	Ret class	Parameter classes	Volatile
363	MINA	1	30	V	R ...	
364	STDEVPA	1	30	V	R ...	
365	VARPA	1	30	V	R ...	
366	STDEVA	1	30	V	R ...	
367	VARA	1	30	V	R ...	

¹⁴ Function exists in BIFF2-BIFF4, but parameter count has been changed.

¹⁵ Function exists in BIFF3-BIFF4, but parameter count has been changed.

5 Worksheet/Workbook Structure

In an Excel file the complex contents are split into several records. To keep the stream consistent, the position and order of the records is very important. This chapter contains details about the correct order and combination of the records inside the stream. The internal structure of the records is described in chapter 6.

A few record listings show, which records are required and which are optional. This is done in front of the record name. The following table shows all possible combinations.

•	RECORD NAME	A single required record
○	RECORD NAME	A single optional record
•	RECORD NAME 1 RECORD NAME 2	Exactly one of the specified records
○	RECORD NAME 1 RECORD NAME 2	Exactly one of the specified records, or no record
••	RECORD NAME	A required record list (record occurs 1 to n times)
○○	RECORD NAME	An optional record list (record occurs 0 to n times)
••	RECORD NAME 1 RECORD NAME 2	An unordered list of any of the specified records
○○	RECORD NAME 1 RECORD NAME 2	An unordered list of any of the specified records, or no record
•	<i>Record Group Name</i>	A required group of records
○	<i>Record Group Name</i>	An optional group of records
••	<i>Record Group Name</i>	A required list of record groups
○○	<i>Record Group Name</i>	An optional list of record groups
••	• RECORD NAME 1 • RECORD NAME 2	A required list of record groups, each group consists only of the specified records
○○	• RECORD NAME 1 • RECORD NAME 2	An optional list of record groups, each group consists only of the specified records
Uninteresting records in this context		

5.1 Worksheet/Workbook Stream

5.1.1 Worksheet Stream (BIFF2-BIFF4S)

The whole worksheet file consists of the worksheet stream. All records of the worksheet are enclosed by a leading BOF record (→6.8) and a trailing EOF record (→6.36). The sheet contents area contains all information about the worksheet, for instance sheet dimension, view settings, a font list, a list of defined names and external references, of course the contents and formats of all cells, row heights, column widths, drawing objects, chart objects, etc.

Common structure of a worksheet stream:

• BOF	Type = worksheet
Sheet records	
• EOF	

5.1.2 Workbook Stream (BIFF4W)

The whole BIFF4W workbook file consists of the workbook stream. It contains the global workbook data in the *Workbook Globals Substream*, and a list of worksheets, the *Worksheet Substreams*.

The *Workbook Globals Substream* contains common information about the workbook, for instance text encoding, global view settings or a list of all sheet names. Additionally, in each workbook a SHEETSOFFSET record (→6.93) is present.

The data of the sheets is stored in *Worksheet Substreams*, which are embedded in the *Workbook Globals Substream*. Each substream is preceded with a SHEETHDR record (→6.90) which contains the name of the sheet and the size of the following substream. The SHEETSOFFSET record mentioned above contains the stream position of the first SHEETHDR record. Each substream is a complete BIFF4S worksheet stream (→5.1.1).

Common structure of a workbook stream, BIFF4W:

•	<i>Workbook Globals Substream</i>
••	Embedded BIFF4S <i>Worksheet Substream(s)</i>

Common structure of the *Workbook Globals Substream*, BIFF4W:

•	BOF	Type = workbook globals
Global workbook records		
•	SHEETSOFFSET	Position of the first SHEETHDR record
Global workbook records		
••	•	SHEETHDR Sheet name, length of following substream
	•	Embedded BIFF4S <i>Worksheet Substream</i> , →5.1.1
•	EOF	

5.1.3 Workbook Stream (BIFF5-BIFF8)

From BIFF5 on an Excel document is stored as an OLE2 storage. The workbook stream is located in the root directory of the storage. In BIFF5/BIFF7 it is named “Book”, in BIFF8 “Workbook” (case sensitive). In difference to the BIFF4W workbook stream, the *Worksheet Substreams* are appended to the *Workbook Globals Substream*, not embedded. The substreams have similar contents as described for BIFF4W workbooks (→5.1.2).

Common structure of a workbook stream, BIFF5-BIFF8:

•	<i>Workbook Globals Substream</i>
••	<i>Worksheet Substream(s)</i>

Common structure of the *Workbook Globals Substream*, BIFF5-BIFF8:

• BOF	Type = workbook globals
	Global workbook records
• EOF	

Common structure of the *Worksheet Substream*, BIFF5-BIFF8:

• BOF	Type = worksheet
	Sheet records
• EOF	

5.2 Record Order in Substreams

Each of the following chapters describes the order of all records as they appear in a file written by Excel. In a few cases it is valid to slightly modify this order, but in general the order described here should be used when writing Excel files.

5.2.1 Record Order in BIFF2

•	BOF	Type = worksheet (→6.8)
○	FILEPASS	→5.19 (File Protection), →6.41
○	INDEX	→5.7 (<i>Row Blocks</i>), →6.55
○	CODEPAGE	→6.17
○	<i>Calculation Settings Block</i>	→5.3
○	PRINTHEADERS	→6.76
○	PRINTGRIDLINES	→6.75
○	DEFAULTROWHEIGHT	→6.28
○○	• FONT	→6.43
	○ EFONT	→6.35
○	<i>Page Settings Block</i>	→5.4
○	BACKUP	→6.5
○	<i>Link Table</i>	→5.10.1
○	• BUILTINFMTCOUNT	→6.13
	•• FORMAT	→6.45
○○	NAME	→6.66
○	<i>Worksheet Protection Block</i>	→5.18
○○	XF	→6.115
○	DEFCOLWIDTH	→6.29
○○	COLWIDTH	→6.20
•	DIMENSIONS	→6.31
○○	COLUMNDEFAULT	→6.19
○○	<i>Row Blocks</i>	→5.7
○○	NOTE	→6.67
○	WINDOW1	→6.108
•	<i>Worksheet View Settings Block</i>	→5.5
•	EOF	→6.36

5.2.2 Record Order in BIFF3

2do

5.2.3 Record Order in BIFF4S

2do

5.2.4 Record Order in BIFF4W

2do

5.2.5 Record Order in BIFF5/BIFF7

• Workbook Globals Substream

• BOF	Type = workbook globals (→6.8)
○ <i>File Protection Block</i>	→5.19
○ CODEPAGE	→6.17
○ TABID	
○ FNGROUPCOUNT	
○ <i>Global Link Table</i>	→5.10.2
○○ NAME	→6.66
○ <i>Workbook Protection Block</i>	→5.18
• WINDOW1	→6.108
○ BACKUP	→6.5
○ HIDEOBJ	→6.52
○ DATEMODE	→6.25
○ PRECISION	→6.74
○ BOOKBOOL	→6.9
•• FONT	→6.43
○○ FORMAT	→6.45
•• XF	→6.115
•• STYLE	→6.99
○ PALETTE	→6.70
•• BOUNDSHEET	→6.12
• EOF	→6.36

• **Worksheet Substream**

• BOF	Type = worksheet (→6.8)
○ UNCALCED	→6.104
○ INDEX	→5.7 (<i>Row Blocks</i>), →6.55
○ <i>Calculation Settings Block</i>	→5.3
○ PRINTHEADERS	→6.76
○ PRINTGRIDLINES	→6.75
○ GRIDSET	→6.48
○ GUTS	→6.49
○ DEFAULTTROWHEIGHT	→6.28
○ COUNTRY	→6.23
○ WSBOOL	→6.113
○ <i>Page Settings Block</i>	→5.4
○ <i>Local Link Table</i>	→5.10.2
○ <i>Worksheet Protection Block</i>	→5.18
○ DEFCOLWIDTH	→6.29
○○ COLINFO	→6.18
○ SORT	→6.95
• DIMENSIONS	→6.31
○○ <i>Row Blocks</i>	→5.7
○○ NOTE	→6.67
• <i>Worksheet View Settings Block</i>	→5.5
○ GCW	→6.47
○ STANDARDWIDTH	→6.97
• EOF	→6.36

5.2.6 Record Order in BIFF8

• Workbook Globals Substream

• BOF	Type = workbook globals (→6.8)
○ <i>File Protection Block</i>	→5.19
○ CODEPAGE	→6.17
○ DSF	→6.32
○ TABID	
○ FNGROUPCOUNT	
○ <i>Workbook Protection Block</i>	→5.18
• WINDOW1	→6.108
○ BACKUP	→6.5
○ HIDEOBJ	→6.52
○ DATEMODE	→6.25
○ PRECISION	→6.74
○ REFRESHALL	
○ BOOKBOOL	→6.9
•• FONT	→6.43
○○ FORMAT	→6.45
•• XF	→6.115
•• STYLE	→6.99
○ PALETTE	→6.70
○ USESELFS	→6.105
•• BOUNDSHEET	→6.12
○ COUNTRY	→6.23
○ <i>Link Table</i>	→5.10.3
○○ NAME	→6.66
○ <i>Shared String Table</i>	→5.11
• EOF	→6.36

• **Worksheet Substream**

• BOF	Type = worksheet (→6.8)
○ UNCALCED	→6.104
○ INDEX	→5.7 (<i>Row Blocks</i>), →6.55
○ <i>Calculation Settings Block</i>	→5.3
○ PRINTHEADERS	→6.76
○ PRINTGRIDLINES	→6.75
○ GRIDSET	→6.48
○ GUTS	→6.49
○ DEFAULTTROWHEIGHT	→6.28
○ WSBOOL	→6.113
○ <i>Page Settings Block</i>	→5.4
○ <i>Worksheet Protection Block</i>	→5.18
○ DEFCOLWIDTH	→6.29
○○ COLINFO	→6.18
○ SORT	→6.95
• DIMENSIONS	→6.31
○○ <i>Row Blocks</i>	→5.7
• <i>Worksheet View Settings Block</i>	→5.5
○ STANDARDWIDTH	→6.97
○○ MERGEDCELLS	→6.63
○ LABELRANGES	→6.60
○ PHONETIC	→6.73
○ <i>Conditional Formatting Table</i>	→5.12
○ <i>Hyperlink Table</i>	→5.13
○ <i>Data Validity Table</i>	→5.14
○ SHEETLAYOUT	→6.91 (BIFF8X only)
○ SHEETPROTECTION	Additional protection, →6.92 (BIFF8X only)
○ RANGEPROTECTION	Additional protection, →6.79 (BIFF8X only)
• EOF	→6.36

5.3 Worksheet Calculation Settings Block

The calculation settings for a worksheet are stored in several records in the *Worksheet Substream*, called the *Calculation Settings Block*. Note that in BIFF4W-BIFF8 the settings are global for the entire document, but the record block occurs equally in every *Worksheet Substream* (→5.2).

Structure of the *Calculation Settings Block* in the *Worksheet Substream*:

○	CALCCOUNT	→6.14
○	CALCMODE	→6.15
○	PRECISION	→6.74 (moved to <i>Workbook Globals Substream</i> in BIFF5-BIFF8)
○	REFMODE	→6.80
○	DELTA	→6.30
○	ITERATION	→6.57
○	DATEMODE	→6.25 (moved to <i>Workbook Globals Substream</i> in BIFF5-BIFF8)
○	SAFERECALC	→6.85 (BIFF3-BIFF8 only)

5.4 Page Settings Block

The records in the *Page Settings Block* describe options and settings for printing. These settings are valid for the current worksheet.

Structure of the *Page Settings Block* in the *Worksheet Substream*:

○	HORIZONTALPAGEBREAKS	→6.54
○	VERTICALPAGEBREAKS	→6.107
○	HEADER	→6.51
○	FOOTER	→6.44
○	HCENTER	→6.50 (BIFF3-BIFF8 only)
○	VCENTER	→6.106 (BIFF3-BIFF8 only)
○	LEFTMARGIN	→6.62
○	RIGHTMARGIN	→6.81
○	TOPMARGIN	→6.103
○	BOTTOMMARGIN	→6.11
○	PLS	
○	SETUP	→6.89 (BIFF4-BIFF8 only)
○	BITMAP	→6.6 (BIFF8 only)

5.5 Worksheet View Settings Block

The view settings for a worksheet are stored in several records in the *Worksheet Substream*, called the *Worksheet View Settings Block*.

Structure of the *Worksheet View Settings Block* in the *Worksheet Substream*:

●	WINDOW2	→6.109
○	SCL	→6.87 (BIFF4-BIFF8 only)
○	PANE	→6.71
○○	SELECTION	→6.88

5.6 Cell Formatting

All cell formatting attributes are stored in XF records (→6.115). The cell records themselves contain an index into the XF record list. This way of storing cell formatting saves memory and decreases the file size.

5.6.1 Cell Formatting and Cell Styles

The XF record is able to store explicit cell formatting attributes or the attributes of a cell style. Explicit formatting includes the reference to a cell style XF record. This allows to extend a defined cell style with some explicit attributes.

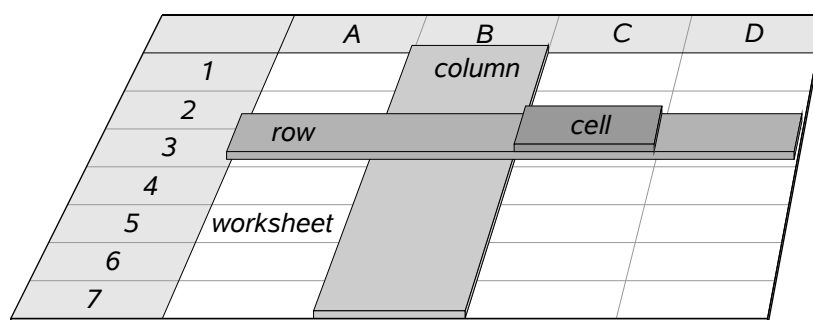
The formatting attributes are divided into 6 groups:

Group	Attributes
Number format	Number format index (index to FORMAT record)
Font	Font index (index to FONT record)
Alignment	Horizontal and vertical alignment, text wrap, indentation, orientation/rotation, text direction
Border	Border line styles and colours
Background	Background area style and colours
Protection	Cell locked, formula hidden

For each group a flag in the cell XF record specifies whether to use the attributes contained in that XF record or in the referenced style XF record. In style XF records, these flags specify whether the attributes will overwrite explicit cell formatting when the style is applied to a cell. Changing a cell style (without applying this style to a cell) will change all cells which already use that style and do not contain explicit cell attributes for the changed style attributes. If a cell XF record does not contain explicit attributes in a group (if the attribute group flag is not set), it repeats the attributes of its style XF record.

5.6.2 Default Formatting

Default formatting is applied to all cells which are not described by a cell record. In this case the default format of the row or column may be used (if defined). If an undefined cell contains a row and a column default format, the row format will overwrite the column format. If there are no row and column default formats available, the worksheet/workbook default cell format will be used.



Default column formatting is stored in the COLUMNDEFAULT record (BIFF2, →6.19) or in the COLINFO record (BIFF3-BIFF8, →6.18). The ROW record (→6.83) contains the default format of a specific row. The default cell format is always present in an Excel file, described by the XF record with the fixed index 15 (0-based). By default, it uses the worksheet/workbook default cell style, described by the very first XF record (index 0).

5.7 Cell Table

The Cell Table describes all cells in a worksheet, together with all row and column settings for that sheet.

Structure of the *Cell Table* in the *Worksheet Substream*, BIFF2:

○	DEFAULTROWHEIGHT	→6.28
○	DEFCOLWIDTH	→6.29
○○	COLWIDTH	→6.20
●	DIMENSIONS	→6.31
○○	COLUMNDEFAULT	→6.19
○○	<i>Row Block(s)</i>	→5.7.1

Structure of the *Cell Table* in the *Worksheet Substream*, BIFF3-BIFF8:

○	DEFAULTROWHEIGHT	→6.28
○	DEFCOLWIDTH	→6.29
○○	COLINFO	→6.18
●	DIMENSIONS	→6.31
○○	<i>Row Block(s)</i>	→5.7.1
○	STANDARDWIDTH	→6.97 (BIFF4-BIFF8 only)

All cells in an Excel document are divided into blocks of 32 consecutive rows, called *Row Blocks*. The first *Row Block* starts with the first used row in that sheet. Inside each *Row Block* there will occur ROW records describing the properties of the rows, and cell records with all the cell contents in this *Row Block*.

Example: The first used cell in the sheet is located in row 10, or row 10 is the first formatted row. The first *Row Block* will contain the rows 10...41, the second *Row Block* will contain the rows 42...73, and so on.

5.7.1 Structure of a Row Block

Each *Row Block* contains ROW records (→6.83) describing the row properties (in ascending order), followed by all cell records in this block. The cell records are stored row by row (ascending), and in each row from left to right. A *used row* either contains any (filled or formatted) cells or is changed in another way (for instance height or default formatting). If a row is not used, there will not occur a ROW record in the *Row Block*.

Structure of a *Row Block* in the *Worksheet Substream*, BIFF2-BIFF4:

●●	ROW	Properties of all used rows
○○	<i>Cell Block(s)</i>	Cell records for all used cells

Structure of a *Row Block* in the *Worksheet Substream*, BIFF5-BIFF8:

●●	ROW	Properties of the used rows
○○	<i>Cell Block(s)</i>	Cell records for all used cells
○	DBCELL	Stream offsets to the cell records of each row

A *Cell Block* is in most cases simply a single cell record. In BIFF2 it may be preceded by an IXFE record containing the index to an XF record.

Structure of a common *Cell Block* in a *Row Block*, BIFF2-BIFF8:

○	IXFE	Index to XF (BIFF2 only, →6.58)
	BLANK	→6.7
	BOOLERR	→6.10
	INTEGER	→6.56 (BIFF2 only)
	LABEL	→6.59 (BIFF2-BIFF7)
	LABELSST	→6.61 (BIFF8 only)
●	MULBLANK	→6.64 (BIFF5-BIFF8)
	MULRK	→6.65 (BIFF5-BIFF8)
	NUMBER	→6.68
	RK	→6.82 (BIFF3-BIFF8)
	RSTRING	→6.84 (BIFF5/BIFF7)

If the cell contains a formula, there may occur additional records for array formulas, shared formulas, multiple operation tables, and/or the result string.

Structure of a *Formula Cell Block* in a *Row Block*, BIFF2-BIFF8:

○	IXFE	Index to XF (BIFF2 only, →6.58)
●	FORMULA	→6.46
	ARRAY	Top left cell of an array formula (→5.8, →6.4)
	SHRFMLA	Top left cell of a shared formula (BIFF5-BIFF8, →5.8, →6.94)
○	TABLEOP	Top left cell of a multiple operations table (→5.9, →6.101)
	TABLEOP2	Top left cell of a multiple operations table (→5.9, →6.102)
○	STRING	Formula cell returns a string value (→6.98)

• **Finding Cells in a Row Block (BIFF2-BIFF4)**

In BIFF2-BIFF4, the ROW record contains a stream offset, pointing to the cell records of the respective row. The offset value of the first ROW record in a *Row Block* represents the difference between the start of the second ROW record and the first cell record. The offset of the second ROW record contains the size of all cell records of the first row, and so on. If a row does not contain any cells, the offset to the cells in the next filled row is stored.

Example: A *Row Block* contains one cell per row in the rows 10, 11, and 12. Row 13 is not used, the height of row 14 is changed (but it does not contain a cell). Row 15 contains a cell again. In this example, the first column shows the absolute stream position of each record.

048E _H	ROW	Row = 10, offset = 0050 _H
04A2 _H	ROW	Row = 11, offset = 000E _H
04B6 _H	ROW	Row = 12, offset = 0012 _H
04CA _H	ROW	Row = 14, offset = 000E _H (no record for row 13)
04DE _H	ROW	Row = 15, offset = 0000 _H
04F2 _H	RK	Address = A10
0500 _H	NUMBER	Address = A11
0512 _H	RK	Address = A12
0520 _H	NUMBER	Address = A15

The base position for the calculation of all cell record positions is the start of the second ROW record, 04A2_H. This is for convenience: After reading the first ROW record it is possible to calculate the position of the first cell record, using the current stream position and the offset contained in the ROW record.

- The first ROW record describes row 10. The cell records of this row start in 04A2_H + 0050_H = 04F2_H.
- The second ROW record describes row 11. The cell records start in 04A2_H + 0050_H + 000E_H = 0500_H.
- The third ROW record describes row 12. The cell records of row 12 start in 04A2_H + 0050_H + 000E_H + 0012_H = 0512_H.
- The fourth ROW record describes row 14. It does not contain cell records. So the offset points to the cells of the next used row (row 15).
The stream position is 04A2_H + 0050_H + 000E_H + 0012_H + 000E_H = 0520_H.
- The last ROW record record describes row 15. Because for row 14 the stream position for row 15 is calculated already, this ROW record contains the offset 0000_H. The stream position evaluates to the correct value 04A2_H + 0050_H + 000E_H + 0012_H + 000E_H + 0000_H = 0520_H.

If the size of all cell records of a row exceeds FFFF_H, the respective ROW record will contain the offset 0000_H. From this row on, the offsets cannot be used anymore to calculate stream positions.

• Finding Cells in a Row Block (BIFF5-BIFF8)

In BIFF5-BIFF8 the DBCELL record (→6.26) follows the cell records and contains exactly the stream offsets which the ROW records would contain in BIFF2-BIFF4. In the DBCELL record this offset list is headed by an offset to the first ROW record in this *Row Block*. This offset is a positive value, although it points backwards to an earlier stream position. While no record would occur in BIFF2-BIFF4 for empty *Row Blocks*, in BIFF5-BIFF8 a DBCELL record is written for each and every *Row Block*.

Example: A *Row Block* contains one cell per row in the rows 10, 11, and 12. Row 13 is not used, the height of row 14 is changed (but it does not contain a cell). Row 15 contains a cell again. In this example, the first column shows the absolute stream position of each record.

07B2 _H	ROW	Row = 10 (Record size with header = 0014 _H)
07C6 _H	ROW	Row = 11
07DA _H	ROW	Row = 12
07EE _H	ROW	Row = 14 (no record for row 13)
0802 _H	ROW	Row = 15
0816 _H	RK	Address = A10
0824 _H	NUMBER	Address = A11
0836 _H	RK	Address = A12
0844 _H	NUMBER	Address = A15
0856 _H	DBCELL	Offset to first ROW record = 00A4 _H Offsets = 0050 _H , 000E _H , 0012 _H , 000E _H , 0000 _H

The base position for the calculation of all cell record positions is the start of the second ROW record, 07C6_H. It is calculated from the position of the DBCELL record (contained in the INDEX record, →5.7.2), the first offset this DBCELL record contains, and the size of the first ROW record:

$$0856_{\text{H}} - 00A4_{\text{H}} + 0014_{\text{H}} = 07B2_{\text{H}} + 0014_{\text{H}} = 07C6_{\text{H}}$$

The calculation of the cell record positions works equally to the example for BIFF2-BIFF4 above (using 07C6_H as base stream position).

If the size of all cell records of a row exceeds FFFF_H, the respective position in the DBCELL record will contain the offset 0000_H. From this point on, the offsets cannot be used anymore to calculate stream positions.

5.7.2 Finding Row Blocks in a Worksheet

Sometimes it may be useful to jump directly to a specific cell in the worksheet stream. The first step is to find the *Row Block* which contains the cell. How to find a cell inside the *Row Block* is described in the next chapter.

The INDEX record (→6.55) stores the index to the first used row and stream offsets for each *Row Block* in the worksheet.

• **Finding Row Blocks in BIFF2-BIFF4**

In BIFF2-BIFF4 the INDEX record contains an array with the stream positions of the first ROW record of each *Row Block*. This array contains also positions for empty *Row Blocks*. In this case the position of the next extant *Row Block* is used.

Example: A worksheet contains data in the rows 2, 4, 65, and 100. The range of used rows in the INDEX record is 2...101 (last used row + 1). There are 4 *Row Blocks* in the sheet: 2...33, 34...65, 66...97, and 98...130. The third *Row Block* is empty.

0000 _H	BOF	Type = worksheet
Sheet records		
002E _H	INDEX	Row range = 2...101 Offsets = 049A _H , 04DE _H , 0500 _H , 0500 _H
Sheet records		
049A _H	ROW	Row = 2
04AE _H	ROW	Row = 4
Cell records for this <i>Row Block</i>		
04DE _H	ROW	Row = 65
Cell records for this <i>Row Block</i>		
0500 _H	ROW	Row = 100
Cell records for this <i>Row Block</i>		
Sheet records		
EOF		

Because the third *Row Block* is empty, the third offset in the INDEX record array points to the fourth *Row Block*.

• Finding Row Blocks in BIFF5-BIFF8

In BIFF5-BIFF8 the INDEX record contains an array with the stream positions of the DBCELL record (→6.26) following the cell records in each *Row Block*. Because the DBCELL record is mandatory, a *Row Block* cannot be empty anymore.

Example: A worksheet contains data in the rows 2, 4, 65, and 100. The range of used rows in the INDEX record is 2...101 (last used row + 1). There are 4 *Row Blocks* in the sheet: 2...33, 34...65, 66...97, and 98...130. The third *Row Block* is empty except its DBCELL record.

0835 _H	BOF	Type = worksheet
0849 _H	INDEX	Row range = 2...101 Offsets = 09A9 _H , 09D7 _H , 09E1 _H , 0A0B _H
		Sheet records
	ROW	Row = 2
	ROW	Row = 4
		Cell records for this <i>Row Block</i>
09A9 _H	DBCELL	
	ROW	Row = 65
		Cell records for this <i>Row Block</i>
09D7 _H	DBCELL	
09E1 _H	DBCELL	
	ROW	Row = 100
		Cell records for this <i>Row Block</i>
0A0B _H	DBCELL	
		Sheet records
		EOF

5.8 Array Formulas, Shared Formulas

Array formulas (BIFF2-BIFF8) and shared formulas (BIFF5-BIFF8) are formulas which span over a range of cells. Array formulas are handled different from single cell formulas in a spreadsheet. Shared formulas are only an optimisation to decrease the file size, they are not distinguishable from other cell formulas. Naturally an array formula cannot be a shared formula at the same time. Shared formulas are created for instance when filling a cell range from a single formula cell.

In general an array or shared formula is stored only once in a file, either in the ARRAY record (→6.4) for array formulas, or in the SHRFMLA record (→6.94) for shared formulas. These records are part of the *Formula Cell Block* (→5.7.1). They immediately follow the first FORMULA record (→6.46) for this range¹⁶. All array or shared formula cells contain a reference to the formula data. This reference (tExp token, →4.10.1) consists of the cell address of the top left cell of the range. In this way each formula cell can be associated with its formula data.

If a formula returns a string value, a STRING record (→6.98) follows the FORMULA record normally. In the case of array and shared formulas, this STRING record follows the ARRAY or SHRFMLA record.

Example: A document contains, among other cells, an array formula in A2:B3, a single formula cell in D2, and a shared formula in F2:F3, which returns string values.

ROW	Row = 1 (row 2 in user interface)
ROW	Row = 2 (row 3 in user interface)
FORMULA	Address = A2, is array formula, cell range origin = A2
ARRAY	Range = A2:B3, token array of the array formula
FORMULA	Address = B2, is array formula, cell range origin = A2
FORMULA	Address = D2, token array of the formula
FORMULA	Address = F2, is shared formula, cell range origin = F2
SHRFMLA	Range = F2:F3, token array of the shared formula
STRING	Result of previous formula (cell F2)
	Cell records
FORMULA	Address = A3, is array formula, cell range origin = A2
FORMULA	Address = B3, is array formula, cell range origin = A2
FORMULA	Address = F3, is shared formula, cell range origin = F2
STRING	Result of previous formula (cell F3)
	Cell records

¹⁶ For shared formulas the first FORMULA record may not be the top-left cell of the range. It is possible to overwrite single cells of a shared formula range without invalidating the shared formula itself (the remaining formula cells).

5.9 Multiple Operation Tables

A multiple operation table is a cell range filled with results of a given series of formulas and input values. Three different kinds of multiple operations are supported:

- 1) A column with formulas is aggregated with a row of input values.
- 2) A row with formulas is aggregated with a column of input values.
- 3) A column and a row of input values are aggregated using one formula.

The formulas and input values are not included in the actual operation table. They must be located in the column left of the table and in the row above the table. In case 3), the single formula is in the cell at top left of the table (outside).

All formulas should refer to a specific cell, the “input cell”. On calculation, this cell reference is replaced by the respective value from the input value range in each formula. If the input values are located in the column, the input cell is called “column input cell”, otherwise “row input cell”. In case 3) there is a column value range and a row value range. Therefore both kinds of input cells are required here. The following examples show how multiple operation tables work.

Example for a multiple operation table in the range C2:D3, with formulas in a row (C1:D1 implicitly), and input values in the column (B2:B3 implicitly). The table has been created with A1 as input cell. The highlighted range is the operation table. The example shows the calculated formulas, but they are not really contained there.

	A	B	C	D	E
1	Input cell		=A1^2	=SQRT(A1)	
2		4	=B2^2	=SQRT(B2)	
3		9	=B3^2	=SQRT(B3)	

Example for a multiple operation table in the range C2:D3, with input values in column and row (C1:D1 and B2:B3 implicitly, formula in B1). The table has been created with A1 as row input cell and A2 as column input cell. Again, the highlighted range is the operation table.

	A	B	C	D	E
1	Row input cell	=A1^A2	4	5	
2	Column input cell	2	=C1^B2	=D1^B2	
3		3	=C1^B3	=D1^B3	

Similar to array and shared formulas a multiple operation table is described only once in a file. The TABLEOP record (→6.101) is used for that, and possibly the TABLEOP2 record (→6.102) in BIFF2. These records are part of the *Formula Cell Block* (→5.7.1). They immediately follow the first FORMULA record (→6.46) for this table. Each cell in the table (which are all formula cells) contain a single tTbl token (→4.10.2) only. The token consists of the cell address of the top left cell of the table. In this way each cell can be associated with the description of the table.

If a formula returns a string value, a STRING record (→6.98) follows the FORMULA record normally. In the case of multiple operation table, this STRING record follows the TABLEOP or TABLEOP2 record.

Example: A document contains, among other cells, a multiple operation table in C2:D3.

ROW	Row = 1 (row 2 in user interface)
ROW	Row = 2 (row 3 in user interface)
FORMULA	Address = C2, is multiple operation, cell range origin = C2
TABLEOP	Range = C2:D3, table mode, input cell
FORMULA	Address = D2, is multiple operation, cell range origin = C2
	Cell records
FORMULA	Address = C3, is array formula, cell range origin = C2
FORMULA	Address = D3, is array formula, cell range origin = C2
	Cell records

5.10 Internal and External References

This chapter describes all types of 3D and external references. In detail, this could be:

- A reference to a cell or a cell range of another sheet in the same workbook (3D reference);
- A reference to a cell or a cell range of a sheet in another workbook (external reference);
- A reference to a global or local defined name (internal name);
- A reference to a defined name in another workbook (external name);
- An external function (add-in, sheet macro, Basic macro);
- A DDE link;
- An OLE object link.

For external references and external names a combination of XCT and CRN records will occur. These records store values of cells of the document. If the external document cannot be found, these values will be used to get the result of an external reference. An XCT record (→6.114) contains the number of following CRN records. A CRN record (→6.24) stores the contents of one cell or a sequence of cells of one row. Fragmentary cell ranges or cell ranges spanning over more than one row are split into several CRN records. 3D references do not use these records because the referenced cells are located in the own document.

It is possible to determine whether the cached values (the XCT and CRN records) will be stored in the file. In BIFF3 and BIFF4 this option is stored in the WSBOOL record (→6.113). From BIFF5 on the optional record BOOKBOOL (→6.9) contains this option.

For all the following examples an external document “example.xls” is used. It contains 3 sheets named “ExtSheet1”, “ExtSheet2” and “ExtSheet3”.

5.10.1 References in BIFF2-BIFF4

2do

5.10.2 References in BIFF5/BIFF7

The data of references is spread in several *Link Tables*. The *Global Link Table* is located in the *Workbook Globals Substream*, and the *Local Link Tables* are located in the *Worksheet Substreams* (→5.2.5).

• Global Link Table

The *Global Link Table* contains reference data used by internal defined names. It is located in the *Workbook Globals Substream* and will be followed by the list of NAME records containing definitions of all internal defined names. If no names are defined in the document, or none of the defined names needs any reference data, the *Global Link Table* will not occur.

• Local Link Table

The reference data used in sheets (for example cell formulas) is stored in *Local Link Tables* inside the *Worksheet Substreams*. Each worksheet that uses references contains its own *Local Link Table*.

• Structure of the Link Table

Common structure of the *Link Table*, BIFF5/BIFF7:

• EXTERNCOUNT	→6.37
•• EXTERNSHEET Block(s)	Settings for a referenced worksheet or document

There are three different kinds of *EXTERNSHEET Blocks*, all starting with an EXTERNSHEET record.

The *Internal EXTERNSHEET Sheet Block* represents a referenced worksheet in the own document. The EXTERNSHEET record contains the sheet name only. There will not be any other records following it.

Common structure of an *Internal EXTERNSHEET Sheet Block*, BIFF5/BIFF7:

• EXTERNSHEET	→6.39
---------------	-------

The *External EXTERNSHEET Sheet Block* represents a referenced worksheet in an external document. The EXTERNSHEET record contains the URL of the document together with the sheet name. It is followed by cached cell contents if enabled (see record BOOKBOOL, →6.9).

Common structure of an *External EXTERNSHEET Sheet Block*, BIFF5/BIFF7:

• EXTERNSHEET	→6.39
○ XCT	→6.114
○○ CRN	→6.24

The *EXTERNSHEET Document Block* represents a complete external document (for example used for external names, add-in functions, or DDE links). The EXTERNSHEET record only contains the URL of the document without a sheet name. It is followed by external name definitions.

Common structure of an *EXTERNSHEET Document Block*, BIFF5/BIFF7:

• EXTERNSHEET	→6.39
•• EXTERNNAME	→6.38

• External and 3D References

External and 3D references are represented in a formula by the tokens tRef3d (→4.9.15) or tArea3d (→4.9.16). These tokens contain an index to an EXTERNSHEET record located in the own *Local Link Table* and indexes to the first and last referenced sheet.

For 3D references, the tokens contain a negative EXTERNSHEET index, indicating a reference into the own workbook. The absolute value is the *one-based* index of the EXTERNSHEET record that contains the name of the first sheet. The tokens additionally contain absolute indexes of the first and last referenced sheet. These indexes are independent of the EXTERNSHEET record list. If the referenced sheets do not exist anymore, these indexes contain the value FFFF_H (3D reference to a deleted sheet), and an EXTERNSHEET record with the special name “<04_H>” (own document) is used.

Each external reference contains the positive *one-based* index to an EXTERNSHEET record containing the URL of the external document and the name of the sheet used. The sheet index fields of the tokens are not used.

Example: A document with 7 sheets (named from “Sheet1” to “Sheet7”) contains on “Sheet1” the formulas

```
=Sheet2!A1,
=Sheet1!A1,
=SUM(Sheet4:Sheet6!A1:B3),
=SUM([example.xls]ExtSheet1!A1:B2) (contents: A1=1.11, B1=2.22, A2=3.33, B2=4.44),
=[example.xls]ExtSheet3!A1 (contents: “ABCD”), and
=Sheet8!A1.
```

The *Local Link Table* of Sheet1 contains 7 EXTERNSHEET Sheet Blocks:

EXTERNCOUNT	Number of EXTERNSHEET records = 8
EXTERNSHEET 1	Name = “<03 _H >Sheet2”
EXTERNSHEET 2	Name = “<02 _H >” (own sheet, →3.9.1)
EXTERNSHEET 3	Name = “<03 _H >Sheet4”
EXTERNSHEET 4	Name = “<03 _H >Sheet6”
EXTERNSHEET 5	Name = “<01 _H >[example.xls]ExtSheet1”
XCT	Number of CRN = 2
CRN 0	Cell range = A1:B1, contents = 1.11, 2.22
CRN 1	Cell range = A2:B2, contents = 3.33, 4.44
EXTERNSHEET 6	Name = “<01 _H >[example.xls]ExtSheet3”
XCT	Number of CRN = 1
CRN 0	Cell range = A1, contents = “ABCD”
EXTERNSHEET 7	Name = “<04 _H >” (own workbook, →3.9.1)

Inside of the first formula the cell reference is represented by the token tRef3d (→4.9.15). The third formula contains the token tArea3d (→4.9.16). The last formula contains a tRef3d token which refers to the last EXTERNSHEET record and contains FFFF_H as sheet indexes.

• Internal Names

All internal names are stored in the *Global Link Table* in a list of NAME records (→6.66). There exist two types of internal names: global names which are valid in the whole workbook and local names which are attached to a specific sheet. For instance the local name “MyCell” of the sheet “Sheet1” can be used from everywhere in the workbook by entering =Sheet1!MyCell. Each NAME record contains the name itself and a *one-based* index to the preceding EXTERNSHEET records to indicate a local name, or the index zero to indicate a global name.

Inside of a formula a global name or a local name of the own sheet is represented by the token tName (→4.9.1) with a *one-based* index to the NAME record list. Local names from other sheets (with explicitly denoted sheet name) are represented by the token tNameX (→4.9.14) with an index to a special EXTERNSHEET record and an index to the NAME record list.

Example: A document contains the global name “GlobalName”, and the local names “Sheet1!LocalName” and “Sheet2!LocalName”. In “Sheet1” there are the formulas

```
=GlobalName,  
=LocalName,  
=Sheet1!LocalName, and  
=Sheet2!LocalName.
```

The *Global Link Table* (together with the list of NAME records) contains the following records:

EXTERNCOUNT	Number of EXTERNSHEET records = 2
EXTERNSHEET 1	Name = “<03 _H >Sheet1”
EXTERNSHEET 2	Name = “<03 _H >Sheet2”
NAME 1	Name = “GlobalName”, EXTERNSHEET = 0 (Global)
NAME 2	Name = “LocalName”, EXTERNSHEET = 1 (Sheet1)
NAME 3	Name = “LocalName”, EXTERNSHEET = 2 (Sheet2)

The *Local Link Table* of Sheet1 contains the following records:

EXTERNCOUNT	Number of EXTERNSHEET records = 1
EXTERNSHEET 1	Name = “<04 _H >” (Unspecified sheet in own workbook, →3.9.1)

The two former formulas contain a tNameV token each. The token in the first formula refers to NAME 1 and the token in the second formula to NAME 2.

The two latter formulas contain a tNameXV token each. Both contain a reference to EXTERNSHEET 1 in the *Local Link Table*. The token in the third formula refers to NAME 2 and the last formula refers to NAME 3.

• External Names

In Excel, formulas can use defined names located in another workbook, called “external names”. The definitions for external names are stored in a combination of *EXTERNSHEET Sheet Blocks* and *EXTERNSHEET Document Blocks*. The *EXTERNSHEET Sheet Blocks* contain the source document and sheet names the external names are located in, and the cached cell contents. Following a single *EXTERNSHEET Document Block* that only contains the source document name and the external name definitions themselves. This is repeated for each external document.

Inside of a formula an external name is represented by the token tNameX (→4.9.14). It contains the *one-based* index to the EXTERNSHEET record (of the *EXTERNSHEET Document Block*) and a *one-based* index to an EXTERNNAME record inside this block. Each EXTERNNAME record refers to a preceding EXTERNSHEET record containing the sheet name.

Example: A document contains the formulas

```
=example.xls!GlobalName (location: ExtSheet3!C33; contents: 33),
=[example.xls]ExtSheet3!LocalName (location: ExtSheet1!B22; contents: “ABCD”),
=[example.xls]ExtSheet1!LocalName (location: ExtSheet1!A11; contents: 11), and
=another_example.xls!GlobalName (location: ExtSheet2!B2; contents: 2).
```

The *Local Link Table* of Sheet1 contains two *EXTERNSHEET Sheet Blocks* followed by an *EXTERNSHEET Document Block* for the document “example.xls”; and one *EXTERNSHEET Sheet Block* and one *EXTERNSHEET Document Block* for the document “another_example.xls”:

EXTERNCOUNT	Number of EXTERNSHEET records = 5
EXTERNSHEET 1	Name = “<01 _H >[example.xls]ExtSheet1”
XCT	Number of CRN = 2
CRN 0	Cell range = A11, contents = 11
CRN 1	Cell range = B22, contents = “ABCD”
EXTERNSHEET 2	Name = “<01 _H >[example.xls]ExtSheet3”
XCT	Number of CRN = 1
CRN 0	Cell range = C33, contents = 33
EXTERNSHEET 3	Name = “<01 _H >example.xls”
EXTERNNAME 1	Name = “GlobalName”, EXTERNSHEET = 0 (global), Formula = “=[example.xls]ExtSheet3!C33”
EXTERNNAME 2	Name = “LocalName”, EXTERNSHEET = 2, Formula = “=[example.xls]ExtSheet1!B22”
EXTERNNAME 3	Name = “LocalName”, EXTERNSHEET = 1, Formula = “=[example.xls]ExtSheet1!A11”
EXTERNSHEET 4	Name = “<01 _H >[another_example.xls]ExtSheet2”
XCT	Number of CRN = 1
CRN 0	Cell range = B2, contents = 2
EXTERNSHEET 5	Name = “<01 _H >another_example.xls”
EXTERNNAME 1	Name = “GlobalName”, EXTERNSHEET = 0 (global), Formula = “=[another_example.xls]ExtSheet2!B2”

All formulas in the example above contain the token tNameXV.

The tokens of the first three formulas refer to EXTERNSHEET 3 (the *EXTERNSHEET Document Block* of “example.xls”). Each token refers to the respective EXTERNNAME record following this EXTERNSHEET record. For instance, the third formula refers to EXTERNNAME 3 which contains the name “LocalName” and refers to EXTERNSHEET 1. EXTERNSHEET 1 is used to obtain the sheet name of the external name. The name is completed to “[example.xls]ExtSheet1!LocalName”.

The tNameXV token of the last formula refers to EXTERNSHEET 5 (the *EXTERNSHEET Document Block* of “another_example.xls”). Because EXTERNSHEET 5 describes a global external name, the name is directly completed to “another_example.xls!GlobalName”.

• **Add-In Functions**

If a worksheet uses add-in functions, a special EXTERNNAME record containing the byte sequence 01_H 3A_H will occur. It is followed by the names of all used add-in functions, each inside of an EXTERNNAME record.

Example: A document contains the formulas =ISODD(1) and =ISEVEN(1).

The *Local Link Table* of Sheet1 contains an EXTERNSHEET Document Block with the special EXTERNSHEET record for add-in functions:

EXTERNCOUNT	Number of EXTERNSHEET records = 1
EXTERNSHEET 1	Name = "<3A _H >" (add-in)
EXTERNNAME 1	Name = "ISODD"
EXTERNNAME 2	Name = "ISEVEN"

The add-in function names are represented by tNameXR tokens in the formula, here referring to EXTERNSHEET 1 and to the respective EXTERNNAME record. The token is the first argument of the EXTERN.CALL function, which invokes the call of the add-in function.

• **DDE Links, OLE Object Links**

DDE links and OLE object links expect the name of the server application (DDE) or the class name (OLE) and the name of a source document. These items are encoded in an EXTERNSHEET record. It is followed by EXTERNNAME records with additional data of the links. An EXTERNNAME record for a DDE links contains the item (data source range) and an EXTERNNAME record for an OLE object link contains the identifier of the object data storage. Inside a formula a DDE link is represented by the token tNameXV (→4.9.14). An OLE object link contains a tNameX token inside of its OBJ record.

Example: A document contains a DDE link to the range "Sheet1.A1:B2" inside of the Calc document "example.sxc" and an OLE object link to the bitmap file "example.bmp".

The *Local Link Table* of Sheet1 contains two EXTERNSHEET Document Blocks, one for the DDE link and one for the OLE object link:

EXTERNCOUNT	Number of EXTERNSHEET records = 1
EXTERNSHEET 1	Name = "soffice<03 _H >example.xls" (Server application = "soffice"; Document = "example.sxc")
EXTERNNAME 1	Type = DDE link, representing the "StdDocumentName" identifier Item = "StdDocumentName"
EXTERNNAME 2	Type = DDE link Item = "Sheet1.A1:B2"
EXTERNSHEET 2	Name = "Package<03 _H >example.bmp" (Class name = "Package"; Document = "example.bmp")
EXTERNNAME 1	Type = OLE object link Storage = 00012345 _H (storage name = "LNK00012345")

5.10.3 References in BIFF8

The main data of all types of references is stored in the *Link Table* inside the *Workbook Globals Substream* (→5.2.6). All formulas use only indexes for specific references. The *Link Table* itself is optional and occurs only, if there are any references in the document.

Common structure of the *Link Table*, BIFF8:

●●	<i>SUPBOOK Block(s)</i>	Settings for a referenced document
●	EXTERNSHEET	→6.39
○○	NAME	→6.66

Each referenced document is represented by a *SUPBOOK Block*. It starts with a *SUPBOOK* record. It contains the name of the document and the names of its sheets. It is followed by additional records that allow to dereference the data in the document. After the last *SUPBOOK Block* occurs only one *EXTERNSHEET* record. It contains a list with indexes to the *SUPBOOKs* for each used reference anywhere in the document. Formulas use indexes into this *EXTERNSHEET* list.

Example: A document contains (among other references) the two formulas

=`[example.xls]ExtSheet2!A1` and

=`[example.xls]ExtSheet1!A1`.

<i>SUPBOOK Block 0</i>	Any content
<i>SUPBOOK Block 1</i>	Document = "example.xls" Sheet 0 = "ExtSheet1" Sheet 1 = "ExtSheet2" Sheet 2 = "ExtSheet3"
<i>SUPBOOK Block 2</i>	Any content
EXTERNSHEET	REF 0 = any reference REF 1 = { <i>SUPBOOK</i> = 1, sheet range = 1...1} REF 2 = any reference REF 3 = { <i>SUPBOOK</i> = 1, sheet range = 0...0} REF 4 = any reference

The first formula uses REF 1 in the *EXTERNSHEET* record. REF 1 refers to *SUPBOOK Block 1* and sheet range 1...1. This means, the document "example.xls" is used (document of *SUPBOOK Block 1*) and the name of the sheet is "ExtSheet2" (sheet 1 of *SUPBOOK Block 1*). In the same way, the second formula uses REF 3 in the *EXTERNSHEET* record. All list entries inside of the *EXTERNSHEET* record are unique. For instance all formulas in the workbook referring to sheet "ExtSheet2" of the document "example.xls" use REF 1. All other *SUPBOOK Blocks* and *REFs* are placeholders for other references in this example.

Common structure of a *SUPBOOK Block*, BIFF8:

●	SUPBOOK	→6.100
○○	EXTERNNAME	→6.38
○○	● XCT	→6.114
	●● CRN	→6.24

The *SUPBOOK Block* may contain cached values of cells in the referenced document. These values are stored in several *XCT* and *CRN* records.

• External and 3D References

The SUPBOOK for the own document has a special format: It contains only the number of all sheets and the byte sequence 01_H 04_H instead of the sheet names. The sheet range indexes in the EXTERNSHEET record refer to the position of the sheets (zero-based). If a referenced sheet does not exist anymore, the sheet index $FFFF_H$ will occur (deleted 3D reference).

Example: A document with 7 sheets (named from “Sheet1” to “Sheet7”) contains the formulas

```
=Sheet2!A1,
=SUM(Sheet4:Sheet6!A1:B3),
=SUM([example.xls]ExtSheet1!A1:B2) (contents: A1=1.11, B1=2.22, A2=3.33, B2=4.44),
=[example.xls]ExtSheet3!A1 (contents: “ABCD”) and
=Sheet8!A1.
```

SUPBOOK 0	Number of sheets: 7 01_H 04_H (own workbook)
SUPBOOK 1	Document = “example.xls” Sheet 0 = “ExtSheet1” Sheet 1 = “ExtSheet2” Sheet 2 = “ExtSheet3”
XCT	Number of CRN = 2, sheet = 0 (ExtSheet1)
CRN 0	Cell range = A1:B1, contents = 1.11, 2.22
CRN 1	Cell range = A2:B2, contents = 3.33, 4.44
XCT	Number of CRN = 1, sheet = 2 (ExtSheet3)
CRN 0	Cell range = A1, contents = “ABCD”
EXTERNSHEET	REF 0 = {SUPBOOK = 0, sheet range = 1...1} REF 1 = {SUPBOOK = 0, sheet range = 3...5} REF 2 = {SUPBOOK = 1, sheet range = 0...0} REF 3 = {SUPBOOK = 1, sheet range = 1...1} REF 4 = {SUPBOOK = 0, sheet range = $FFFF_H$... $FFFF_H$ }

Inside of the first formula the cell reference is represented by the token tRef3d (→4.9.15). The second formula contains the token tArea3d (→4.9.16).

• Internal Names

All internal names are stored in a list of NAME records (→6.66) that follows the EXTERNSHEET record. There exist two types of internal names: global names which are valid in the whole workbook and local names which are attached to a specific sheet. For instance the local name “MyCell” of the sheet “Sheet1” can be used from everywhere in the workbook by entering =Sheet1!MyCell. Each NAME record contains the name itself and a *one-based* sheet index. The index zero indicates a global name. If the document contains local names, a special REF entry will be created in the EXTERNSHEET record. It contains the index to the internal SUPBOOK and the sheet range FFFE_H...FFFE_H.

Inside of a formula a global name or a local name of the own sheet is represented by the token tName (→4.9.1) with a *one-based* index to the NAME record list. Local names from other sheets (with explicitly denoted sheet name) are represented by the token tNameX (→4.9.14) with an index to the special REF entry of the EXTERNSHEET record and an index to the NAME record list.

Example for internal names: A document contains the global name “GlobalName”, and the local names “Sheet1!LocalName” and “Sheet2!LocalName”. In “Sheet1” there are the formulas

```
=GlobalName,  
=LocalName,  
=Sheet1!LocalName, and  
=Sheet2!LocalName.
```

SUPBOOK 0	Number of sheets: 3 01 _H 04 _H (own workbook)
EXTERNSHEET	REF 0 = {SUPBOOK = 0, sheet range = 0...0} REF 1 = {SUPBOOK = 0, sheet range = FFFE _H ...FFFE _H }
NAME 1	Name = “GlobalName”, sheet (<i>one-based</i>) = 0 (Global)
NAME 2	Name = “LocalName”, sheet (<i>one-based</i>) = 1 (Sheet1)
NAME 3	Name = “LocalName”, sheet (<i>one-based</i>) = 2 (Sheet2)

The first formula in the example above contains the token tNameV referring to NAME 1 and the second formula the same token referring to NAME 2.

The two latter formulas contain the token tNameXV with a reference to REF 1 in the EXTERNSHEET record. REF 1 refers to SUPBOOK 0 with the special sheet indexes for defined names. The token of the third formula refers to NAME 2 and the token of the last formula refers to NAME 3.

• **External Names**

In Excel, formulas can use defined names located in another workbook, called “external names”. In this case for each name an EXTERNNAME record (→6.38) occurs after the SUPBOOK record. The EXTERNNAME record contains the name itself and the *one-based* index to the sheet name of the SUPBOOK record. Again the index zero indicates a global name. If a SUPBOOK contains external names, a special REF entry will be created in the EXTERNSHEET record. It contains the index to the SUPBOOK and the sheet range FFFE_H...FFFE_H.

Inside of a formula an external name is represented by the token tNameX (→4.9.14). It contains the index to the special REF entry inside of the EXTERNSHEET record and the index to an EXTERNNAME record (*one-based*).

Example: A document contains the formulas

=example.xls!GlobalName (location: ExtSheet3!C33; contents: 33),
 =[example.xls]ExtSheet3!LocalName (location: ExtSheet1!B22; contents: “ABCD”), and
 =[example.xls]ExtSheet1!LocalName (location: ExtSheet1!A11; contents: 11).

SUPBOOK 0	Document = “example.xls” Sheet 0 = “ExtSheet1” Sheet 1 = “ExtSheet2” Sheet 2 = “ExtSheet3”
EXTERNNAME 1	Name = “GlobalName”, sheet (<i>one-based</i>) = 0 (Global), Formula = “=[example.xls]ExtSheet3!C33”
EXTERNNAME 2	Name = “LocalName”, sheet (<i>one-based</i>) = 3 (ExtSheet3), Formula = “=[example.xls]ExtSheet1!B22”
EXTERNNAME 3	Name = “LocalName”, sheet (<i>one-based</i>) = 1 (ExtSheet1), Formula = “=[example.xls]ExtSheet1!A11”
XCT	Number of CRN = 2, sheet = 0 (ExtSheet1)
CRN 0	Cell range = A11, contents = 11
CRN 1	Cell range = B22, contents = “ABCD”
XCT	Number of CRN = 1, sheet = 2 (ExtSheet3)
CRN 0	Cell range = C33, contents = 33
EXTERNSHEET	REF 0 = {SUPBOOK = 0, sheet range = FFFE _H ...FFFE _H }

All formulas in the example above contain the token tNameXV with a reference to REF 0 in the EXTERNSHEET record. REF 0 refers to SUPBOOK 0 with the special sheet indexes for defined names. Therefore the EXTERNNAME records of SUPBOOK 0 are used.

The token of the first formula refers to EXTERNNAME 1, the token of the second formula refers to EXTERNNAME 2, and the token of the last formula refers to EXTERNNAME 3.

• **Add-In Functions**

If a workbook uses add-in functions, a special SUPBOOK containing the byte sequence 01_H 3A_H will occur. It is followed by the names of all used add-in functions, each inside of an EXTERNNAME record. A special REF entry with the sheet range FFFE_H...FFFE_H will be inserted into the EXTERNSHEET reference list.

Example: A document contains the formulas =ISODD (1) and =ISEVEN (1) .

SUPBOOK 0	01 _H 3A _H (add-in)
EXTERNNAME 1	Name = “ISODD”
EXTERNNAME 2	Name = “ISEVEN”
EXTERNSHEET	REF 0 = {SUPBOOK = 0, sheet range = FFFE _H ...FFFE _H }

The add-in function names are represented by tNameXR tokens in the formula, here referring to REF 0 and to the respective EXTERNNAME record. The token is the first argument of the EXTERN.CALL function, which invokes the call of the add-in function.

• DDE Links, OLE Object Links

DDE links and OLE object links expect the name of the server application (DDE) or the class name (OLE) and the name of a source document. These items are encoded in a SUPBOOK record. The SUPBOOK is followed by EXTERNNAME records with additional data of the links. An EXTERNNAME record for a DDE links contains the item (data source range) and an EXTERNNAME record for an OLE object link contains the identifier of the object data storage. Inside a formula a DDE link is represented by the token tNameXV (→4.9.14). An OLE object link contains a tNameX token inside of its OBJ record.

Example: A document contains a DDE link to the range “Sheet1.A1:B2” inside of the Calc document “example.sxc” and an OLE object link to the bitmap file “example.bmp”.

SUPBOOK 0	Server application = “soffice” Document = “example.sxc”
EXTERNNAME 1	Type = DDE link, representing the “StdDocumentName” identifier Item = “StdDocumentName”
EXTERNNAME 2	Type = DDE link Item = “Sheet1.A1:B2”
SUPBOOK 1	Class name = “Package” Document = “example.bmp”
EXTERNNAME 1	Type = OLE object link Storage = 00012345 _H (storage name = “LNK00012345”)
EXTERNSHEET	REF 0 = {SUPBOOK = 0, sheet range = FFFE _H ...FFFE _H } REF 1 = {SUPBOOK = 1, sheet range = FFFE _H ...FFFE _H }

5.11 Shared String Table (BIFF8)

5.11.1 The SST Record

A BIFF8 workbook collects the strings of all text cells in a global list, the *Shared String Table*. This table is located in the record SST in the *Workbook Globals Substream* (→5.2.6). An SST record is followed by an EXTSSST record which stores stream positions for a string hash table.

Common structure of the *Shared String Table*, BIFF8:

• SST	→6.96
• EXTSSST	→6.40

Text cells are represented by LABELSST records (→6.61) which contain indexes into the SST record. For reading Excel files only the SST record and the LABELSST records are important, but Excel needs a valid EXTSSST record, when it reads a file.

Example: A workbook contains anywhere the strings “AAA”, “BBB” and “CCC”.

BOF	Type = workbook globals
	Workbook globals records
SST	Total number of strings in document = 4, strings in SST = 3 String 0 = “AAA” String 1 = “BBB” String 2 = “CCC”
EXTSSST	See below
	Workbook globals records
EOF	
BOF	Type = worksheet
	Cell records
LABELSST	String index = 0 (results in “AAA”)
LABELSST	String index = 2 (results in “CCC”)
	Cell records
LABELSST	String index = 1 (results in “BBB”)
LABELSST	String index = 0 (results in “AAA”)
	Cell records
EOF	

5.11.2 The EXTSSST Record

The EXTSSST record is used by Excel to create a string hash table, while loading the document. It contains stream positions of specific strings in the SST record. The SST record will be divided into several portions, which all contain the same number of strings. The first string in each portion will be referenced in the EXTSSST record.

The following example shows the absolute stream position in the first column and the relative record position (including the record headers) in the second column. The SST record is too long so that some of the strings are following in the CONTINUE record.

Abs. stream offset	Rel. rec. offset	Contents	Description
00020000 _H	0000 _H	00FC _H	SST identifier
00020002 _H	0002 _H	1000 _H	Size of the SST record
00020004 _H	0004 _H	00000011 _H	Total number of strings in the document
00020008 _H	0008 _H	00000011 _H	Number of unique strings following
0002000C _H	000C _H		String 0 (total size = 0100 _H bytes)
0002010C _H	010C _H		String 1 (total size = 0200 _H bytes)
0002030C _H	030C _H		String 2 (total size = 0100 _H bytes)
⋮	⋮		⋮
00020800 _H	0800 _H		String 8 (total size = 0100 _H bytes)
⋮	⋮		⋮
00021004 _H	0000 _H	003C _H	CONTINUE identifier
00021006 _H	0002 _H	0320 _H	Size of the CONTINUE record
00021008 _H	0004 _H		Continuation of string 14 (size = 0020 _H bytes)
00021028 _H	0024 _H		String 15 (total size = 0100 _H bytes)
00021128 _H	0124 _H		String 16 (total size = 0200 _H bytes)
00021328 _H	0000 _H	00FF _H	EXTSSST identifier
0002132A _H	0002 _H	001A _H	Size of the EXTSSST record
0002132C _H	0004 _H	0008 _H	8 strings in each portion
0002132E _H	0006 _H	0002000C _H	Absolute stream position of string 0
00021332 _H	000A _H	000C _H	Relative record position of string 0 (in SST)
00021334 _H	000C _H	0000 _H	Not used
00021336 _H	000E _H	00020800 _H	Absolute stream position of string 8
0002133A _H	0012 _H	0800 _H	Relative record position of string 8 (in SST)
0002133C _H	0014 _H	0000 _H	Not used
0002133E _H	0016 _H	00021128 _H	Absolute stream position of string 16
00021342 _H	001A _H	0124 _H	Relative record position of string 16 (in CONTINUE)
00021344 _H	001C _H	0000 _H	Not used

5.12 Conditional Formatting Table (BIFF8)

The settings of conditional formattings are stored for each sheet in the *Conditional Formatting Table*. It is contained in the *Worksheet Substream* following the cell records (→5.2.6).

Structure of the *Conditional Formatting Table*, BIFF8:

••	• CONDFMT	→6.21
	•• CF	→6.16

Each CONDFMT record in this table with its following CF records describes identical conditional formatting attributes for several cells in the worksheet.

5.13 Hyperlink Table (BIFF8)

Hyperlinks are stored for each sheet in the *Hyperlink Table*. It is contained in the *Worksheet Substream* following the cell records (→5.2.6).

Structure of the *Hyperlink Table*, BIFF8:

••	• HLINK	→6.53
	○ QUICKTIP	→6.78

Each HLINK record in this table describes a hyperlink. The optional QUICKTIP record contains a tool tip that occurs when the mouse pointer is over the hyperlink cell.

5.14 Data Validity Table (BIFF8)

Data validity settings are stored for each sheet in the *Data Validity Table*. It is contained in the *Worksheet Substream* following the cell records (→5.2.6).

Structure of the *Data Validity Table*, BIFF8:

•	DVAL	→6.34
••	DV	→6.33

A DVAL record introduces the list of DV records. The DVAL record contains the number of DV records. Each DV record contains data validity settings for and the addresses of all affected cells.

5.15 AutoFilter, Advanced Filter

2do

5.16 Scenarios

2do

5.17 Web Queries (BIFF8)

2do

5.18 Worksheet/Workbook Protection

An Excel document may contain different types of protection.

- Worksheet/workbook protection: Protects specific contents of the worksheet or workbook, for instance window settings, cell contents, or objects.
- File protection: Protects the file itself, either sets to read-only, or restricts read and write access. File protection does not care about the contents of the document. This type of protection is described in the next chapter (→5.19).

Not all records regarding protection must occur in the stream. If a record is omitted, the corresponding item is not active (for instance: omitting the OBJECTPROTECT record leaves objects unprotected). For the position of the mentioned records in the substreams see →5.2.

5.18.1 Single Worksheet Protection (BIFF2-BIFF4S)

Several records in the *Worksheet Protection Block* determine the protected items in the sheet. PROTECT protects the cell contents, WINDOWPROTECT protects the window settings, and OBJECTPROTECT protects the embedded objects. The PASSWORD record contains the hash value of the password (→5.18.4) used to protect the sheet. All the mentioned records occur always, if the file was written by Excel, but may be omitted.

If a protection password is set in the PASSWORD record, the read/write file protection (→5.19) will be enabled, which causes encryption of the file (→5.19.1) in every case. If this protection is not enabled manually while saving the file, the built-in password “VelvetSweatshop” is used, *not* the password set for worksheet protection.

Structure of the *Worksheet Protection Block*, BIFF2-BIFF4S:

○ PROTECT	Cell contents: 1 = protected (→6.77)
○ WINDOWPROTECT	Window settings: 1 = protected (→6.110)
○ OBJECTPROTECT	Embedded objects: 1 = protected (→6.69)
○ PASSWORD	Hash value of the password; 0 = No password (→6.72)

5.18.2 Sheet Protection in a Workbook (BIFF4W-BIFF8)

The PROTECT record in the *Worksheet Protection Block* indicates that the sheet is protected. There may follow a SCENPROTECT record or/and an OBJECTPROTECT record. The optional PASSWORD record contains the hash value of the password used to protect the sheet (→5.18.4). In BIFF8X there may occur additional records following the cell records in the *Worksheet Substream* (→5.2.6).

Sheet protection with password does not cause to switch on read/write file protection. Therefore the file will not be encrypted.

Structure of the *Worksheet Protection Block*, BIFF4W-BIFF8:

○ PROTECT	Worksheet contents: 1 = protected (→6.77)
○ WINDOWPROTECT	Window settings: 1 = protected (BIFF4W only, →6.110)
○ OBJECTPROTECT	Embedded objects: 1 = protected (→6.69)
○ SCENPROTECT	Scenarios: 1 = protected (BIFF5-BIFF8, →6.86)
○ PASSWORD	Hash value of the password; 0 = no password (→6.72)

5.18.3 Workbook Protection (BIFF4W-BIFF8)

The Workbook *Protection Block* in the *Workbook Globals Substream* determines the protected items in the entire workbook. The PROTECT record protects the workbook contents and the WINDOWPROTECT record protects the window settings. The PASSWORD record contains the hash value of the password used to protect the workbook (→5.18.4). All the mentioned records occur always, if the file was written by Excel, but may be omitted.

If a protection password is set in the PASSWORD record, the read/write file protection (→5.19) will be enabled, which causes encryption of the file (→5.19.1) in every case. If this protection is not enabled manually while saving the file, the built-in password “VelvetSweatshop” is used, *not* the password set for workbook protection.

Structure of the *Workbook Protection Block*:

○ WINDOWPROTECT	Window settings: 1 = protected (→6.110)
○ PROTECT	Workbook contents: 1 = protected (→6.77)
○ OBJECTPROTECT	Embedded objects: 1 = protected (BIFF4W only, →6.69)
○ PASSWORD	Hash value of the password; 0 = no password (→6.72)
○ PROT4REV	Shared workbook: 1 = protected
○ PROT4REVPASS	Hash value of the shared password; 0 = no password

5.18.4 Password Hash Value

In several records the hash value of a password is stored, used for later verification of an entered password. The length of the password is restricted to 15 characters.

The following pseudo-code algorithm shows how to create such a hash value from a given byte-string password:

```
ALGORITHM Get_Password_Hash( password )
```

- 1) hash ← 0 ; char_index ← 0 ; char_count ← character count of password
- 2) char ← character from password with index char_index {left-to-right, 0 is leftmost character}
- 3) char_index ← char_index + 1
- 4) rotate the lower 15 bits of char left by char_index bits
- 5) hash ← hash XOR char
- 6) IF char_index < char_count THEN JUMP 2)
- 7) RETURN hash XOR char_count XOR CE4B_H

Example: The password is “abcdefghij” (10 characters).

Step	char_index	char (step 2)	char (step 4)	hash (step 5)
1)				0000 _H
2) - 5)	0	61 _H ('a')	00C2 _H	00C2 _H
2) - 5)	1	62 _H ('b')	0188 _H	014A _H
2) - 5)	2	63 _H ('c')	0318 _H	0252 _H
2) - 5)	3	64 _H ('d')	0640 _H	0412 _H
2) - 5)	4	65 _H ('e')	0CA0 _H	08B2 _H
2) - 5)	5	66 _H ('f')	1980 _H	1132 _H
2) - 5)	6	67 _H ('g')	3380 _H	22B2 _H
2) - 5)	7	68 _H ('h')	6800 _H	4AB2 _H
2) - 5)	8	69 _H ('i')	5201 _H	18B3 _H
2) - 5)	9	6A _H ('j')	2803 _H	30B0 _H
7)				FEF1 _H

5.19 File Protection

A file might be protected with a password against modifying (write protection), or against opening at all (read/write protection). These passwords are set in the Save-As dialogue of Excel.

- Write protection (BIFF3-BIFF8): The WRITEPROT record marks the file to be protected against modifying. The password to unprotect the file is stored in the FILESHARING record. Write protection does not cause to encrypt the file.
- Read/write protection (BIFF2-BIFF8): A FILEPASS record occurs containing stream encryption information, which includes the encrypted password. All following records are encrypted (→5.19.1).

The records of the *File Protection Block* are the first records in a file (→5.2).

Structure of the *File Protection Block*:

○ WRITEPROT	File is write protected (BIFF3-BIFF8, →6.112), password in FILESHARING
○ FILEPASS	File is read/write-protected, encryption information (→6.41)
○ WRITEACCESS	User name (BIFF3-BIFF8, →6.111)
○ FILESHARING	File sharing options (BIFF3-BIFF8, →6.42)

5.19.1 BIFF2-BIFF7 XOR Stream Encryption

Encryption takes place for the contents of a record. The record header (record identifier and size) is not encrypted. There are a few records or data fields that are never encrypted either:

- The entire BOF record (→6.8)
- The entire INTERFACEHDR record
- The stream position field in the BOUNDSHEET record (→6.12)

Stream encryption takes place in several steps:

- 1) get the password from the user
- 2) create hash value (16-bit, →5.18.4) and encryption key (16-bit, see below) from the password
- 3) store hash value and encryption key in FILEPASS record (→6.41)
- 4) create the 128-bit sized key sequence from password and encryption key (see below)
- 5) encrypt all following records using the key sequence (see below)

Stream decryption works similar to encryption:

- 1) get the password from the user
- 2) create hash value (16-bit, →5.18.4) and encryption key (16-bit, see below) from the password
- 3) compare with values contained in FILEPASS record
- 4) if password is correct, create the 128-bit sized key sequence from password and encryption key (see below)
- 5) decrypt all following records using the key sequence (see below)

• BIFF2-BIFF7 Encryption Key

The 16-bit encryption key is used for the BIFF2-BIFF7 XOR stream encryption algorithm (→5.19.1). It is calculated from the password. The FILEPASS record stores this key and the hash value (→5.18.4) of the password (the hash value is not used to encrypt the data). The length of the password is restricted to 15 characters.

The following pseudo-code algorithm shows how to create the encryption key from a given byte-string password:

```
ALGORITHM Get_Encryption_Key_XOR( password )
1) key ← 0 ; key_base ← 8000H ; key_final ← FFFFH ; char_index ← 0
2) char_count ← character count of password
3) char ← character from password with index char_index {right-to-left, 0 is rightmost}
4) char ← char AND 7FH {use only the lower 7 bits of each character}
5) bit_index ← 0
6) rotate the lower 16 bits of key_base left by 1 bit
7) IF (least significant bit of key_base) = 1 THEN key_base ← key_base XOR 1020H
8) rotate the lower 16 bits of key_final left by 1 bit
9) IF (least significant bit of key_final) = 1 THEN key_final ← key_final XOR 1020H
10) IF (bit with index bit_index in char) = 1 THEN key ← key XOR key_base
11) bit_index ← bit_index + 1
12) IF bit_index < 8 THEN JUMP 6)
13) char_index ← char_index + 1
14) IF char_index < char_count THEN JUMP 3)
15) RETURN key XOR key_final
```

• BIFF2-BIFF7 Encryption Key Sequence

The encryption sequence is needed to encrypt or decrypt the record contents. It is generated from the password and the encryption key (which is generated from the password too, see above). The size of the encryption key sequence is always 128 bit.

The following pseudo-code algorithm shows how to create the encryption key from a given byte-string password (and the encryption key “key”):

```
ALGORITHM Get_Key_Sequence_XOR( password, key )
1) char_count ← character count of password
2) create a sequence key_seq[] containing 16 bytes
3) fill the first char_count bytes of key_seq[] with all characters of password
4) fill the remaining (16 - char_count) bytes of key_seq[] with the first bytes of the sequence
   { BBH, FFH, FFH, BAH, FFH, FFH, B9H, 80H, 00H, BEH, 0FH, 00H, BFH, 0FH, 00H }
5) key_lower ← lower 8 bits of key ; key_upper ← upper 8 bits of key
6) seq_index ← 0
7) key_seq[seq_index] ← key_seq[seq_index] XOR key_lower
8) key_seq[seq_index + 1] ← key_seq[seq_index + 1] XOR key_upper
9) seq_index ← seq_index + 2
10) IF seq_index < 16 THEN JUMP 7)
11) rotate all bytes of key_seq left by 2 bits
12) RETURN key_seq[]
```

• BIFF2-BIFF7 Stream Encryption

To encrypt record data, a 128-bit sized key sequence is used, which has been generated from a password before (see above). All stream data is encrypted in blocks of 16 bytes using the key sequence. In every data block, each byte is encoded using the corresponding byte of the key sequence. The offset pointer into the key sequence is reinitialised whenever a new record is started. The initial value of this offset is dependent on the stream position and size of the new record (it does *not* simply restart at 0). For details see the following algorithm.

The following pseudo-code algorithm shows how to encrypt the data of an entire record (assuming that stream points to the start of the record data):

```
ALGORITHM Write_Record_XOR( stream, record_data[], record_size, key_seq[] )
  1) key_index ← ((position of stream) + record_size) AND 0FH
  2) record_index ← 0
  3) IF record_index = record_size THEN RETURN
  4) data_byte = record_data[record_index] XOR key_seq[key_index]
  5) rotate all 8 bits of data_byte right by 3 bits
  6) write data_byte to stream
  7) key_index ← (key_index + 1) AND 0FH           {rotate inside key_seq[]}
  8) record_index ← record_index + 1
  9) JUMP 3)
```

• BIFF2-BIFF7 Stream Decryption

To decrypt record data, a 128-bit sized key sequence is used, which has been generated from a password before (see above). Decryption works similar to encryption (see above), but in reversed order of operations.

The following pseudo-code algorithm shows how to decrypt the data of an entire record (assuming that stream points to the start of the record data):

```
ALGORITHM Read_Record_XOR( stream, record_size, key_seq[] )
  1) key_index ← ((position of stream) + record_size) AND 0FH
  2) create a sequence record_data[] containing record_size bytes
  3) record_index ← 0
  4) IF record_index = record_size THEN RETURN record_data[]
  5) data_byte ← read 1 byte from stream
  6) rotate all 8 bits of data_byte left by 3 bits
  7) record_data[record_index] ← data_byte XOR key_seq[key_index]
  8) key_index ← (key_index + 1) AND 0FH           {rotate inside key_seq[]}
  9) record_index ← record_index + 1
  10) JUMP 4)
```


6 Worksheet/Workbook Records

The filled dot “●” means, that Excel supports the record on import and export. The unfilled dot “○” means, that Excel can read the record, but does not write it anymore or only in special cases in the respective BIFF version. An empty field denotes that the record is not supported, or that the identifier has been changed in this BIFF version.

6.1 Overview, Ordered by Record Identifier

Record ID	Record name	Occurs in BIFF versions							
		2	3	4S	4W	5	7	8	8X
0000 _H	DIMENSIONS	●							
0001 _H	BLANK	●							
0002 _H	INTEGER	●							
0003 _H	NUMBER	●							
0004 _H	LABEL	●							
0005 _H	BOOLERR	●							
0006 _H	FORMULA	●				●	●	●	●
0007 _H	STRING	●							
0008 _H	ROW	●							
0009 _H	BOF	●							
000A _H	EOF	●	●	●	●	●	●	●	●
000B _H	INDEX	●							
000C _H	CALCCOUNT	●	●	●	●	●	●	●	●
000D _H	CALCMODE	●	●	●	●	●	●	●	●
000E _H	PRECISION	●	●	●	●	●	●	●	●
000F _H	REFMODE	●	●	●	●	●	●	●	●
0010 _H	DELTA	●	●	●	●	●	●	●	●
0011 _H	ITERATION	●	●	●	●	●	●	●	●
0012 _H	PROTECT	●	●	●	●	●	●	●	●
0013 _H	PASSWORD	●	●	●	●	●	●	●	●
0014 _H	HEADER	●	●	●	●	●	●	●	●
0015 _H	FOOTER	●	●	●	●	●	●	●	●
0016 _H	EXTERNCOUNT	●	●	●	●	●			
0017 _H	EXTERNSHEET	●	●	●	●	●	●	●	●
0018 _H	NAME	●				●	●	●	●
0019 _H	WINDOWPROTECT	●	●	●	●	●	●	●	●

Record ID	Record name	Occurs in BIFF versions							
		2	3	4S	4W	5	7	8	8X
001A _H	VERTICALPAGEBREAKS	•	•	•	•	•	•	•	•
001B _H	HORIZONTALPAGEBREAKS	•	•	•	•	•	•	•	•
001C _H	NOTE	•	•	•	•	•	•	•	•
001D _H	SELECTION	•	•	•	•	•	•	•	•
001E _H	FORMAT	•	•						
001F _H	BUILTINFMTCOUNT	•							
0020 _H	COLUMNDEFAULT	•							
0021 _H	ARRAY	•							
0022 _H	DATEMODE	•	•	•	•	•	•	•	•
0023 _H	EXTERNNAME	•				•	•	•	•
0024 _H	COLWIDTH	•							
0025 _H	DEFAULTROWHEIGHT	•							
0026 _H	LEFTMARGIN	•	•	•	•	•	•	•	•
0027 _H	RIGHTMARGIN	•	•	•	•	•	•	•	•
0028 _H	TOPMARGIN	•	•	•	•	•	•	•	•
0029 _H	BOTTOMMARGIN	•	•	•	•	•	•	•	•
002A _H	PRINTHEADERS	•	•	•	•	•	•	•	•
002B _H	PRINTGRIDLINES	•	•	•	•	•	•	•	•
002F _H	FILEPASS	•	•	•	•	•	•	•	•
0031 _H	FONT	•				•	•	•	•
0036 _H	TABLEOP	•							
0037 _H	TABLEOP2	•							
003C _H	CONTINUE	•	•	•	•	•	•	•	•
003D _H	WINDOW1	•	•	•	•	•	•	•	•
003E _H	WINDOW2	•							
0040 _H	BACKUP	•	•	•	•	•	•	•	•
0041 _H	PANE	•	•	•	•	•	•	•	•
0042 _H	CODEPAGE	•	•	•	•	•	•	•	•
0043 _H	XF	•							
0044 _H	IXFE	•							
0045 _H	EFONT	•							
0051 _H	DCONREF	•	•	•	•	•	•	•	•
0055 _H	DEFCOLWIDTH	•	•	•	•	•	•	•	•
0056 _H	BUILTINFMTCOUNT		•	•	•				
0059 _H	XCT		•	•	•	•	•	•	•
005A _H	CRN		•	•	•	•	•	•	•
005B _H	FILESHARING		•	•	•	•	•	•	•
005C _H	WRITEACCESS		•	•	•	•	•	•	•
005E _H	UNCALCED		•	•	•	•	•	•	•
005F _H	SAVERECALC		•	•	•	•	•	•	•
0063 _H	OBJECTPROTECT		•	•	•	•	•	•	•
007D _H	COLINFO		•	•	•	•	•	•	•
0080 _H	GUTS		•	•	•	•	•	•	•
0081 _H	WSBOOL		•	•	•	•	•	•	•
0082 _H	GRIDSET		•	•	•	•	•	•	•
0083 _H	HCENTER		•	•	•	•	•	•	•

Record ID	Record name	Occurs in BIFF versions							
		2	3	4S	4W	5	7	8	8X
0084 _H	VCENTER		•	•	•	•	•	•	•
0085 _H	BOUNDSHEET				•	•	•	•	•
0086 _H	WRITEPROT		•	•	•	•	•	•	•
008C _H	COUNTRY		•	•	•	•	•	•	•
008D _H	HIDEOBJ		•	•	•	•	•	•	•
008E _H	SHEETSOFFSET				•				
008F _H	SHEETHDR				•				
0090 _H	SORT					•	•	•	•
0092 _H	PALETTE		•	•	•	•	•	•	•
0099 _H	STANDARDWIDTH			•	•	•	•	•	•
00A0 _H	SCL			•	•	•	•	•	•
00A1 _H	SETUP			•	•	•	•	•	•
00AB _H	GCW			•	•	•	•		
00BD _H	MULRK					•	•	•	•
00BE _H	MULBLANK					•	•	•	•
00D6 _H	RSTRING					•	•	○	○
00D7 _H	DBCELL					•	•	•	•
00DA _H	BOOKBOOL					•	•	•	•
00DD _H	SCENPROTECT					•	•	•	•
00E0 _H	XF					•	•	•	•
00E5 _H	MERGEDCELLS							•	•
00E9 _H	BITMAP							•	•
00EF _H	PHONETIC							•	•
00FC _H	SST							•	•
00FD _H	LABELSST							•	•
00FF _H	EXTSST							•	•
015F _H	LABELRANGES							•	•
0160 _H	USESELFS							•	•
0161 _H	DSF							•	•
01AE _H	SUPBOOK							•	•
01B0 _H	CONDFMT							•	•
01B2 _H	DVAL							•	•
01B8 _H	HLINK							•	•
01BE _H	DV							•	•
0200 _H	DIMENSIONS		•	•	•	•	•	•	•
0201 _H	BLANK		•	•	•	•	•	•	•
0203 _H	NUMBER		•	•	•	•	•	•	•
0204 _H	LABEL		•	•	•	•	•	○	○
0205 _H	BOOLERR		•	•	•	•	•	•	•
0206 _H	FORMULA		•						
0207 _H	STRING		•	•	•	•	•	•	•
0208 _H	ROW		•	•	•	•	•	•	•
0209 _H	BOF		•						
020B _H	INDEX		•	•	•	•	•	•	•
0218 _H	NAME		•	•	•				
0221 _H	ARRAY		•	•	•	•	•	•	•

Record ID	Record name	Occurs in BIFF versions							
		2	3	4S	4W	5	7	8	8X
0223 _H	EXTERNNAME		•	•	•				
0225 _H	DEFAULTROWHEIGHT		•	•	•	•	•	•	•
0231 _H	FONT		•	•	•				
0236 _H	TABLEOP		•	•	•	•	•	•	•
023E _H	WINDOW2		•	•	•	•	•	•	•
0243 _H	XF		•						
027E _H	RK		•	•	•	•	•	•	•
0293 _H	STYLE		•	•	•	•	•	•	•
0406 _H	FORMULA			•	•				
0409 _H	BOF			•	•				
041E _H	FORMAT			•	•	•	•	•	•
0443 _H	XF			•	•				
04BC _H	SHRFMLA					•	•	•	•
0800 _H	QUICKTIP							•	•
0809 _H	BOF					•	•	•	•
0862 _H	SHEETLAYOUT								•
0867 _H	SHEETPROTECTION								•
0868 _H	RANGEPROTECTION								•

6.2 Overview, Ordered by Record Names

Record ID	Record name	Occurs in BIFF versions							
		2	3	4S	4W	5	7	8	8X
0021 _H 0221 _H	ARRAY	•	•	•	•	•	•	•	•
0040 _H	BACKUP	•	•	•	•	•	•	•	•
00E9 _H	BITMAP							•	•
0001 _H 0201 _H	BLANK	•	•	•	•	•	•	•	•
0*09 _H	BOF	•	•	•	•	•	•	•	•
00DA _H	BOOKBOOL					•	•	•	•
0005 _H 0205 _H	BOOLERR	•	•	•	•	•	•	•	•
0029 _H	BOTTOMMARGIN	•	•	•	•	•	•	•	•
0085 _H	BOUNDSHEET				•	•	•	•	•
001F _H 0056 _H	BUILTINFMTCOUNT	•	•	•	•				
000C _H	CALCCOUNT	•	•	•	•	•	•	•	•
000D _H	CALCMODE	•	•	•	•	•	•	•	•
0042 _H	CODEPAGE	•	•	•	•	•	•	•	•
007D _H	COLINFO		•	•	•	•	•	•	•
0020 _H	COLUMNDEFAULT	•							
0024 _H	COLWIDTH	•							
01B0 _H	CONDFMT							•	•
003C _H	CONTINUE	•	•	•	•	•	•	•	•
008C _H	COUNTRY		•	•	•	•	•	•	•
005A _H	CRN		•	•	•	•	•	•	•
0022 _H	DATEMODE	•	•	•	•	•	•	•	•
00D7 _H	DBCELL					•	•	•	•
0051 _H	DCONREF	•	•	•	•	•	•	•	•
0025 _H 0225 _H	DEFAULTROWHEIGHT	•	•	•	•	•	•	•	•
0055 _H	DEFCOLWIDTH	•	•	•	•	•	•	•	•
0010 _H	DELTA	•	•	•	•	•	•	•	•
0000 _H 0200 _H	DIMENSIONS	•	•	•	•	•	•	•	•
0161 _H	DSF							•	•
01BE _H	DV							•	•
01B2 _H	DVAL							•	•
0045 _H	EFONT	•							
000A _H	EOF	•	•	•	•	•	•	•	•
0016 _H	EXTERNCOUNT	•	•	•	•	•	•		
0023 _H 0223 _H	EXTERNNAME	•	•	•	•	•	•	•	•
0017 _H	EXTERNSHEET	•	•	•	•	•	•	•	•
00FF _H	EXTSST							•	•
002F _H	FILEPASS	•	•	•	•	•	•	•	•
005B _H	FILESHARING		•	•	•	•	•	•	•
0031 _H 0231 _H	FONT	•	•	•	•	•	•	•	•
0015 _H	FOOTER	•	•	•	•	•	•	•	•
001E _H 041E _H	FORMAT	•	•	•	•	•	•	•	•
0*06 _H	FORMULA	•	•	•	•	•	•	•	•
00AB _H	GCW			•	•	•	•		

Record ID	Record name	Occurs in BIFF versions							
		2	3	4S	4W	5	7	8	8X
0082 _H	GRIDSET		•	•	•	•	•	•	•
0080 _H	GUTS		•	•	•	•	•	•	•
0083 _H	HCENTER		•	•	•	•	•	•	•
0014 _H	HEADER	•	•	•	•	•	•	•	•
008D _H	HIDEOBJ		•	•	•	•	•	•	•
01B8 _H	HLINK							•	•
001B _H	HORIZONTALPAGEBREAKS	•	•	•	•	•	•	•	•
000B _H 020B _H	INDEX	•	•	•	•	•	•	•	•
0002 _H	INTEGER	•							
0011 _H	ITERATION	•	•	•	•	•	•	•	•
0044 _H	IXFE	•							
0004 _H 0204 _H	LABEL	•	•	•	•	•	•	○	○
015F _H	LABELRANGES							•	•
00FD _H	LABELSST							•	•
0026 _H	LEFTMARGIN	•	•	•	•	•	•	•	•
00E5 _H	MERGEDCELLS							•	•
00BE _H	MULBLANK					•	•	•	•
00BD _H	MULRK					•	•	•	•
0018 _H 0218 _H	NAME	•	•	•	•	•	•	•	•
001C _H	NOTE	•	•	•	•	•	•	•	•
0003 _H 0203 _H	NUMBER	•	•	•	•	•	•	•	•
0063 _H	OBJECTPROTECT		•	•	•	•	•	•	•
0092 _H	PALETTE	•	•	•	•	•	•	•	•
0041 _H	PANE	•	•	•	•	•	•	•	•
0013 _H	PASSWORD	•	•	•	•	•	•	•	•
00EF _H	PHONETIC							•	•
000E _H	PRECISION	•	•	•	•	•	•	•	•
002B _H	PRINTGRIDLINES	•	•	•	•	•	•	•	•
002A _H	PRINTHEADERS	•	•	•	•	•	•	•	•
0012 _H	PROTECT	•	•	•	•	•	•	•	•
0800 _H	QUICKTIP							•	•
0868 _H	RANGEPROTECTION								•
000F _H	REFMODE	•	•	•	•	•	•	•	•
0027 _H	RIGHTMARGIN	•	•	•	•	•	•	•	•
027E _H	RK		•	•	•	•	•	•	•
0008 _H 0208 _H	ROW	•	•	•	•	•	•	•	•
00D6 _H	RSTRING					•	•	○	○
005F _H	SAVERECALC		•	•	•	•	•	•	•
00DD _H	SCENPROTECT					•	•	•	•
00A0 _H	SCL			•	•	•	•	•	•
001D _H	SELECTION	•	•	•	•	•	•	•	•
00A1 _H	SETUP			•	•	•	•	•	•
008F _H	SHEETHDR				•				
0862 _H	SHEETLAYOUT								•
0867 _H	SHEETPROTECTION								•
008E _H	SHEETSOFFSET				•				

Record ID	Record name	Occurs in BIFF versions							
		2	3	4S	4W	5	7	8	8X
04BC _H	SHRFMLA					•	•	•	•
0090 _H	SORT					•	•	•	•
00FC _H	SST							•	•
0099 _H	STANDARDWIDTH			•	•	•	•	•	•
0007 _H 0207 _H	STRING	•	•	•	•	•	•	•	•
0293 _H	STYLE		•	•	•	•	•	•	•
01AE _H	SUPBOOK							•	•
0036 _H 0236 _H	TABLEOP	•	•	•	•	•	•	•	•
0037 _H	TABLEOP2	•							
0028 _H	TOPMARGIN	•	•	•	•	•	•	•	•
005E _H	UNCALCED		•	•	•	•	•	•	•
0160 _H	USESELFS							•	•
0084 _H	VCENTER		•	•	•	•	•	•	•
001A _H	VERTICALPAGEBREAKS	•	•	•	•	•	•	•	•
003D _H	WINDOW1	•	•	•	•	•	•	•	•
003E _H 023E _H	WINDOW2	•	•	•	•	•	•	•	•
0019 _H	WINDOWPROTECT	•	•	•	•	•	•	•	•
005C _H	WRITEACCESS		•	•	•	•	•	•	•
0086 _H	WRITEPROT		•	•	•	•	•	•	•
0081 _H	WSBOOL		•	•	•	•	•	•	•
0059 _H	XCT		•	•	•	•	•	•	•
0*43 _H 00E0 _H	XF	•	•	•	•	•	•	•	•

6.3 Overview, Ordered by BIFF Versions

6.3.1 New Records

• All Records in BIFF2

Record ID	Record name	Occurs in BIFF versions							
		2	3	4S	4W	5	7	8	8X
0000 _H 0200 _H	DIMENSIONS	•	•	•	•	•	•	•	•
0001 _H 0201 _H	BLANK	•	•	•	•	•	•	•	•
0002 _H	INTEGER	•							
0003 _H 0203 _H	NUMBER	•	•	•	•	•	•	•	•
0004 _H 0204 _H	LABEL	•	•	•	•	•	○	○	
0005 _H 0205 _H	BOOLERR	•	•	•	•	•	•	•	•
0*06 _H	FORMULA	•	•	•	•	•	•	•	•
0007 _H 0207 _H	STRING	•	•	•	•	•	•	•	•
0008 _H 0208 _H	ROW	•	•	•	•	•	•	•	•
0*09 _H	BOF	•	•	•	•	•	•	•	•
000A _H	EOF	•	•	•	•	•	•	•	•
000B _H 020B _H	INDEX	•	•	•	•	•	•	•	•
000C _H	CALCCOUNT	•	•	•	•	•	•	•	•
000D _H	CALCMODE	•	•	•	•	•	•	•	•
000E _H	PRECISION	•	•	•	•	•	•	•	•
000F _H	REFMODE	•	•	•	•	•	•	•	•
0010 _H	DELTA	•	•	•	•	•	•	•	•
0011 _H	ITERATION	•	•	•	•	•	•	•	•
0012 _H	PROTECT	•	•	•	•	•	•	•	•
0013 _H	PASSWORD	•	•	•	•	•	•	•	•
0014 _H	HEADER	•	•	•	•	•	•	•	•
0015 _H	FOOTER	•	•	•	•	•	•	•	•
0016 _H	EXTERNCOUNT	•	•	•	•	•			
0017 _H	EXTERNSHEET	•	•	•	•	•	•	•	•
0018 _H 0218 _H	NAME	•	•	•	•	•	•	•	•
0019 _H	WINDOWPROTECT	•	•	•	•	•	•	•	•
001A _H	VERTICALPAGEBREAKS	•	•	•	•	•	•	•	•
001B _H	HORIZONTALPAGEBREAKS	•	•	•	•	•	•	•	•
001C _H	NOTE	•	•	•	•	•	•	•	•
001D _H	SELECTION	•	•	•	•	•	•	•	•
001E _H 041E _H	FORMAT	•	•	•	•	•	•	•	•
001E _H 0056 _H	BUILTINFMTCOUNT	•	•	•	•				
0020 _H	COLUMNDEFAULT	•							
0021 _H 0221 _H	ARRAY	•	•	•	•	•	•	•	•
0022 _H	DATEMODE	•	•	•	•	•	•	•	•
0023 _H 0223 _H	EXTERNNAME	•	•	•	•	•	•	•	•
0024 _H	COLWIDTH	•							
0025 _H 0225 _H	DEFAULTROWHEIGHT	•	•	•	•	•	•	•	•

Record ID	Record name	Occurs in BIFF versions							
		2	3	4S	4W	5	7	8	8X
0026 _H	LEFTMARGIN	•	•	•	•	•	•	•	•
0027 _H	RIGHTMARGIN	•	•	•	•	•	•	•	•
0028 _H	TOPMARGIN	•	•	•	•	•	•	•	•
0029 _H	BOTTOMMARGIN	•	•	•	•	•	•	•	•
002A _H	PRINTHEADERS	•	•	•	•	•	•	•	•
002B _H	PRINTGRIDLINES	•	•	•	•	•	•	•	•
002F _H	FILEPASS	•	•	•	•	•	•	•	•
0031 _H 0231 _H	FONT	•	•	•	•	•	•	•	•
0036 _H 0236 _H	TABLEOP	•	•	•	•	•	•	•	•
0037 _H	TABLEOP2	•							
003C _H	CONTINUE	•	•	•	•	•	•	•	•
003D _H	WINDOW1	•	•	•	•	•	•	•	•
003E _H 023E _H	WINDOW2	•	•	•	•	•	•	•	•
0040 _H	BACKUP	•	•	•	•	•	•	•	•
0041 _H	PANE	•	•	•	•	•	•	•	•
0042 _H	CODEPAGE	•	•	•	•	•	•	•	•
0*43 _H 00E0 _H	XF	•	•	•	•	•	•	•	•
0044 _H	IXFE	•							
0045 _H	EFONT	•							
0051 _H	DCONREF	•	•	•	•	•	•	•	•
0055 _H	DEFCOLWIDTH	•	•	•	•	•	•	•	•

• **New Records in BIFF3**

Record ID	Record name	Occurs in BIFF versions						
		3	4S	4W	5	7	8	8X
0059 _H	XCT	•	•	•	•	•	•	•
005A _H	CRN	•	•	•	•	•	•	•
005B _H	FILESHARING	•	•	•	•	•	•	•
005C _H	WRITEACCESS	•	•	•	•	•	•	•
005E _H	UNCALCED	•	•	•	•	•	•	•
005F _H	SAVERECALC	•	•	•	•	•	•	•
0063 _H	OBJECTPROTECT	•	•	•	•	•	•	•
007D _H	COLINFO	•	•	•	•	•	•	•
027E _H	RK	•	•	•	•	•	•	•
0080 _H	GUTS	•	•	•	•	•	•	•
0081 _H	WSBOOL	•	•	•	•	•	•	•
0082 _H	GRIDSET	•	•	•	•	•	•	•
0083 _H	HCENTER	•	•	•	•	•	•	•
0084 _H	VCENTER	•	•	•	•	•	•	•
0086 _H	WRITEPROT	•	•	•	•	•	•	•
008C _H	COUNTRY	•	•	•	•	•	•	•
008D _H	HIDEOBJ	•	•	•	•	•	•	•
0092 _H	PALETTE	•	•	•	•	•	•	•
0293 _H	STYLE	•	•	•	•	•	•	•

• **New Records in BIFF4**

Record ID	Record name	Occurs in BIFF versions					
		4S	4W	5	7	8	8X
0085 _H	BOUNDSHEET		•	•	•	•	•
008E _H	SHEETSOFFSET		•				
008F _H	SHEETHDR		•				
0099 _H	STANDARDWIDTH	•	•	•	•	•	•
00A0 _H	SCL	•	•	•	•	•	•
00A1 _H	SETUP	•	•	•	•	•	•
00AB _H	GCW	•	•	•	•		

• **New Records in BIFF5/BIFF7**

Record ID	Record name	Occurs in BIFF versions			
		5	7	8	8X
0090 _H	SORT	•	•	•	•
00BD _H	MULRK	•	•	•	•
00BE _H	MULBLANK	•	•	•	•
00D6 _H	RSTRING	•	•	○	○
00D7 _H	DBCELL	•	•	•	•
00DA _H	BOOKBOOL	•	•	•	•
00DD _H	SCENPROTECT	•	•	•	•
04BC _H	SHRFMLA	•	•	•	•

• **New Records in BIFF8/BIFF8X**

Record ID	Record name	Occurs in BIFF versions	
		8	8X
00E5 _H	MERGEDCELLS	•	•
00E9 _H	BITMAP	•	•
00EF _H	PHONETIC	•	•
00FC _H	SST	•	•
00FD _H	LABELSST	•	•
00FF _H	EXTSST	•	•
015F _H	LABELRANGES	•	•
0160 _H	USESELFS	•	•
0161 _H	DSF	•	•
01AE _H	SUPBOOK	•	•
01B0 _H	CONDFMT	•	•
01B2 _H	DVAL	•	•
01B8 _H	HLINK	•	•
01BE _H	DV	•	•
0800 _H	QUICKTIP	•	•
0862 _H	SHEETLAYOUT		•
0867 _H	SHEETPROTECTION		•
0868 _H	RANGEPROTECTION		•

6.3.2 Deleted Records

• Records Deleted in BIFF3

Record ID	Record name	Occurs in BIFF versions							
		2	3	4S	4W	5	7	8	8X
0002 _H	INTEGER	•							
0020 _H	COLUMNDEFAULT	•							
0024 _H	COLWIDTH	•							
0037 _H	TABLEOP2	•							
0044 _H	IXFE	•							
0045 _H	EFONT	•							

• Records Deleted in BIFF5/BIFF7

Record ID	Record name	Occurs in BIFF versions							
		2	3	4S	4W	5	7	8	8X
001F _H 0056 _H	BULTINFMTCOUNT	•	•	•	•				
008E _H	SHEETSOFFSET				•				
008F _H	SHEETHDR				•				

• Records Deleted in BIFF8/BIFF8X

Record ID	Record name	Occurs in BIFF versions							
		2	3	4S	4W	5	7	8	8X
0004 _H 0204 _H	LABEL	•	•	•	•	•	•	○	○
0016 _H	EXTERNCOUNT	•	•	•	•	•	•		
00AB _H	GCW			•	•	•	•		
00D6 _H	RSTRING					•	•	○	○

6.4 ARRAY

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0021 _H	0221 _H	0221 _H	0221 _H	0221 _H	0221 _H	0221 _H	0221 _H

This record stores the token array of an array formula. It is not a real cell record, but follows the first FORMULA record (→6.46) of the array cell range. For more information about array formulas see →5.8.

Record ARRAY, BIFF2:

Offset	Size	Contents
0	6	The cell range address of the array formula (→3.13.1)
6	1	0 = Do not recalculate the array formula, 1 = Always recalculate array formula
7	var.	Token array of the array formula (→4)

Record ARRAY, BIFF3-BIFF4:

Offset	Size	Contents									
0	6	The cell range address of the array formula (→3.13.1)									
6	2	Option flags: <table border="1" data-bbox="416 808 1402 920"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0001_H</td> <td>1 = Always recalculate array formula</td> </tr> <tr> <td>1</td> <td>0002_H</td> <td>1 = Calculate array formula on open</td> </tr> </tbody> </table>	Bit	Mask	Contents	0	0001 _H	1 = Always recalculate array formula	1	0002 _H	1 = Calculate array formula on open
Bit	Mask	Contents									
0	0001 _H	1 = Always recalculate array formula									
1	0002 _H	1 = Calculate array formula on open									
8	var.	Token array of the array formula (→4)									

Record ARRAY, BIFF5-BIFF8:

Offset	Size	Contents									
0	6	The cell range address of the array formula (→3.13.1). Column indexes are always 8-bit values, also in BIFF8.									
6	2	Option flags: <table border="1" data-bbox="416 1155 1402 1267"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0001_H</td> <td>1 = Always recalculate array formula</td> </tr> <tr> <td>1</td> <td>0002_H</td> <td>1 = Calculate array formula on open</td> </tr> </tbody> </table>	Bit	Mask	Contents	0	0001 _H	1 = Always recalculate array formula	1	0002 _H	1 = Calculate array formula on open
Bit	Mask	Contents									
0	0001 _H	1 = Always recalculate array formula									
1	0002 _H	1 = Calculate array formula on open									
8	4	Not used									
12	var.	Token array of the array formula (→4)									

6.5 BACKUP

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0040 _H	0040 _H	0040 _H	0040 _H	0040 _H	0040 _H	0040 _H	0040 _H

This record contains a Boolean value determining whether Excel makes a backup of the file while saving.

Record BACKUP, BIFF2-BIFF8:

Offset	Size	Contents
0	2	1 = Create a backup on saving

6.6 BITMAP

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	—	—	00E9 _H	00E9 _H

This record is part of the *Page Settings Block* (→5.4). It stores the background bitmap of a worksheet.

Record BITMAP, BIFF8:

Offset	Size	Contents
0	2	Unknown value 0009 _H
2	2	Unknown value 0001 _H
4	4	Total size of the following record data, without this field (including CONTINUE records)
8	2	Unknown value 000C _H
10	2	Unknown value 0000 _H
12	2	Width of the picture (<i>width</i>), in pixel
14	2	Height of the picture (<i>height</i>), in pixel
16	2	Number of planes, must be 0001 _H
18	2	Colour depth, must be 0018 _H (24 bit true-colour)
20	var.	Pixel data (array of <i>height</i> lines of the bitmap, from bottom line to top line, see below)

In each line all pixels are written from left to right. Each pixel is stored as 3-byte array: the red, green, and blue component of the colour of the pixel, in this order. The size of each line is aligned to multiples of 4 by inserting zero bytes after the last pixel.

Example of the bitmap data for a 3 × 3 image, each entry represents one byte. The three pixels of one line are stored in 9 bytes, therefore each line is expanded to 12 bytes (next multiple of 4).

Offset	Contents	Remarks
20	R G B R G B R G B 0 0 0	Third (bottom) line of the bitmap
32	R G B R G B R G B 0 0 0	Second line of the bitmap
44	R G B R G B R G B 0 0 0	First (top) line of the bitmap

6.7 BLANK

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0001 _H	0201 _H	0201 _H	0201 _H	0201 _H	0201 _H	0201 _H	0201 _H

This record represents an empty cell. It contains the cell address and formatting information.

Record BLANK, BIFF2:

Offset	Size	Contents
0	2	Index to row
2	2	Index to column
4	3	Cell attributes (→3.12)

Record BLANK, BIFF3-BIFF8:

Offset	Size	Contents
0	2	Index to row
2	2	Index to column
4	2	Index to XF record (→6.115)

6.8 BOF – Beginning of File

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0009 _H	0209 _H	0409 _H	0409 _H	0809 _H	0809 _H	0809 _H	0809 _H

The BOF record is the first record of a *Worksheet Substream*, the *Workbook Globals Substream*, a *Chart Substream*, or a macro sheet. A BOF record will never be encrypted, regardless of its type and position in the stream.

If a BIFF8 version of Excel (Excel 97 and newer) writes a BIFF5/BIFF7 workbook, it writes a wrong BIFF version in worksheet BOF records. Only the leading workbook globals BOF record contains the correct value and should be used to determine the BIFF version for the entire stream.

6.8.1 BOF Records Written by Excel

Record BOF, BIFF2 (record identifier is 0009_H):

Offset	Size	Contents
0	2	BIFF version (not used)
2	2	Type of the following data: <ul style="list-style-type: none"> 0010_H = Worksheet 0020_H = Chart 0040_H = Macro sheet

Record BOF, BIFF3 (record identifier is 0209_H) and BIFF4 (record identifier is 0409_H):

Offset	Size	Contents
0	2	BIFF version (not used)
2	2	Type of the following data: <ul style="list-style-type: none"> 0010_H = Worksheet 0020_H = Chart 0040_H = Macro sheet 0100_H = Workbook globals (BIFF4W only)
4	2	Not used

Record BOF, BIFF5/BIFF7 (record identifier is 0809_H):

Offset	Size	Contents
0	2	BIFF version (always 0500 _H for BIFF5/BIFF7). Should only be used, if this record is the leading workbook globals BOF (see above).
2	2	Type of the following data: <ul style="list-style-type: none"> 0005_H = Workbook globals 0006_H = Visual Basic module 0010_H = Worksheet 0020_H = Chart 0040_H = Macro sheet 0100_H = Workspace file
4	2	Build identifier
6	2	Build year

Record BOF, BIFF8/BIFF8X (record identifier is 0809_h):

Offset	Size	Contents
0	2	BIFF version (always 0600 _h for BIFF8)
2	2	Type of the following data: <ul style="list-style-type: none"> 0005_h = Workbook globals 0006_h = Visual Basic module 0010_h = Worksheet 0020_h = Chart 0040_h = Macro sheet 0100_h = Workspace file
4	2	Build identifier
6	2	Build year
8	4	File history flags
12	4	Lowest Excel version that can read all records in this file

6.8.2 BOF Records Written by Several External Tools

Various external tools write non-standard BOF records with the record identifier 0809_h (determining a BIFF5-BIFF8 BOF record), but with a different BIFF version field. In this case, the record identifier is ignored, and only the version field is used to set the BIFF version of the workbook.

Record BOF (record identifier is 0809_h):

Offset	Size	Contents
0	2	BIFF version: <ul style="list-style-type: none"> 0000_h = BIFF2 0007_h = BIFF2 0200_h = BIFF2 0300_h = BIFF3 0400_h = BIFF4 0500_h = BIFF5/BIFF7 0600_h = BIFF8
2	2	Type of the following data: <ul style="list-style-type: none"> 0005_h = Workbook globals 0006_h = Visual Basic module 0010_h = Worksheet 0020_h = Chart 0040_h = Macro sheet 0100_h = Workspace file
[4]	var.	(optional) Additional fields of a BOF record, should be ignored

6.9 BOOKBOOL

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	00DA _H	00DA _H	00DA _H	00DA _H

This record contains a Boolean value determining whether to save values linked from external workbooks (CRN records, →6.24 and XCT records, →6.114). In BIFF3 and BIFF4 this option is stored in the WSBOOL record (→6.113). See →5.10 for details about external references.

Record BOOKBOOL, BIFF5-BIFF8:

Offset	Size	Contents
0	2	0 = Save external linked values; 1 = <i>Do not</i> save external linked values

6.10 BOOLERR

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0005 _H	0205 _H	0205 _H	0205 _H	0205 _H	0205 _H	0205 _H	0205 _H

This record represents a Boolean value or error value cell.

Record BOOLERR, BIFF2:

Offset	Size	Contents
0	2	Index to row
2	2	Index to column
4	3	Cell attributes (→3.12)
7	1	Boolean or error value (type depends on the following byte)
8	1	0 = Boolean value; 1 = Error code

Record BOOLERR, BIFF3-BIFF8:

Offset	Size	Contents
0	2	Index to row
2	2	Index to column
4	2	Index to XF record (→6.115)
6	1	Boolean or error value (type depends on the following byte)
7	1	0 = Boolean value; 1 = Error code

If the value field is a Boolean value, it will contain 0 for FALSE and 1 for TRUE. See →3.7 for a list of error codes.

6.11 BOTTOMMARGIN

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0029 _H	0029 _H	0029 _H	0029 _H	0029 _H	0029 _H	0029 _H	0029 _H

This record is part of the *Page Settings Block* (→5.4). It contains the bottom page margin of the current worksheet.

Record BOTTOMMARGIN, BIFF2-BIFF8:

Offset	Size	Contents
0	8	Bottom page margin in inches (IEEE 754 floating-point value, 64-bit double precision)

6.12 BOUNDSHEET

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	0085 _H	0085 _H	0085 _H	0085 _H	0085 _H

This record is located in the workbook globals area and represents a sheet inside of the workbook. For each sheet a BOUNDSHEET record is written. It stores the sheet name and a stream offset to the BOF record (→6.8) within the workbook stream. The record is also known as BUNDLESHEET.

Record BOUNDSHEET, BIFF4W:

2do

Record BOUNDSHEET, BIFF5-BIFF8:

Offset	Size	Contents
0	4	Absolute stream position of the BOF record of the sheet represented by this record. This field is never encrypted in protected files.
4	1	Visibility: <ul style="list-style-type: none"> 00_H = Visible 01_H = Hidden 02_H = Strong hidden (see below)
5	1	Sheet type: <ul style="list-style-type: none"> 00_H = Worksheet 02_H = Chart 06_H = Visual Basic module
6	var.	Sheet name: <ul style="list-style-type: none"> BIFF5/BIFF7: Byte string, 8-bit string length (→3.3) BIFF8: Unicode string, 8-bit string length (→3.4)

The strong hidden flag can only be set and cleared with a Visual Basic macro. It is not possible to make such a sheet visible via the user interface.

6.13 BUILTINFMTCOUNT

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
001F _H	0056 _H	0056 _H	0056 _H	—	—	—	—

This record contains the number of following FORMAT records (→6.45) that contain built-in number formats. All additional FORMAT records contain user-defined number formats. Note that the record identifier changes in BIFF3.

Record BUILTINFMTCOUNT, BIFF2-BIFF4:

Offset	Size	Contents
0	2	Number of following FORMAT records containing built-in number formats

6.14 CALCCOUNT

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
000C _H	000C _H	000C _H	000C _H	000C _H	000C _H	000C _H	000C _H

This record is part of the *Calculation Settings Block* (→5.3). It specifies the maximum number of times the formulas should be iteratively calculated. This is a fail-safe against mutually recursive formulas locking up a spreadsheet application.

Record CALCCOUNT, BIFF2-BIFF8:

Offset	Size	Contents
0	2	Maximum number of iterations allowed in circular references

6.15 CALCMODE

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
000D _H	000D _H	000D _H	000D _H	000D _H	000D _H	000D _H	000D _H

This record is part of the *Calculation Settings Block* (→5.3). It specifies whether to calculate formulas manually, automatically or automatically except for multiple table operations.

Record CALCMODE, BIFF2-BIFF8:

Offset	Size	Contents
0	2	FFFF _H = automatic except for multiple table operations 0000 _H = manually 0001 _H = automatically (default)

6.16 CF – Conditional Formatting

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	—	—	01B1 _H	01B1 _H

This record is part of the *Conditional Formatting Table* (→5.12). A list of CF records always follows a CONDFMT record (→6.21). Each CF record contains a condition and the formatting attributes applied to the cells specified in the CONDFMT record, if the condition is met.

There are some restrictions in the usage of conditional formattings:

- In the user interface it is possible to modify the font style (boldness and posture), the text colour, the underline style, and the strikethrough style. It is not possible to change the used font, the font height, and the escapement style, though it is possible to specify a new font height and escapement style in this record which are correctly displayed.
- It is not possible to change a border line style, but to preserve the line colour, and vice versa. Diagonal lines are not supported at all. The user interface only offers thin line styles, but files containing other line styles work correctly too.
- It is not possible to set the background pattern colour to “No colour” (using system window background colour) and to preserve the pattern style of the cell. “No colour” will always set the pattern to transparent.

Record CF, BIFF8:

Offset	Size	Contents
0	1	Type of the conditional formatting: 01 _H = Compare with current cell value (the comparison specified below is used) 02 _H = Evaluate a formula (condition is met if formula evaluates to a value not equal to 0)
1	1	Comparison operator: 00 _H = No comparison (only valid for formula type, see above) 01 _H = Between 02 _H = Not between 03 _H = Equal 04 _H = Not equal 05 _H = Greater than 06 _H = Less than 07 _H = Greater or equal 08 _H = Less or equal
2	2	Size of the formula data for first value or formula (<u>sz1</u>)
4	2	Size of the formula data for second value or formula (<u>sz2</u> , used for second part of “Between” and “Not between” comparison, this field is 0 for other comparisons)
6	4	Option flags (see below)
10	2	Not used
[12]	118	(optional, only if <u>font</u> = 1, see option flags) Font formatting block, see below
[var.]	8	(optional, only if <u>bord</u> = 1, see option flags) Border formatting block, see below
[var.]	4	(optional, only if <u> patt</u> = 1, see option flags) Pattern formatting block, see below
var.	<u>sz1</u>	Formula data for first value or formula (RPN token array without size field, →4)
var.	<u>sz2</u>	Formula data for second value or formula (RPN token array without size field, →4)

• Option Flags

If none of the formatting attributes is set, the option flags field contains 00000000_H. The following table assumes that the conditional formatting contains at least one modified formatting attribute (it will occur at least one of the formatting information blocks in the record). In difference to the first case some of the bits are always set now.

- ! All flags specifying that an attribute is modified are 0₂, if the conditional formatting changes the respective attribute, and 1₂, if the original cell formatting is preserved. The flags for modified font attributes are not contained in this option flags field, but in the font formatting block itself.

Bit	Mask	Contents
9-0	000003FF _H	Always 11 . 1111 . 1111 ₂ (but not used)
10	00000400 _H	0 = Left border style and colour modified (<u>bord-left</u>)
11	00000800 _H	0 = Right border style and colour modified (<u>bord-right</u>)
12	00001000 _H	0 = Top border style and colour modified (<u>bord-top</u>)
13	00002000 _H	0 = Bottom border style and colour modified (<u>bord-bot</u>)
15-14	0000C000 _H	Always 11 ₂ (but not used)
16	00010000 _H	0 = Pattern style modified (<u>patt-style</u>)
17	00020000 _H	0 = Pattern colour modified (<u>patt-col</u>)
18	00040000 _H	0 = Pattern background colour modified (<u>patt-bgcol</u>)
21-19	00380000 _H	Always 111 ₂ (but not used)
26	04000000 _H	1 = Record contains font formatting block (<u>font</u>)
28	10000000 _H	1 = Record contains border formatting block (<u>bord</u>)
29	20000000 _H	1 = Record contains pattern formatting block (<u>patt</u>)

• **Font Formatting Block**

Offset	Size	Contents												
0	64	Not used												
64	4	Font height (in twips = $\frac{1}{20}$ of a point); or FFFFFFFF _H to preserve the cell font height												
68	4	Font options: <table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>00000002_H</td> <td>Posture: 0 = Normal; 1 = Italic (only if <u>font-style</u> = 0)</td> </tr> <tr> <td>7</td> <td>00000080_H</td> <td>Cancellation: 0 = Off; 1 = On (only if <u>font-canc</u> = 0)</td> </tr> </tbody> </table>	Bit	Mask	Contents	1	00000002 _H	Posture: 0 = Normal; 1 = Italic (only if <u>font-style</u> = 0)	7	00000080 _H	Cancellation: 0 = Off; 1 = On (only if <u>font-canc</u> = 0)			
Bit	Mask	Contents												
1	00000002 _H	Posture: 0 = Normal; 1 = Italic (only if <u>font-style</u> = 0)												
7	00000080 _H	Cancellation: 0 = Off; 1 = On (only if <u>font-canc</u> = 0)												
72	2	Font weight (100-1000, only if <u>font-style</u> = 0). Standard values are 0190 _H (400) for normal text and 02BC _H (700) for bold text.												
74	2	Escapement type (only if <u>font-esc</u> = 0): 0000 _H = None; 0001 _H = Superscript; 0002 _H = Subscript												
76	1	Underline type (only if <u>font-underl</u> = 0): 00 _H = None 01 _H = Single 21 _H = Single accounting 02 _H = Double 22 _H = Double accounting												
77	3	Not used												
80	4	Font colour index (→6.70); or FFFFFFFF _H to preserve the cell font colour												
84	4	Not used												
88	4	Option flags for modified font attributes: <table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>00000002_H</td> <td>0 = Font style (posture or boldness) modified (<u>font-style</u>)</td> </tr> <tr> <td>4-3</td> <td>00000018_H</td> <td>Always 11₂ (but not used)</td> </tr> <tr> <td>7</td> <td>00000080_H</td> <td>0 = Font cancellation modified (<u>font-canc</u>)</td> </tr> </tbody> </table>	Bit	Mask	Contents	1	00000002 _H	0 = Font style (posture or boldness) modified (<u>font-style</u>)	4-3	00000018 _H	Always 11 ₂ (but not used)	7	00000080 _H	0 = Font cancellation modified (<u>font-canc</u>)
Bit	Mask	Contents												
1	00000002 _H	0 = Font style (posture or boldness) modified (<u>font-style</u>)												
4-3	00000018 _H	Always 11 ₂ (but not used)												
7	00000080 _H	0 = Font cancellation modified (<u>font-canc</u>)												
92	4	0 = Escapement type modified (<u>font-esc</u>)												
96	4	0 = Underline type modified (<u>font-underl</u>)												
100	16	Not used												
116	2	0001 _H												

• **Border Formatting Block**

Offset	Size	Contents															
0	2	Border line styles: <table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>3-0</td> <td>000F_H</td> <td>Left line style (only if <u>bord-left</u> = 0, →3.10)</td> </tr> <tr> <td>7-4</td> <td>00F0_H</td> <td>Right line style (only if <u>bord-right</u> = 0, →3.10)</td> </tr> <tr> <td>11-8</td> <td>0F00_H</td> <td>Top line style (only if <u>bord-top</u> = 0, →3.10)</td> </tr> <tr> <td>15-12</td> <td>F000_H</td> <td>Bottom line style (only if <u>bord-bot</u> = 0, →3.10)</td> </tr> </tbody> </table>	Bit	Mask	Contents	3-0	000F _H	Left line style (only if <u>bord-left</u> = 0, →3.10)	7-4	00F0 _H	Right line style (only if <u>bord-right</u> = 0, →3.10)	11-8	0F00 _H	Top line style (only if <u>bord-top</u> = 0, →3.10)	15-12	F000 _H	Bottom line style (only if <u>bord-bot</u> = 0, →3.10)
Bit	Mask	Contents															
3-0	000F _H	Left line style (only if <u>bord-left</u> = 0, →3.10)															
7-4	00F0 _H	Right line style (only if <u>bord-right</u> = 0, →3.10)															
11-8	0F00 _H	Top line style (only if <u>bord-top</u> = 0, →3.10)															
15-12	F000 _H	Bottom line style (only if <u>bord-bot</u> = 0, →3.10)															
2	4	Border line colour indexes: <table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>6-0</td> <td>0000007F_H</td> <td>Colour index (→6.70) for left line (only if <u>bord-left</u> = 0)</td> </tr> <tr> <td>13-7</td> <td>00003F80_H</td> <td>Colour index (→6.70) for right line (only if <u>bord-right</u> = 0)</td> </tr> <tr> <td>22-16</td> <td>007F0000_H</td> <td>Colour index (→6.70) for top line (only if <u>bord-top</u> = 0)</td> </tr> <tr> <td>29-23</td> <td>3F800000_H</td> <td>Colour index (→6.70) for bottom line (only if <u>bord-bot</u> = 0)</td> </tr> </tbody> </table>	Bit	Mask	Contents	6-0	0000007F _H	Colour index (→6.70) for left line (only if <u>bord-left</u> = 0)	13-7	00003F80 _H	Colour index (→6.70) for right line (only if <u>bord-right</u> = 0)	22-16	007F0000 _H	Colour index (→6.70) for top line (only if <u>bord-top</u> = 0)	29-23	3F800000 _H	Colour index (→6.70) for bottom line (only if <u>bord-bot</u> = 0)
Bit	Mask	Contents															
6-0	0000007F _H	Colour index (→6.70) for left line (only if <u>bord-left</u> = 0)															
13-7	00003F80 _H	Colour index (→6.70) for right line (only if <u>bord-right</u> = 0)															
22-16	007F0000 _H	Colour index (→6.70) for top line (only if <u>bord-top</u> = 0)															
29-23	3F800000 _H	Colour index (→6.70) for bottom line (only if <u>bord-bot</u> = 0)															
6	2	Not used															

• **Pattern Formatting Block**

Offset	Size	Contents									
0	2	Fill pattern style:									
		<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>15-10</td> <td>FC00_H</td> <td>Fill pattern style (only if <code>patt-style = 0</code>, →3.11)</td> </tr> </tbody> </table>	Bit	Mask	Contents	15-10	FC00 _H	Fill pattern style (only if <code>patt-style = 0</code> , →3.11)			
Bit	Mask	Contents									
15-10	FC00 _H	Fill pattern style (only if <code>patt-style = 0</code> , →3.11)									
2	2	Fill pattern colour indexes:									
		<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>6-0</td> <td>007F_H</td> <td>Colour index (→6.70) for pattern (only if <code>patt-col = 0</code>)</td> </tr> <tr> <td>13-7</td> <td>3F80_H</td> <td>Colour index (→6.70) for pattern background (only if <code>patt-bgcol = 0</code>)</td> </tr> </tbody> </table>	Bit	Mask	Contents	6-0	007F _H	Colour index (→6.70) for pattern (only if <code>patt-col = 0</code>)	13-7	3F80 _H	Colour index (→6.70) for pattern background (only if <code>patt-bgcol = 0</code>)
Bit	Mask	Contents									
6-0	007F _H	Colour index (→6.70) for pattern (only if <code>patt-col = 0</code>)									
13-7	3F80 _H	Colour index (→6.70) for pattern background (only if <code>patt-bgcol = 0</code>)									

6.17 CODEPAGE

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0042 _H	0042 _H	0042 _H	0042 _H	0042 _H	0042 _H	0042 _H	0042 _H

This record stores the text encoding used to write byte strings, stored as MS Windows code page identifier.

- ! The CODEPAGE record in BIFF8 always contains the code page 1200 (UTF-16, →3.4). Therefore it is not possible to obtain the encoding used for a protection password (it is not UTF-16).

Record CODEPAGE, BIFF2-BIFF8:

Offset	Size	Contents
0	2	Code page identifier used for byte string text encoding ¹⁷ : 016F _H = 367 = ASCII 01B5 _H = 437 = IBM PC CP-437 (US) 02D0 _H = 720 = IBM PC CP-720 (OEM Arabic) 02E1 _H = 737 = IBM PC CP-737 (Greek) 0307 _H = 775 = IBM PC CP-775 (Baltic) 0352 _H = 850 = IBM PC CP-850 (Latin I) 0354 _H = 852 = IBM PC CP-852 (Latin II (Central European)) 0357 _H = 855 = IBM PC CP-855 (Cyrillic) 0359 _H = 857 = IBM PC CP-857 (Turkish) 035A _H = 858 = IBM PC CP-858 (Multilingual Latin I with Euro) 035C _H = 860 = IBM PC CP-860 (Portuguese) 035D _H = 861 = IBM PC CP-861 (Icelandic) 035E _H = 862 = IBM PC CP-862 (Hebrew) 035F _H = 863 = IBM PC CP-863 (Canadian (French)) 0360 _H = 864 = IBM PC CP-864 (Arabic) 0361 _H = 865 = IBM PC CP-865 (Nordic) 0362 _H = 866 = IBM PC CP-866 (Cyrillic (Russian)) 0365 _H = 869 = IBM PC CP-869 (Greek (Modern)) 036A _H = 874 = Windows CP-874 (Thai) 03A4 _H = 932 = Windows CP-932 (Japanese Shift-JIS) 03A8 _H = 936 = Windows CP-936 (Chinese Simplified GBK) 03B5 _H = 949 = Windows CP-949 (Korean (Wansung)) 03B6 _H = 950 = Windows CP-950 (Chinese Traditional BIG5) 04B0 _H = 1200 = UTF-16 (BIFF8) 04E2 _H = 1250 = Windows CP-1250 (Latin II) (Central European) 04E3 _H = 1251 = Windows CP-1251 (Cyrillic) 04E4 _H = 1252 = Windows CP-1252 (Latin I) (BIFF4-BIFF7) 04E5 _H = 1253 = Windows CP-1253 (Greek) 04E6 _H = 1254 = Windows CP-1254 (Turkish) 04E7 _H = 1255 = Windows CP-1255 (Hebrew) 04E8 _H = 1256 = Windows CP-1256 (Arabic) 04E9 _H = 1257 = Windows CP-1257 (Baltic) 04EA _H = 1258 = Windows CP-1258 (Vietnamese) 0551 _H = 1361 = Windows CP-1361 (Korean (Johab)) 2710 _H = 10000 = Apple Roman 8000 _H = 32768 = Apple Roman 8001 _H = 32769 = Windows CP-1252 (Latin I) (BIFF2-BIFF3)

¹⁷ For more information see http://en.wikipedia.org/wiki/Character_encoding.

6.18 COLINFO

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	007D _H	007D _H	007D _H	007D _H	007D _H	007D _H	007D _H

This record specifies the width and default cell formatting for a given range of columns.

In BIFF3, if a COLINFO record is missing for a column, the width specified in the record DEFCOLWIDTH (→6.29) is used instead.

In BIFF4-BIFF7, the width set in this record is only used, if the corresponding bit for this column is cleared in the GCW record (→6.47), otherwise the column width set in the DEFCOLWIDTH record (→6.29) is used (the STANDARDWIDTH record (→6.97) is always ignored in this case).

In BIFF8, if a COLINFO record is missing for a column, the width specified in the record STANDARDWIDTH (→6.97) is used. If this record is also missing, the column width of the record DEFCOLWIDTH (→6.29) is used instead.

This record also specifies a default XF record (→6.115) to use for cells in the columns that are not described by any cell record (which contain the XF index for that cell). Additionally, the option flags field contains hidden, outline, and collapsed options applied at the columns.

In BIFF2, the column width is stored in the record COLWIDTH (→6.20) and default column formatting in the record COLUMNDEFAULT (→6.19).

Record COLINFO, BIFF3-BIFF8:

Offset	Size	Contents												
0	2	Index to first column in the range												
2	2	Index to last column in the range												
4	2	Width of the columns in $\frac{1}{256}$ of the width of the zero character, using default font (first FONT record in the file)												
6	2	Index to XF record (→6.115) for default column formatting												
8	2	Option flags: <table border="1" data-bbox="491 1115 1481 1272"> <thead> <tr> <th>Bits</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0001_H</td> <td>1 = Columns are hidden</td> </tr> <tr> <td>10-8</td> <td>0700_H</td> <td>Outline level of the columns (0 = no outline)</td> </tr> <tr> <td>12</td> <td>1000_H</td> <td>1 = Columns are collapsed</td> </tr> </tbody> </table>	Bits	Mask	Contents	0	0001 _H	1 = Columns are hidden	10-8	0700 _H	Outline level of the columns (0 = no outline)	12	1000 _H	1 = Columns are collapsed
Bits	Mask	Contents												
0	0001 _H	1 = Columns are hidden												
10-8	0700 _H	Outline level of the columns (0 = no outline)												
12	1000 _H	1 = Columns are collapsed												
10	2	Not used												

6.19 COLUMNDEFAULT

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0020 _H	—	—	—	—	—	—	—

This record contains default formatting attributes for a given range of columns. The formatting attributes are stored separate for each column in the given range. From BIFF3 on the record COLINFO (→6.18) is used to specify default column formatting.

Record COLUMNDEFAULT, BIFF2:

Offset	Size	Contents
0	2	Index to first column referred in this record (<u>f</u> <u>c</u>)
2	2	Index to last column referred in this record (<u>l</u> <u>c</u>)
4	3· <u>n</u> <u>c</u>	List of <u>n</u> <u>c</u> = <u>l</u> <u>c</u> - <u>f</u> <u>c</u> +1 cell attribute structures (3 bytes each, →3.12)
4+3· <u>n</u> <u>c</u>	2	Not used

6.20 COLWIDTH

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0024 _H	—	—	—	—	—	—	—

This record specifies the width for a given range of columns. If a column does not have a corresponding COLWIDTH record, the width specified by the record DEFCOLWIDTH (→6.29) is used instead. From BIFF3 on the record COLINFO (→6.18) is used to specify the column width.

Offset	Size	Contents
0	1	Index to first column
1	1	Index to last column
2	2	Width of the columns in $1/256$ of the width of the zero character, using default font (first FONT record in the file)

6.21 CONDFMT

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	—	—	01B0 _H	01B0 _H

This record is part of the Conditional Formatting Table (→5.12). It contains a list of cell range addresses for all cells with equal conditional formatting. It is followed by a list of CF records (up to 3 records in BIFF8, →6.16) which contain the conditions and cell formatting attributes.

Record CONDFMT, BIFF8:

Offset	Size	Contents
0	2	Number of following CF records
2	2	1 = Conditionally formatted cells need recalculation or redraw
4	8	Cell range address of the range enclosing all conditionally formatted ranges (→3.13.1)
12	var.	Cell range address list of all conditionally formatted ranges (→3.13.2)

The record contains the cell range address list and the minimal cell range enclosing all the ranges from this list.

Example: This record describes the cells B9, G3:I8 and E15. The address of the cell range enclosing this list is B3:I15.

6.22 CONTINUE

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
003C _H	003C _H	003C _H	003C _H	003C _H	003C _H	003C _H	003C _H

Whenever the content of a record exceeds the given limits (see table), the record must be split. Several CONTINUE records containing the additional data are added after the parent record.

BIFF version	Maximum data size of a record
BIFF2-BIFF7	2080 bytes (2084 bytes including record header)
BIFF8	8224 bytes (8228 bytes including record header)

Record CONTINUE, BIFF2-BIFF8:

Offset	Size	Contents
0	var.	Data continuation of the previous record

Unicode strings are split in a special way. At the beginning of each CONTINUE record the option flags byte is repeated. Only the character size flag will be set in this flags byte, the Rich-Text flag and the Far-East flag are set to zero.

- ! In each CONTINUE record it is possible that the character size changes from 8-bit characters to 16-bit characters and vice versa.
- ! Never a Unicode string is split until and including the first character. That means, all header fields (string length, option flags, optional Rich-Text size, and optional Far-East data size) and the first character of the string have to occur together in the leading record, or have to be moved completely into the CONTINUE record.
- ! Formatting runs (→3.2) cannot be split between their components (character index and FONT record index). If a string is split between two formatting runs, the option flags field will not be repeated in the CONTINUE record.

Example: The remaining size of a record may be 10 bytes (it has 8214 bytes of data). Now the string “ABCDEFGHØI” has to be stored in this record. “Ø” may be a special character with the character code 1234_H.

Note: The records are shown with their headers to make the example clearer.

Offset	Size	Contents	Description
0	2		Any record identifier
2	2	2020 _H (8224)	Record data size
4	8214		Any data
8218	2	000A _H (10)	Unicode string character count
8220	1	00 _H	Unicode string option flags (8-bit characters)
8221	7	41 _H 42 _H ... 47 _H	8-bit character array “ABCDEFGH”
8228	2	003C _H	Record identifier CONTINUE
8230	2	0007 _H (7)	Record data size
8232	1	01 _H	Unicode string option flags (16-bit characters)
8233	2	0048 _H	16-bit character “H”
8235	2	1234 _H	16-bit character “Ø”
8237	2	0049 _H	16-bit character “I”

6.23 COUNTRY

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	008C _H	008C _H	008C _H	008C _H	008C _H	008C _H	008C _H

This record stores two Windows country identifiers. The first represents the user interface language of the Excel version that has saved the file, and the second represents the system regional settings at the time the file was saved.

Record COUNTRY, BIFF3-BIFF8:

Offset	Size	Contents
0	2	Windows country identifier of the user interface language of Excel
2	2	Windows country identifier of the system regional settings

The following table shows most of the used country identifiers. Most of these identifiers are equal to the international country calling codes¹⁸.

ID	Country	ID	Country	ID	Country	ID	Country	ID	Country	ID	Country
1	USA ¹⁹	66	Thailand	240	Equatorial Guinea ²²	350	Gibraltar ²²	506	Costa Rica	692	Marshall Islands ²²
2	Canada ²⁰	81	Japan	241	Gabon ²²	351	Portugal	507	Panama	850	North Korea ²²
7	Russia ²¹	82	South Korea	242	Congo ²²	352	Luxembourg	508	St. Pierre ²²	852	Hong Kong S.A.R.
20	Egypt	84	Viet Nam	243	Zaire ²³	353	Ireland	509	Haiti ²²	853	Macao S.A.R.
27	South Africa	86	PR China	244	Angola ²²	354	Iceland	590	Guadeloupe ²²	855	Cambodia ²²
30	Greece	90	Turkey	245	Guinea-Bissau ²²	355	Albania	591	Bolivia	856	Laos ²²
31	Netherlands	91	India	246	Diego Garcia ²²	356	Malta ²²	592	Guyana ²²	880	Bangladesh ²²
32	Belgium	92	Pakistan	247	Ascension Island ²²	357	Cyprus ²²	593	Ecuador	886	Taiwan
33	France	93	Afghanistan ²²	248	Seychelles ²²	358	Finland	594	French Guiana ²²	960	Maldives
34	Spain	94	Sri Lanka ²²	249	Sudan ²²	359	Bulgaria	595	Paraguay	961	Lebanon
36	Hungary	95	Burma (Myanmar) ²²	250	Rwanda ²²	370	Lithuania	596	Martinique ²²	962	Jordan
39	Italy	212	Morocco	251	Ethiopia ²²	371	Latvia	597	Suriname ²²	963	Syria
40	Romania	213	Algeria	252	Somalia ²²	372	Estonia	598	Uruguay	964	Iraq
41	Switzerland	216	Tunisia	253	Djibouti ²²	373	Moldova ²²	599	Netherlands Antilles ²²	965	Kuwait
43	Austria	218	Libya	254	Kenya	374	Armenia	670	East Timor ²²	966	Saudi Arabia
44	United Kingdom	220	Gambia ²²	255	Tanzania ²²	375	Belarus ²⁶	672	Antarctica ²²	967	Yemen
45	Denmark	221	Senegal ²³	256	Uganda ²²	376	Andorra ²²	673	Brunei Darussalam	968	Oman
46	Sweden	222	Mauritania ²²	257	Burundi ²²	377	Monaco	674	Narupu ²²	970	Palestine ²²
47	Norway	223	Mali ²³	258	Mozambique ²²	378	San Marino ²²	675	Papua New Guinea ²²	971	U.A.E.
48	Poland	224	Guinea ²²	259	Zanzibar ²²	379	Vatican City ²²	676	Tonga ²²	972	Israel
49	Germany	225	Côte d'Ivoire ²³	260	Zambia ²²	380	Ukraine	677	Solomon Islands ²²	973	Bahrain
51	Peru	226	Burkina Faso ²²	261	Madagascar ²²	381	Serbia	678	Vanuatu ²²	974	Qatar
52	Mexico	227	Niger ²²	262	Reunion Island ²³	385	Croatia	679	Fiji ²²	975	Bhutan ²²
53	Cuba ²²	228	Togo ²²	263	Zimbabwe	386	Slovenia	680	Palau ²²	976	Mongolia
54	Argentina	229	Benin ²²	264	Namibia ²²	387	Bosnia, Herzegovina ²²	681	Wallis and Futuna ²²	977	Nepal ²²
55	Brazil	230	Mauritius ²²	265	Malawi ²²	389	Macedonia	682	Cook Islands ²²	981	Iran ²⁵
56	Chile	231	Liberia ²²	266	Lesotho ²²	420	Czech	683	Niue Island ²²	992	Tajikistan ²⁶
57	Colombia	232	Sierra Leone ²²	267	Botswana ²²	421	Slovak	684	American Samoa ²²	993	Turkmenistan ²²
58	Venezuela	233	Ghana ²²	268	Swaziland ²²	423	Liechtenstein ²⁴	685	Western Samoa ²²	994	Azerbaijan
60	Malaysia	234	Nigeria ²²	269	Comoros, Mayotte ²²	500	Falkland Islands ²²	686	Kiribati ²²	995	Georgia
61	Australia	235	Chad ²²	290	St. Helena ²²	501	Belize	687	New Caledonia ²²	996	Kyrgyzstan
62	Indonesia	236	Central African Rep. ²²	291	Eritrea ²²	502	Guatemala	688	Tuvalu ²²	998	Uzbekistan ²⁶
63	Philippines	237	Cameroon ²³	297	Aruba ²²	503	El Salvador	689	French Polynesia ²²		
64	New Zealand	238	Cape Verde ²²	298	Faeroe Islands	504	Honduras	690	Tokelau ²²		
65	Singapore	239	Sao Tome ²²	299	Green Island ²²	505	Nicaragua	691	Micronesia ²²		

¹⁸ Source: http://en.wikipedia.org/wiki/List_of_country_calling_codes

¹⁹ Including the countries of the North America Numbering Plan (NANP), e.g. Bahamas, Dominican Republic, Jamaica, Puerto Rico.

²⁰ Real country calling code of Canada is 1.

²¹ Including Kazakhstan and Tatarstan.

²² Not used in Windows.

²³ Windows uses France (33) instead.

²⁴ Windows uses Switzerland (41) instead.

²⁵ Real country calling code of Iran is 98.

²⁶ Windows uses Russia (7) instead.

6.24 CRN

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	005A _H	005A _H	005A _H	005A _H	005A _H	005A _H	005A _H

This record stores the contents of an external cell or cell range. An external cell range has one row only. If a cell range spans over more than one row, several CRN records will be created. See →5.10 for details about external references.

Record CRN, BIFF3-BIFF8:

Offset	Size	Contents
0	1	Index to last column inside of the referenced sheet (<u>l</u> c)
1	1	Index to first column inside of the referenced sheet (<u>f</u> c)
2	2	Index to row inside of the referenced sheet
4	var.	List of <u>l</u> c- <u>f</u> c+1 cached values (→3.8.1)

6.25 DATEMODE

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0022 _H	0022 _H	0022 _H	0022 _H	0022 _H	0022 _H	0022 _H	0022 _H

This record specifies the base date for displaying date values. All dates are stored as count of days past this base date. In BIFF2-BIFF4 this record is part of the *Calculation Settings Block* (→5.3). In BIFF5-BIFF8 it is stored in the *Workbook Globals Substream*.

Record DATEMODE, BIFF2-BIFF8:

Offset	Size	Contents
0	2	0 = Base date is 1899-Dec-31 (the cell value 1 represents 1900-Jan-01) 1 = Base date is 1904-Jan-01 (the cell value 1 represents 1904-Jan-02)

6.26 DBCELL

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	00D7 _H	00D7 _H	00D7 _H	00D7 _H

This record is written once in a *Row Block*. It contains relative offsets to calculate the stream position of the first cell record for each row. The offset list in this record contains as many offsets as ROW records are present in the *Row Block*. For details about calculation of cell record positions see →5.7.

Record DBCELL, BIFF5-BIFF8:

Offset	Size	Contents
0	4	Relative offset to first ROW record in the <i>Row Block</i> (difference between record position of this record and the ROW record; positive offset for an earlier stream position)
4	2· <u>nm</u>	Array of <u>nm</u> relative offsets (16-bit values) to calculate stream position of the first cell record for the respective row (→5.7.1). <u>nm</u> is the number of ROW records in this <i>Row Block</i>

6.27 DCONREF – Data Consolidation Reference

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0051 _H	0051 _H	0051 _H	0051 _H	0051 _H	0051 _H	0051 _H	0051 _H

2do

6.28 DEFAULTROWHEIGHT

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0025 _H	0225 _H	0225 _H	0225 _H	0225 _H	0225 _H	0225 _H	0225 _H

This record specifies the default height and default flags for rows that do not have a corresponding ROW record (→6.83).

Record DEFAULTROWHEIGHT, BIFF2:

Offset	Size	Contents
0	2	Default height for unused rows, in twips = $\frac{1}{20}$ of a point

Record DEFAULTROWHEIGHT, BIFF3-BIFF8:

Offset	Size	Contents															
0	2	Option flags: <table border="1" data-bbox="491 1691 1471 1877"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0001_H</td> <td>1 = Row height and default font height do not match</td> </tr> <tr> <td>1</td> <td>0002_H</td> <td>1 = Row is hidden</td> </tr> <tr> <td>2</td> <td>0004_H</td> <td>1 = Additional space above the row</td> </tr> <tr> <td>3</td> <td>0008_H</td> <td>1 = Additional space below the row</td> </tr> </tbody> </table>	Bit	Mask	Contents	0	0001 _H	1 = Row height and default font height do not match	1	0002 _H	1 = Row is hidden	2	0004 _H	1 = Additional space above the row	3	0008 _H	1 = Additional space below the row
Bit	Mask	Contents															
0	0001 _H	1 = Row height and default font height do not match															
1	0002 _H	1 = Row is hidden															
2	0004 _H	1 = Additional space above the row															
3	0008 _H	1 = Additional space below the row															
2	2	Default height for unused rows, in twips = $\frac{1}{20}$ of a point															

6.29 DEFCOLWIDTH

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0055 _H	0055 _H	0055 _H	0055 _H	0055 _H	0055 _H	0055 _H	0055 _H

This record specifies the default column width for columns that do not have a specific width set using the records COLWIDTH (BIFF2, →6.20), COLINFO (BIFF3-BIFF8, →6.18), or STANDARDWIDTH (→6.97).

Record DEFCOLWIDTH, BIFF2-BIFF8:

Offset	Size	Contents
0	2	Column width in characters, using the width of the zero character from default font (first FONT record in the file). Excel adds some extra space to the default width, depending on the default font and default font size. The algorithm how to exactly calculate the resulting column width is not known.
<p>Example: The default width of 8 set in this record results in a column width of 8.43 using Arial font with a size of 10 points.</p>		

6.30 DELTA

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0010 _H	0010 _H	0010 _H	0010 _H	0010 _H	0010 _H	0010 _H	0010 _H

This record is part of the *Calculation Settings Block* (→5.3). It stores the maximum change of the result to exit an iteration.

Record DELTA, BIFF2-BIFF8:

Offset	Size	Contents
0	8	Maximum change in iteration (IEEE 754 floating-point value, 64-bit double precision)

6.31 DIMENSIONS

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0000 _H	0200 _H	0200 _H	0200 _H	0200 _H	0200 _H	0200 _H	0200 _H

This record contains the range address of the used area in the current sheet.

Record DIMENSIONS, BIFF2:

Offset	Size	Contents
0	2	Index to first used row
2	2	Index to last used row, increased by 1
4	2	Index to first used column
6	2	Index to last used column, increased by 1

Record DIMENSIONS, BIFF3-BIFF7:

Offset	Size	Contents
0	2	Index to first used row
2	2	Index to last used row, increased by 1
4	2	Index to first used column
6	2	Index to last used column, increased by 1
8	2	Not used

Record DIMENSIONS, BIFF8:

Offset	Size	Contents
0	4	Index to first used row
4	4	Index to last used row, increased by 1
8	2	Index to first used column
10	2	Index to last used column, increased by 1
12	2	Not used

6.32 DSF – Double Stream File

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	—	—	0161 _H	0161 _H

This record specifies if the file contains an additional BIFF5/BIFF7 workbook stream.

Record DSF, BIFF8:

Offset	Size	Contents
0	2	0 = Only the BIFF8 “Workbook” stream is present 1 = Additional BIFF5/BIFF7 “Book” stream is in the file

A double stream file can be read by Excel 5.0 and Excel 95, and still contains all new features added to BIFF8 (which are left out in the BIFF5/BIFF7 “Book” stream).

6.33 DV – Data Validity Settings

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	—	—	01BE _H	01BE _H

This record is part of the *Data Validity Table* (→5.14). It stores data validity settings and a list of cell ranges which contain these settings. The “prompt box” appears while editing such a cell. The “error box” appears, if the entered value does not fit the conditions. The data validity settings of a sheet are stored in a sequential list of DV records. This list is precluded by an DVAL record (→6.34). If a string is empty and the default text should appear in the prompt box or error box, the string must contain a single zero character (string length will be 1).

Record DV, BIFF8:

Offset	Size	Contents
0	4	Option flags (see below)
4	var.	Title of the prompt box (Unicode string, 16-bit string length, →3.4)
var.	var.	Title of the error box (Unicode string, 16-bit string length, →3.4)
var.	var.	Text of the prompt box (Unicode string, 16-bit string length, →3.4)
var.	var.	Text of the error box (Unicode string, 16-bit string length, →3.4)
var.	2	Size of the formula data for first condition (<u>sz1</u>)
var.	2	Not used
var.	<u>sz1</u>	Formula data for first condition (RPN token array without size field, →4)
var.	2	Size of the formula data for second condition (<u>sz2</u>)
var.	2	Not used
var.	<u>sz2</u>	Formula data for second condition (RPN token array without size field, →4)
var.	var.	Cell range address list with all affected ranges (→3.13.2)

Option flags field:

Bit	Mask	Contents								
3-0	0000000F _H	Data type: <table style="display: inline-table; vertical-align: top; margin-left: 20px;"> <tr> <td>00_H = Any value</td> <td>04_H = Date</td> </tr> <tr> <td>01_H = Integer values</td> <td>05_H = Time</td> </tr> <tr> <td>02_H = Decimal values</td> <td>06_H = Text length</td> </tr> <tr> <td>03_H = User defined list</td> <td>07_H = Formula</td> </tr> </table>	00 _H = Any value	04 _H = Date	01 _H = Integer values	05 _H = Time	02 _H = Decimal values	06 _H = Text length	03 _H = User defined list	07 _H = Formula
00 _H = Any value	04 _H = Date									
01 _H = Integer values	05 _H = Time									
02 _H = Decimal values	06 _H = Text length									
03 _H = User defined list	07 _H = Formula									
6-4	00000070 _H	Error style: <table style="display: inline-table; vertical-align: top; margin-left: 20px;"> <tr> <td>00_H = Stop</td> </tr> <tr> <td>01_H = Warning</td> </tr> <tr> <td>02_H = Info</td> </tr> </table>	00 _H = Stop	01 _H = Warning	02 _H = Info					
00 _H = Stop										
01 _H = Warning										
02 _H = Info										
7	00000080 _H	1 = In list type validity the string list is explicitly given in the formula								
8	00000100 _H	1 = Empty cells allowed								
9	00000200 _H	1 = Suppress the drop down arrow in list type validity								
18	00040000 _H	1 = Show prompt box if cell selected								
19	00080000 _H	1 = Show error box if invalid values entered								
23-20	00F00000 _H	Condition operator: <table style="display: inline-table; vertical-align: top; margin-left: 20px;"> <tr> <td>00_H = Between</td> <td>04_H = Greater than</td> </tr> <tr> <td>01_H = Not between</td> <td>05_H = Less than</td> </tr> <tr> <td>02_H = Equal</td> <td>06_H = Greater or equal</td> </tr> <tr> <td>03_H = Not equal</td> <td>07_H = Less or equal</td> </tr> </table>	00 _H = Between	04 _H = Greater than	01 _H = Not between	05 _H = Less than	02 _H = Equal	06 _H = Greater or equal	03 _H = Not equal	07 _H = Less or equal
00 _H = Between	04 _H = Greater than									
01 _H = Not between	05 _H = Less than									
02 _H = Equal	06 _H = Greater or equal									
03 _H = Not equal	07 _H = Less or equal									

In list type validity it is possible to enter an explicit string list. This string list is stored as tStr token (→4.8.2). The string items are separated by zero characters. There is no zero character at the end of the string list.

Example for a string list with the 3 strings A, B, and C: “A<00_H>B<00_H>C” (contained in a tStr token, string length is 5).

6.34 DVAL – Data Validity List

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	—	—	01B2 _H	01B2 _H

This record is the list header of the *Data Validity Table* (→5.14) in the current sheet.

Record DVAL, BIFF8:

Offset	Size	Contents												
0	2	Option flags: <table border="1" data-bbox="491 539 1481 689"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0001_H</td> <td>0 = Prompt box not visible 1 = Prompt box currently visible</td> </tr> <tr> <td>1</td> <td>0002_H</td> <td>0 = Prompt box has fixed position 1 = Prompt box appears at cell</td> </tr> <tr> <td>2</td> <td>0004_H</td> <td>1 = Cell validity data cached in following DV records</td> </tr> </tbody> </table>	Bit	Mask	Contents	0	0001 _H	0 = Prompt box not visible 1 = Prompt box currently visible	1	0002 _H	0 = Prompt box has fixed position 1 = Prompt box appears at cell	2	0004 _H	1 = Cell validity data cached in following DV records
Bit	Mask	Contents												
0	0001 _H	0 = Prompt box not visible 1 = Prompt box currently visible												
1	0002 _H	0 = Prompt box has fixed position 1 = Prompt box appears at cell												
2	0004 _H	1 = Cell validity data cached in following DV records												
2	4	Horizontal position of the prompt box, if it has fixed position, in pixel												
6	4	Vertical position of the prompt box, if it has fixed position, in pixel												
10	4	Object identifier of the drop down arrow object for a list box (→7), if a list box is visible at the current cursor position, FFFFFFFF _H otherwise												
14	4	Number of following DV records (→6.33)												

6.35 EFONT

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0045 _H	—	—	—	—	—	—	—

This record stores the colour of the font that is described in the preceding FONT record (→6.43). From BIFF3 on, the colour index is part of the FONT record.

Record EFONT, BIFF2:

Offset	Size	Contents										
0	2	Font colour index for the font described in the preceding FONT record: <table data-bbox="502 1368 1189 1525"> <tr> <td>0000_H = Black</td> <td>0004_H = Blue</td> </tr> <tr> <td>0001_H = White</td> <td>0005_H = Yellow</td> </tr> <tr> <td>0002_H = Red</td> <td>0006_H = Magenta</td> </tr> <tr> <td>0003_H = Green</td> <td>0007_H = Cyan</td> </tr> <tr> <td>7FFF_H = Automatic (system window text colour)</td> <td></td> </tr> </table> <p>These values are equal to the colour indexes described in the PALETTE record (→6.70).</p>	0000 _H = Black	0004 _H = Blue	0001 _H = White	0005 _H = Yellow	0002 _H = Red	0006 _H = Magenta	0003 _H = Green	0007 _H = Cyan	7FFF _H = Automatic (system window text colour)	
0000 _H = Black	0004 _H = Blue											
0001 _H = White	0005 _H = Yellow											
0002 _H = Red	0006 _H = Magenta											
0003 _H = Green	0007 _H = Cyan											
7FFF _H = Automatic (system window text colour)												

6.36 EOF – End of File

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
000A _H	000A _H	000A _H	000A _H	000A _H	000A _H	000A _H	000A _H

This record has no content. It indicates the end of a record block with leading BOF record (→6.8). This could be the end of the workbook globals, a worksheet, a chart, etc.

6.37 EXTERNCOUNT

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0016 _H	0016 _H	0016 _H	0016 _H	0016 _H	0016 _H	—	—

This record contains the number of following EXTERNSHEET records. In BIFF8 this record is omitted because there occurs only one EXTERNSHEET record. See →5.10.1 for details about external references in BIFF2-BIFF4 and →5.10.2 for BIFF5/BIFF7.

Record EXTERNCOUNT, BIFF2-BIFF7:

Offset	Size	Contents
0	2	Number of following EXTERNSHEET records (→6.39)

6.38 EXTERNNAME

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0023 _H	0223 _H	0223 _H	0223 _H	0023 _H	0023 _H	0023 _H	0023 _H

This record contains the name of an external defined name, the name of an add-in function, a DDE item or an OLE object storage identifier.

• EXTERNNAME in BIFF2-BIFF4

2do – partly wrong

The meaning of the name is dependent on the leading EXTERNSHEET record (→6.39). See →5.10.1 for details about external references in BIFF2-BIFF4.

Record EXTERNNAME, BIFF2-BIFF4:

Offset	Size	Contents
0	var.	2do

If the record contains an item of a DDE link, a list with cached values will be appended to the string. These values are used as results for the DDE link. They are saved row by row for a DDE link that spans over several cells. Note: Only the results of the DDE link (the contents of the referenced cells) are stored, not the results of the complete formulas.

Record EXTERNNAME for DDE items, BIFF2-BIFF4:

Offset	Size	Contents
0	var.	DDE item (byte string, 8-bit string length, →3.3)
[var.]	var.	(optional) Last received results of the DDE link (constant value array, →3.8.2)

• EXTERNNAME in BIFF5/BIFF7

All EXTERNNAME records follow an EXTERNSHEET record that contains only the name of the source document. EXTERNNAME records representing external defined names refer to earlier EXTERNSHEET records containing the sheet name in that document. See →5.10.2 for details about external references in BIFF5/BIFF7.

Record EXTERNNAME for external names, BIFF5/BIFF7:

Offset	Size	Contents
0	2	Option flags (see below)
2	2	<i>One-based</i> index to EXTERNSHEET record containing the sheet name or 0 for global defined names
4	2	Not used
6	var.	External name (byte string, 8-bit string length, →3.3). See NAME record (→6.66) for a list of built-in names, if the built-in flag is set in the option flags above.
var.	var.	Formula data (RPN token array, →4)

Record EXTERNNAME for add-in functions, BIFF5/BIFF7:

Offset	Size	Contents
0	2	Option flags (always 0000 _H for add-in function names)
2	4	Not used
6	var.	Add-in function name (byte string, 8-bit string length, →3.3)
var.	4	02 _H 00 _H 1C _H 17 _H (formula representing the #REF! error code)

Record EXTERNNAME for DDE links, BIFF5/BIFF7:

Offset	Size	Contents
0	2	Option flags (see below)
2	4	Not used
6	var.	DDE item (byte string, 8-bit string length, →3.3)
[var.]	var.	(optional) Last received results of the DDE link (constant value array, →3.8.2)

Record EXTERNNAME for OLE object links, BIFF5/BIFF7:

Offset	Size	Contents
0	2	Option flags (see below)
2	4	Storage identifier
6	2	01 _H 27 _H (byte string, 8-bit string length, containing a single apostroph)

• EXTERNNAME in BIFF8

The record must follow the SUPBOOK record (→6.100) that contains the URL of the source document. See →5.10.3 for details about external references in BIFF8.

Record EXTERNNAME for external names, BIFF8:

Offset	Size	Contents
0	2	Option flags (see below)
2	2	<i>One-based</i> index to sheet in preceding SUPBOOK record or 0 for global defined names
4	2	Not used
6	var.	External name (Unicode string, 8-bit string length, →3.4)
var.	var.	Formula data (RPN token array, →4)

Record EXTERNNAME for add-in functions, BIFF8:

Offset	Size	Contents
0	2	Option flags (always 0000 _H for add-in function names)
2	4	Not used
6	var.	Add-in function name (Unicode string, 8-bit string length, →3.4)
var.	4	02 _H 00 _H 1C _H 17 _H (formula representing the #REF! error code)

Record EXTERNNAME for DDE links, BIFF8:

Offset	Size	Contents
0	2	Option flags (see below)
2	4	Not used
6	var.	DDE item (Unicode string, 8-bit string length, →3.4)
[var.]	var.	(optional) Last received results of the DDE link (constant value array, →3.8.2)

Record EXTERNNAME for OLE object links, BIFF5-BIFF8:

Offset	Size	Contents
0	2	Option flags (see below)
2	4	Storage identifier
6	3	01 _H 00 _H 27 _H (Unicode string, 8-bit string length, containing a single apostroph)

• Option Flags

Option flags for external names (BIFF5-BIFF8)

Bit	Mask	Contents
0	0001 _H	0 = Standard name; 1 = Built-in name
4	0010 _H	Always 0 for external names

Option flags for DDE links (BIFF5-BIFF8)

Bit	Mask	Contents
1	0002 _H	0 = Manual DDE link; 1 = Automatic DDE link
3	0008 _H	1 = This is the "StdDocumentName" identifier
4	0010 _H	Always 0 for DDE links
14-5	7FE0 _H	Clipboard format of last successful update (FFF _H for "StdDocumentName")

Option flags for OLE object links (BIFF5-BIFF8)

Bit	Mask	Contents
1	0002 _H	0 = Manual OLE object link; 1 = Automatic OLE object link
4	0010 _H	Always 1 for OLE object links

6.39 EXTERNSHEET

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0017 _H	0017 _H	0017 _H	0017 _H	0017 _H	0017 _H	0017 _H	0017 _H

6.39.1 EXTERNSHEET in BIFF2-BIFF7

In the file format versions up to BIFF7 this record stores the name of an external document and a sheet name inside of this document. See →5.10.1 for details about external references in BIFF2-BIFF4 and →5.10.2 for BIFF5/BIFF7.

Record EXTERNSHEET, BIFF2-BIFF7:

Offset	Size	Contents
0	var.	Encoded document and sheet name (→3.9). Byte string, 8-bit string length (→3.3).

- ! The string length field is decreased by 1, if the EXTERNSHEET stores a reference to one of the own sheets (first character is 03_H). Example: The formula =Sheet2!A1 contains a reference to an EXTERNSHEET record with the string "<03_H>Sheet2". The string consists of 7 characters but the string length field contains the value 6.

If a formula uses an add-in function, a special EXTERNSHEET record will occur, followed by an EXTERNNAME record with the name of the function.

Record EXTERNSHEET for add-in functions, BIFF2-BIFF7:

Offset	Size	Contents
0	2	01 _H 34 _H (byte string, 8-bit string length, containing "#")

6.39.2 EXTERNSHEET in BIFF8

In BIFF8 the record stores a list with indexes to SUPBOOK records (list of REF structures, →6.100). See →5.10.3 for details about external references in BIFF8.

Record EXTERNSHEET, BIFF8:

Offset	Size	Contents												
0	2	Number of following REF structures (<u>nm</u>)												
2	6· <u>nm</u>	List of <u>nm</u> REF structures. Each REF contains the following data:												
		<table border="1"> <thead> <tr> <th>Offset</th> <th>Size</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>2</td> <td>Index to SUPBOOK record</td> </tr> <tr> <td>2</td> <td>2</td> <td>Index to first SUPBOOK sheet</td> </tr> <tr> <td>4</td> <td>2</td> <td>Index to last SUPBOOK sheet</td> </tr> </tbody> </table>	Offset	Size	Contents	0	2	Index to SUPBOOK record	2	2	Index to first SUPBOOK sheet	4	2	Index to last SUPBOOK sheet
Offset	Size	Contents												
0	2	Index to SUPBOOK record												
2	2	Index to first SUPBOOK sheet												
4	2	Index to last SUPBOOK sheet												

6.40 EXTSST – Extended SST

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	—	—	00FF _H	00FF _H

This record occurs in conjunction with the SST record (→6.96). It is used by Excel to create a hash table with stream offsets to the SST record to optimise string search operations. Excel may not shorten this record if strings are deleted from the shared string table, so the last part might contain invalid data. The stream indexes in this record divide the SST into portions containing a constant number of strings. See →5.11 for more information about shared string tables.

Record EXTSST, BIFF8:

Offset	Size	Contents												
0	2	Number of strings in a portion, this number is ≥ 8												
2	var.	List of OFFSET structures for all portions. Each OFFSET contains the following data:												
		<table border="1"> <thead> <tr> <th>Offset</th> <th>Size</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>4</td> <td>Absolute stream position of first string of the portion</td> </tr> <tr> <td>4</td> <td>2</td> <td>Position of first string of the portion inside of current record, including record header. This counter restarts at zero, if the SST record is continued with a CONTINUE record.</td> </tr> <tr> <td>6</td> <td>2</td> <td>Not used</td> </tr> </tbody> </table>	Offset	Size	Contents	0	4	Absolute stream position of first string of the portion	4	2	Position of first string of the portion inside of current record, including record header. This counter restarts at zero, if the SST record is continued with a CONTINUE record.	6	2	Not used
Offset	Size	Contents												
0	4	Absolute stream position of first string of the portion												
4	2	Position of first string of the portion inside of current record, including record header. This counter restarts at zero, if the SST record is continued with a CONTINUE record.												
6	2	Not used												

6.41 FILEPASS

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
002F _H	002F _H	002F _H	002F _H	002F _H	002F _H	002F _H	002F _H

This record is part of the file protection (→5.19). It contains information about the read/write password of the file. All record contents following this record will be encrypted (→5.19.1).

• Record Contents (BIFF2-BIFF7)

Record FILEPASS, BIFF2-BIFF7:

Offset	Size	Contents
0	2	Encryption key calculated from the read/write password (→5.19.1)
2	2	Hash value calculated from the read/write password (→5.18.4)

• Record Contents (BIFF8/BIFF8X)

From BIFF8 on it is possible to use different encryption algorithms.

Record FILEPASS, BIFF8/BIFF8X, for BIFF2-BIFF7 weak XOR encryption:

Offset	Size	Contents
0	2	0000 _H = BIFF2-BIFF7 weak XOR encryption
2	2	Encryption key calculated from the read/write password (→5.19.1)
4	2	Hash value calculated from the read/write password (→5.18.4)

Record FILEPASS, BIFF8/BIFF8X, for BIFF8 standard encryption:

Offset	Size	Contents
0	2	0001 _H = BIFF8 standard encryption or BIFF8X strong encryption
2	2	not used, should be equal to next field
4	2	0001 _H = BIFF8 standard encryption
6	16	Unique document identifier used to initialise the encryption algorithm
22	16	Encrypted document identifier used to verify the entered password
38	16	Digest used to verify the entered password

Record FILEPASS, BIFF8X only, for BIFF8X strong encryption:

Offset	Size	Contents									
0	2	0001 _H = BIFF8 standard encryption or BIFF8X strong encryption									
2	2	not used, should be equal to next field									
4	2	0002 _H = BIFF8X strong encryption									
6	4	Option flags: <table border="1" data-bbox="418 1594 1402 1706"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>00000004_H</td> <td>Always 1₂</td> </tr> <tr> <td>3</td> <td>00000008_H</td> <td>1 = <i>Do not</i> encrypt document properties</td> </tr> </tbody> </table>	Bit	Mask	Contents	2	00000004 _H	Always 1 ₂	3	00000008 _H	1 = <i>Do not</i> encrypt document properties
Bit	Mask	Contents									
2	00000004 _H	Always 1 ₂									
3	00000008 _H	1 = <i>Do not</i> encrypt document properties									
10	var.	<i>FILEPASS Info Block</i> (see below)									
var.	4	Size of unique document identifier (<u>u</u> <u>i</u> <u>d</u> <u>l</u>)									
var.	<u>u</u> <u>i</u> <u>d</u> <u>l</u>	Unique document identifier used to initialise the encryption algorithm									
var.	<u>u</u> <u>i</u> <u>d</u> <u>l</u>	Encrypted document identifier used to verify the entered password									
var.	4	Size of following verification digest (<u>d</u> <u>l</u>)									
var.	<u>d</u> <u>l</u>	Digest used to verify the entered password									

FILEPASS Info Block, BIFF8X only, for BIFF8X strong encryption:

Offset	Size	Contents
0	4	Size of the following data in this block, without this field
4	4	Repeated option flags from main record (see above)
8	4	Not used
12	4	Stream encryption algorithm identifier: 00006801 _H = RC4 (Ron's Code 4) 00006802 _H = SEAL (Secure Encryption Algorithm)
16	4	Password hashing algorithm identifier: 00008001 _H = MD2 (Message Digest 2) 00008002 _H = MD4 (Message Digest 4) 00008003 _H = MD5 (Message Digest 5) 00008004 _H = SHA-1 (Secure Hash Algorithm)
20	4	Hash key length (bits)
24	4	Cryptographic provider type: 00000001 _H = RSA 0000000C _H = RSA SChannel 0000000D _H = DSS and Diffie-Hellman 00000012 _H = DH SChannel 00000018 _H = RSA and AES
28	8	Not used
36	var.	Cryptographic provider name, Unicode character array with trailing null character

6.42 FILESHARING

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	005B _H	005B _H	005B _H	005B _H	005B _H	005B _H	005B _H

This record is part of the file protection (→5.19). It contains information about write protection, for instance the write protection password. The write protection state of the file is switched on with the WRITEPROT record (→6.112).

Record FILESHARING, BIFF2-BIFF8:

Offset	Size	Contents
0	2	1 = Recommend read-only state while loading the file
2	2	Hash value calculated from the read-only password (→5.18.4)
4	var.	User name of the file creator
		BIFF2-BIFF7: Byte string, 8-bit string length (→3.3)
		BIFF8: Unicode string, 16-bit string length (→3.4)

6.43 FONT

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0031 _H	0231 _H	0231 _H	0231 _H	0031 _H	0031 _H	0031 _H	0031 _H

This record contains information about a used font, including character formatting. All FONT records occur together in a sequential list. Other records referencing a FONT record contain an index into this list.

- ! The font with index 4 is omitted in all BIFF versions. This means the first four fonts have zero-based indexes, and
- ! the fifth font and all following fonts are referenced with one-based indexes.

Record FONT, BIFF2:

Offset	Size	Contents															
0	2	Height of the font (in twips = $\frac{1}{20}$ of a point)															
2	2	Option flags: <table border="1" data-bbox="491 667 1471 862"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0001_H</td> <td>1 = Characters are bold</td> </tr> <tr> <td>1</td> <td>0002_H</td> <td>1 = Characters are italic</td> </tr> <tr> <td>2</td> <td>0004_H</td> <td>1 = Characters are underlined</td> </tr> <tr> <td>3</td> <td>0008_H</td> <td>1 = Characters are struck out</td> </tr> </tbody> </table>	Bit	Mask	Contents	0	0001 _H	1 = Characters are bold	1	0002 _H	1 = Characters are italic	2	0004 _H	1 = Characters are underlined	3	0008 _H	1 = Characters are struck out
Bit	Mask	Contents															
0	0001 _H	1 = Characters are bold															
1	0002 _H	1 = Characters are italic															
2	0004 _H	1 = Characters are underlined															
3	0008 _H	1 = Characters are struck out															
4	var.	Font name (byte string, 8-bit string length, →3.3)															

An EFONT record (→6.35) containing the font colour may follow.

Record FONT, BIFF3-BIFF4:

Offset	Size	Contents															
0	2	Height of the font (in twips = $\frac{1}{20}$ of a point)															
2	2	Option flags: <table border="1" data-bbox="491 1099 1471 1294"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0001_H</td> <td>1 = Characters are bold</td> </tr> <tr> <td>1</td> <td>0002_H</td> <td>1 = Characters are italic</td> </tr> <tr> <td>2</td> <td>0004_H</td> <td>1 = Characters are underlined</td> </tr> <tr> <td>3</td> <td>0008_H</td> <td>1 = Characters are struck out</td> </tr> </tbody> </table>	Bit	Mask	Contents	0	0001 _H	1 = Characters are bold	1	0002 _H	1 = Characters are italic	2	0004 _H	1 = Characters are underlined	3	0008 _H	1 = Characters are struck out
Bit	Mask	Contents															
0	0001 _H	1 = Characters are bold															
1	0002 _H	1 = Characters are italic															
2	0004 _H	1 = Characters are underlined															
3	0008 _H	1 = Characters are struck out															
4	2	Colour index (→6.70)															
6	var.	Font name (byte string, 8-bit string length, →3.3)															

Record FONT, BIFF5-BIFF8:

Offset	Size	Contents															
0	2	Height of the font (in twips = $\frac{1}{20}$ of a point)															
2	2	Option flags: <table border="1" data-bbox="416 338 1401 524"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0001_H</td> <td>1 = Characters are bold (redundant, see below)</td> </tr> <tr> <td>1</td> <td>0002_H</td> <td>1 = Characters are italic</td> </tr> <tr> <td>2</td> <td>0004_H</td> <td>1 = Characters are underlined (redundant, see below)</td> </tr> <tr> <td>3</td> <td>0008_H</td> <td>1 = Characters are struck out</td> </tr> </tbody> </table>	Bit	Mask	Contents	0	0001 _H	1 = Characters are bold (redundant, see below)	1	0002 _H	1 = Characters are italic	2	0004 _H	1 = Characters are underlined (redundant, see below)	3	0008 _H	1 = Characters are struck out
Bit	Mask	Contents															
0	0001 _H	1 = Characters are bold (redundant, see below)															
1	0002 _H	1 = Characters are italic															
2	0004 _H	1 = Characters are underlined (redundant, see below)															
3	0008 _H	1 = Characters are struck out															
4	2	Colour index (→6.70)															
6	2	Font weight (100-1000). Standard values are 0190 _H (400) for normal text and 02BC _H (700) for bold text.															
8	2	Escapement type: 0000 _H = None 0001 _H = Superscript 0002 _H = Subscript															
10	1	Underline type: 00 _H = None 01 _H = Single 21 _H = Single accounting 02 _H = Double 22 _H = Double accounting															
11	1	Font family: 00 _H = None (unknown or don't care) 01 _H = Roman (variable width, serifed) 02 _H = Swiss (variable width, sans-serifed) 03 _H = Modern (fixed width, serifed or sans-serifed) 04 _H = Script (cursive) 05 _H = Decorative (specialised, for example Old English, Fraktur)															
12	1	Character set: 00 _H = 0 = ANSI Latin 01 _H = 1 = System default 02 _H = 2 = Symbol 4D _H = 77 = Apple Roman 80 _H = 128 = ANSI Japanese Shift-JIS 81 _H = 129 = ANSI Korean (Hangul) 82 _H = 130 = ANSI Korean (Johab) 86 _H = 134 = ANSI Chinese Simplified GBK 88 _H = 136 = ANSI Chinese Traditional BIG5 A1 _H = 161 = ANSI Greek A2 _H = 162 = ANSI Turkish A3 _H = 163 = ANSI Vietnamese B1 _H = 177 = ANSI Hebrew B2 _H = 178 = ANSI Arabic BA _H = 186 = ANSI Baltic CC _H = 204 = ANSI Cyrillic DE _H = 222 = ANSI Thai EE _H = 238 = ANSI Latin II (Central European) FF _H = 255 = OEM Latin I															
13	1	Not used															
14	var.	Font name: BIFF5/BIFF7: Byte string, 8-bit string length (→3.3) BIFF8: Unicode string, 8-bit string length (→3.4)															

The boldness and underline flags are still set in the options field, but not used on reading the font. Font weight and underline type are specified in separate fields instead.

6.44 FOOTER

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0015 _H	0015 _H	0015 _H	0015 _H	0015 _H	0015 _H	0015 _H	0015 _H

This record is part of the *Page Settings Block* (→5.4). It specifies the page footer string for the current worksheet. If this record is not present or completely empty (record size is 0), the sheet does not contain a page footer.

Record FOOTER for non-empty page footer, BIFF2-BIFF8:

Offset	Size	Contents
0	var.	Page footer string
		BIFF2-BIFF7: Non-empty byte string, 8-bit string length (→3.3)
		BIFF8: Non-empty Unicode string, 16-bit string length (→3.4)

The structure of the page footer string is equal to the page header string. For a detailed description see HEADER record, →6.51.

6.45 FORMAT

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
001E _H	001E _H	041E _H	041E _H	041E _H	041E _H	041E _H	041E _H

This record contains information about a number format. All FORMAT records occur together in a sequential list.

In BIFF2-BIFF4 other records referencing a FORMAT record contain a zero-based index into this list. From BIFF5 on the FORMAT record contains the index itself that will be used by other records.

Record FORMAT, BIFF2-BIFF3:

Offset	Size	Contents
0	var.	Number format string (byte string, 8-bit string length, →3.3)

Record FORMAT, BIFF4-BIFF7:

Offset	Size	Contents
0	2	BIFF4: Not used BIFF5/BIFF7: Format index used in other records
2	var.	Number format string (byte string, 8-bit string length, →3.3)

Record FORMAT, BIFF8:

Offset	Size	Contents
0	2	Format index used in other records
2	var.	Number format string (Unicode string, 16-bit string length, →3.4)

From BIFF5 on, the built-in number formats will be omitted. The built-in formats are dependent on the current regional settings of the operating system. The following table shows which number formats are used by default in a US-English environment. All indexes from 0 to 163 are reserved for built-in formats. The first user-defined format starts at 164.

The built-in number formats, BIFF5-BIFF8:

Index	Type	Format string	Index	Type	Format string
0	General	General	18	Time	h:mm AM/PM
1	Decimal	0	19	Time	h:mm:ss AM/PM
2	Decimal	0.00	20	Time	h:mm
3	Decimal	#,##0	21	Time	h:mm:ss
4	Decimal	#,##0.00	22 ²⁸	Date/Time	M/D/YY h:mm
5 ²⁷	Currency	"\$"#,##0_);("\$"#,##0)	37	Account.	_(#,##0_);(#,##0)
6 ²⁷	Currency	"\$"#,##0_);[Red]("\$"#,##0)	38	Account.	_(#,##0_);[Red](#,##0)
7 ²⁷	Currency	"\$"#,##0.00_);("\$"#,##0.00)	39	Account.	_(#,##0.00_);(#,##0.00)
8 ²⁷	Currency	"\$"#,##0.00_);[Red]("\$"#,##0.00)	40	Account.	_(#,##0.00_);[Red](#,##0.00)
9	Percent	0%	41 ²⁷	Currency	_"\$"* #,##0_);_"\$"* (#,##0);_"\$"* "-";_(@_)
10	Percent	0.00%	42 ^{27 29}	Currency	_* #,##0_);_* (#,##0);_* "-";_(@_)
11	Scientific	0.00E+00	43 ²⁷	Currency	_"\$"* #,##0.00_);_"\$"* (#,##0.00);_"\$"* "-"?;_(@_)
12	Fraction	# ?/?	44 ^{27 29}	Currency	_* #,##0.00_);_* (#,##0.00);_* "-"?;_(@_)
13	Fraction	# ??/??	45	Time	mm:ss
14 ²⁸	Date	M/D/YY	46	Time	[h]:mm:ss
15	Date	D-MMM-YY	47	Time	mm:ss.0
16	Date	D-MMM	48	Scientific	##0.0E+0
17	Date	MMM-YY	49	Text	@

²⁷ These formats are always written by Excel, though they are built-in. They contain the currency symbol of the current locale as plain text. It precedes or follows the value, according to the local settings. It is *not* required to write these formats.

²⁸ These formats are taken from the Microsoft Windows regional settings. Changes of these system settings will be reflected in the Excel document.

²⁹ These formats contain a blind currency symbol (an underscore precedes each character), if it follows the value. This happens for example in a German format with the Euro sign. Here the built-in format 42 looks like this: _* #,##0 _€ _);_* #,##0 _€ _);_* "- _€ _);_@ _)

Some of the built-in number formats are only used in special locales. As an example, the following table shows special Japanese formats:

Index	Type	Format string	Index	Type	Format string
27	Date	[\$-0411]GE.M.D	50	Date	[\$-0411]GE.M.D
28	Date	[\$-0411]GGGE年M月D日	51	Date	[\$-0411]GGGE年M月D日
29	Date	[\$-0411]GGGE年M月D日	52	Date	[\$-0411]YYYY年M月
30	Date	[\$-0411]M/D/YY	53	Date	[\$-0411]M月D日
31	Date	[\$-0411]YYYY年M月D日	54	Date	[\$-0411]GGGE年M月D日
32	Time	[\$-0411]h時mm分	55	Date	[\$-0411]YYYY年M月
33	Time	[\$-0411]h"時"mm"分"ss"秒"	56	Date	[\$-0411]M月D日
34	Date	[\$-0411]YYYY年M月	57	Date	[\$-0411]GE.M.D
35	Date	[\$-0411]M月D日	58	Date	[\$-0411]GGGE年M月D日
36	Date	[\$-0411]GE.M.D			

6.46 FORMULA

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0006 _H	0206 _H	0406 _H	0406 _H	0006 _H	0006 _H	0006 _H	0006 _H

This record contains the token array and the result of a formula cell.

• Record Contents

Record FORMULA, BIFF2:

Offset	Size	Contents
0	2	Index to row
2	2	Index to column
4	3	Cell attributes (→3.12)
7	8	Result of the formula (IEEE 754 floating-point value, 64-bit double precision)
15	1	0 = Do not recalculate, 1 = Recalculate always
16	var.	Formula data (RPN token array, →4)

Record FORMULA, BIFF3-BIFF4:

Offset	Size	Contents									
0	2	Index to row									
2	2	Index to column									
4	2	Index to XF record (→6.115)									
6	8	Result of the formula. See below for details.									
14	2	Option flags: <table border="1" data-bbox="416 1099 1402 1211"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0001_H</td> <td>1 = Recalculate always</td> </tr> <tr> <td>1</td> <td>0002_H</td> <td>1 = Calculate on open</td> </tr> </tbody> </table>	Bit	Mask	Contents	0	0001 _H	1 = Recalculate always	1	0002 _H	1 = Calculate on open
Bit	Mask	Contents									
0	0001 _H	1 = Recalculate always									
1	0002 _H	1 = Calculate on open									
16	var.	Formula data (RPN token array, →4)									

Record FORMULA, BIFF5-BIFF8:

Offset	Size	Contents												
0	2	Index to row												
2	2	Index to column												
4	2	Index to XF record (→6.115)												
6	8	Result of the formula. See below for details.												
14	2	Option flags: <table border="1" data-bbox="416 1529 1402 1682"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0001_H</td> <td>1 = Recalculate always</td> </tr> <tr> <td>1</td> <td>0002_H</td> <td>1 = Calculate on open</td> </tr> <tr> <td>3</td> <td>0008_H</td> <td>1 = Part of a shared formula</td> </tr> </tbody> </table>	Bit	Mask	Contents	0	0001 _H	1 = Recalculate always	1	0002 _H	1 = Calculate on open	3	0008 _H	1 = Part of a shared formula
Bit	Mask	Contents												
0	0001 _H	1 = Recalculate always												
1	0002 _H	1 = Calculate on open												
3	0008 _H	1 = Part of a shared formula												
16	4	Not used												
20	var.	Formula data (RPN token array, →4)												

• Result of the Formula

Dependent on the type of value the formula returns, the result field has the following format:

Result is a numeric value:

Offset	Size	Contents
0	8	IEEE 754 floating-point value (64-bit double precision)

Result is a string (the string follows in a STRING record, →6.98):

Offset	Size	Contents
0	1	00 _H (identifier for a string value)
1	5	Not used
6	2	FFFF _H

Note: In BIFF8 the string must not be empty. For empty cells there is a special identifier defined (see below).

Result is a Boolean value:

Offset	Size	Contents
0	1	01 _H (identifier for a Boolean value)
1	1	Not used
2	1	0 = FALSE, 1 = TRUE
3	3	Not used
6	2	FFFF _H

Result is an error value:

Offset	Size	Contents
0	1	02 _H (identifier for an error value)
1	1	Not used
2	1	Error code (→3.7)
3	3	Not used
6	2	FFFF _H

Result is an empty cell (BIFF8), for example an empty string:

Offset	Size	Contents
0	1	03 _H (identifier for an empty cell)
1	5	Not used
6	2	FFFF _H

6.47 GCW – Global Column Width

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	00AB _H	00AB _H	00AB _H	00AB _H	—	—

This record contains a bitfield with one bit for every column in the worksheet, describing how to set the column width for the respective column. If this record is omitted, all columns are using the default column width.

Record GCW, BIFF4-BIFF7:

Offset	Size	Contents
0	2	Size of the following bitfield (in bytes, always 0020 _H)
2	32	<p>Bit field with one bit for every column in the worksheet. The first byte of the bitfield contains flags for the first 8 columns (A to H), the second byte for the next 8 columns and so on. Bit 0 (mask 0x01) of each byte is for the leftmost column in the column interval, bit 7 (mask 0x80) is for the rightmost column.</p> <p>If a bit is set, the corresponding column uses the width set in the STANDARDWIDTH record (→6.97). If a bit is cleared, the corresponding column uses the width set in the COLINFO record (→6.18) for this column.</p> <p>If a bit is set, and the worksheet does not contain the STANDARDWIDTH record, or if the bit is cleared, and the worksheet does not contain the COLINFO record, the DEFCOLWIDTH record (→6.29) of the worksheet will be used instead.</p>

6.48 GRIDSET

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	0082 _H	0082 _H	0082 _H	0082 _H	0082 _H	0082 _H	0082 _H

This record specifies if the option to print sheet grid lines (record PRINTGRIDLINES, →6.75) has ever been changed.

Record GRIDSET, BIFF3-BIFF8:

Offset	Size	Contents
0	2	0 = Print grid lines option never changed 1 = Print grid lines option changed

6.49 GUTS

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	0080 _H	0080 _H	0080 _H	0080 _H	0080 _H	0080 _H	0080 _H

This record contains information about the layout of outline symbols.

Record GUTS, BIFF3-BIFF8:

Offset	Size	Contents
0	2	Width of the area to display row outlines (left of the sheet), in pixel
2	2	Height of the area to display column outlines (above the sheet), in pixel
4	2	Number of visible row outline levels (used row levels + 1; or 0, if not used)
6	2	Number of visible column outline levels (used column levels + 1; or 0, if not used)

6.50 HCENTER

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	0083 _H	0083 _H	0083 _H	0083 _H	0083 _H	0083 _H	0083 _H

This record is part of the *Page Settings Block* (→5.4). It specifies if the sheet is centred horizontally when printed.

Record HCENTER, BIFF3-BIFF8:

Offset	Size	Contents
0	2	0 = Print sheet left aligned 1 = Print sheet centred horizontally

6.51 HEADER

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0014 _H	0014 _H	0014 _H	0014 _H	0014 _H	0014 _H	0014 _H	0014 _H

This record is part of the *Page Settings Block* (→5.4). It specifies the page header string for the current worksheet. If this record is not present or completely empty (record size is 0), the sheet does not contain a page header.

Record HEADER for non-empty page header, BIFF2-BIFF8:

Offset	Size	Contents
0	var.	Page header string
		BIFF2-BIFF7: Non-empty byte string, 8-bit string length (→3.3)
		BIFF8: Non-empty Unicode string, 16-bit string length (→3.4)

The header string may contain special commands, for example placeholders for the page number, current date, or text formatting attributes. These fields are represented by single letters (exception: font name and size, see below) with a leading ampersand (“&”). If the ampersand is part of the regular header text, it will be duplicated (“&&”).

The page header is divided into 3 sections: the left, the centred, and the right section. Each section is introduced by a special command. All text and all commands following are part of the selected section. Each section starts with the text formatting specified in the default font (first FONT record in the file). Active formatting attributes from a previous section do not go into the next section.

The following table shows all available commands:

Command	Contents
&&	The “&” character itself
&L	Start of the left section
&C	Start of the centred section
&R	Start of the right section
&P	Current page number
&N	Page count
&D	Current date
&T	Current time
&A	Sheet name (BIFF5-BIFF8)
&F	File name without path
&Z	File path without file name (BIFF8X)
&G	Picture (BIFF8X)
&B	Bold on/off (BIFF2-BIFF4)
&I	Italic on/off (BIFF2-BIFF4)
&U	Underlining on/off
&E	Double underlining on/off (BIFF5-BIFF8)
&S	Strikeout on/off
&X	Superscript on/off (BIFF5-BIFF8)
&Y	Subscript on/off (BIFF5-BIFF8)
&"<fontname>"	Set new font <fontname>
&"<fontname>,<fontstyle>"	Set new font with specified style <fontstyle>. The style <fontstyle> is in most cases one of “Regular”, “Bold”, “Italic”, or “Bold Italic”. But this setting is dependent on the used font, it may differ (localised style names, or “Standard”, “Oblique”, ...). (BIFF5-BIFF8)
&<fontheight>	Set font height in points (<fontheight> is a decimal value). If this command is followed by a plain number to be printed in the header, it will be separated from the font height with a space character.

! In BIFF2 the commands differ in the localised versions of Excel.

6.52 HIDEOBJ

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	008D _H	008D _H	008D _H	008D _H	008D _H	008D _H	008D _H

This record specifies whether and how to show objects in the workbook.

Record HIDEOBJ, BIFF3-BIFF8:

Offset	Size	Contents
0	2	Viewing mode for objects: 0 = Show all objects 1 = Show placeholders 2 = Do not show objects

6.53 HLINK – Hyperlink

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	—	—	01B8 _H	01B8 _H

In Excel, every cell may contain a hyperlink. The HLINK record refers to one cell address or a cell range where all cells contain the same hyperlink. It is part of the *Hyperlink Table* in the *Worksheet Substream* (→5.13). Every hyperlink can contain a text mark and a description that is shown in the sheet instead of the real link. Text marks are appended behind a link, separated by the hash sign (“#”).

Examples for text marks: `www.example.org#table1` or `C:\example.xls#Sheet1!A1`.

Inside of this record strings are stored in several formats. Sometimes occurs the character count, otherwise the character array size (in 16-bit character arrays the character count is half of the array size). Furthermore some strings are zero-terminated, others not. They are stored either as 16-bit character arrays or as 8-bit character arrays, independent of the characters.

6.53.1 Common Record Contents

Each HLINK record starts with the same data items and continues with special data related to the current type of hyperlink. It starts with a cell range. Each cell of this range will contain the same hyperlink.

Record HLINK, BIFF8:

Offset	Size	Contents
0	8	Cell range address of all cells containing this hyperlink (→3.13.1)
8	16	GUID of StdLink: D0 _H C9 _H EA _H 79 _H F9 _H BA _H CE _H 11 _H 8C _H 82 _H 00 _H AA _H 00 _H 4B _H A9 _H 0B _H (79EAC9D0-BAF9-11CE-8C82-00AA004BA90B)
24	4	Unknown value: 00000002 _H
28	4	Option flags (see below)
[32]	4	(optional, see option flags) Character count of description text, including trailing zero word (d ₁)
[36]	2·d ₁	(optional, see option flags) Character array of description text, no Unicode string header, always 16-bit characters, zero-terminated
[var.]	4	(optional, see option flags) Character count of target frame, including trailing zero word (f ₁)
[var.]	2·f ₁	(optional, see option flags) Character array of target frame, no Unicode string header, always 16-bit characters, zero-terminated
var.	var.	Special data (→6.53.2 and following)
[var.]	4	(optional, see option flags) Character count of the text mark, including trailing zero word (t ₁)
[var.]	2·t ₁	(optional, see option flags) Character array of the text mark without “#” sign, no Unicode string header, always 16-bit characters, zero-terminated

The special data parts in the following are described with relative offsets (starting again by zero). The real offset inside of the record data (without header) is either 32 (without description) or 36+2·d₁ (with description).

• Option Flags

The option flags specify the following content of the record.

Bit	Mask	Contents
0	00000001 _H	0 = No link extant 1 = File link or URL
1	00000002 _H	0 = Relative file path 1 = Absolute path or URL
2 and 4	00000014 _H	0 = No description 1 (both bits) = Description
3	00000008 _H	0 = No text mark 1 = Text mark
7	00000080 _H	0 = No target frame 1 = Target frame
8	00000100 _H	0 = File link or URL 1 = UNC path (incl. server name)

6.53.2 Hyperlink containing a URL (Uniform Resource Locator)

These data fields occur for links which are not local files or files in the local network (for instance HTTP and FTP links and e-mail addresses). The lower 9 bits of the option flags field must be 0 . x00x . xx11₂ (x means optional, depending on hyperlink content). The GUID could be used to distinguish a URL from a file link.

Offset	Size	Contents
0	16	GUID of URL Moniker: E0 _H C9 _H EA _H 79 _H F9 _H BA _H CE _H 11 _H 8C _H 82 _H 00 _H AA _H 00 _H 4B _H A9 _H 0B _H (79EAC9E0-BAF9-11CE-8C82-00AA004BA90B)
16	4	Size of character array of the URL, including trailing zero word (<u>u.s.</u>). There are <u>u.s.</u> /2-1 characters in the following string.
20	<u>u.s.</u>	Character array of the URL, no Unicode string header, always 16-bit characters, zero-terminated

6.53.3 Hyperlink to a Local File

These data fields are for links to files on local drives. The path of the file can be complete with drive letter (absolute) or relative to the location of the workbook. The lower 9 bits of the option flags field must be $0 \cdot x00x \cdot xxx1_2$. The GUID could be used to distinguish a URL from a file link.

Offset	Size	Contents
0	16	GUID of File Moniker: 03 _H 03 _H 00 _H 00 _H 00 _H 00 _H 00 _H 00 _H C0 _H 00 _H 00 _H 00 _H 00 _H 00 _H 00 _H 46 _H (00000303-0000-0000-C000-000000000046)
16	2	Directory up-level count. Each leading “.” in the file link is deleted and increases this counter.
18	4	Character count of the shortened file path and name, including trailing zero byte (<u>s.l</u>)
22	<u>s.l</u>	Character array of the shortened file path and name in 8.3-DOS-format. This field can be filled with a long file name too. No Unicode string header, always 8-bit characters, zero-terminated.
22+ <u>s.l</u>	24	Unknown byte sequence: FF _H FF _H AD _H DE _H 00 _H
46+ <u>s.l</u>	4	Size of the following file link field including string length field and additional data field (<u>s.z</u>). If <u>s.z</u> is zero, nothing will follow (except a text mark).
[50+ <u>s.l</u>]	4	(optional) Size of character array of the extended file path and name (<u>x.l</u>). There are <u>x.l</u> /2 characters in the following string.
[54+ <u>s.l</u>]	2	(optional) Unknown byte sequence: 03 _H 00 _H
[56+ <u>s.l</u>]	<u>x.l</u>	(optional) Character array of the extended file path and name (<u>x.l</u>), no Unicode string header, always 16-bit characters, <i>not</i> zero-terminated

6.53.4 Hyperlink to a File with UNC (Universal Naming Convention) Path

These data fields are for UNC paths containing a server name (for instance “\\server\path\file.xls”). The lower 9 bits of the option flags field must be $1 \cdot x00x \cdot xx11_2$.

Offset	Size	Contents
0	4	Character count of the UNC, including trailing zero word (<u>f.l</u>)
4	2· <u>f.l</u>	Character array of the UNC, no Unicode string header, always 16-bit characters, zero-terminated.

6.53.5 Hyperlink to the Current Workbook

In this case only the text mark field is present (optional with description).

Example: The URL “#Sheet2!B1:C2” refers to the given range in the current workbook.

The lower 9 bits of the option flags field must be $0 \cdot x00x \cdot 1x00_2$.

6.54 HORIZONTALPAGEBREAKS

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
001B _H	001B _H	001B _H	001B _H	001B _H	001B _H	001B _H	001B _H

This record is part of the *Page Settings Block* (→5.4). It contains all horizontal manual page breaks.

Record HORIZONTALPAGEBREAKS, BIFF2-BIFF7:

Offset	Size	Contents
0	2	Number of following row indexes (<u>nm</u>)
2	2· <u>nm</u>	List of <u>nm</u> row indexes. Each row index specifies the first row after the page break.

Record HORIZONTALPAGEBREAKS, BIFF8:

Offset	Size	Contents												
0	2	Number of following row index structures (<u>nm</u>)												
2	6· <u>nm</u>	List of <u>nm</u> row index structures. Each row index structure contains: <table border="1" data-bbox="491 745 1487 896"> <thead> <tr> <th>Offset</th> <th>Size</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>2</td> <td>Index to first row below the page break</td> </tr> <tr> <td>2</td> <td>2</td> <td>Index to first column of this page break</td> </tr> <tr> <td>4</td> <td>2</td> <td>Index to last column of this page break</td> </tr> </tbody> </table>	Offset	Size	Contents	0	2	Index to first row below the page break	2	2	Index to first column of this page break	4	2	Index to last column of this page break
Offset	Size	Contents												
0	2	Index to first row below the page break												
2	2	Index to first column of this page break												
4	2	Index to last column of this page break												

The row indexes in the lists must be ordered ascending. If in BIFF8 a row contains several page breaks, they must be ordered ascending by start column index.

6.55 INDEX

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
000B _H	020B _H	020B _H	020B _H	020B _H	020B _H	020B _H	020B _H

This record stores the range of used rows and stream positions of several records of the current sheet. In particular the position of the first NAME record and XF record is stored (BIFF2-BIFF4) and the position of a specific record in each *Row Block* (the first ROW record in BIFF2-BIFF4, and the DBCELL record in BIFF5-BIFF8). This stream position array also contains stream offsets to empty *Row Blocks*, they will point to the next extant *Row Block*.

The number of entries \underline{nm} in this array can be calculated from the row range given in this record (\underline{rf} is the index to the first used row, \underline{rl} is the index to the first row of unused tail of sheet): $\underline{nm} = (\underline{rl} - \underline{rf} - 1) / 32 + 1$ (using integer division).

For details about *Row Blocks* in a worksheet see →5.7.

Record INDEX, BIFF2:

Offset	Size	Contents
0	4	Absolute stream position of the first NAME record (→6.66)
4	2	Index to first used row (\underline{rf} , 0-based)
6	2	Index to first row of unused tail of sheet (\underline{rl} , last used row + 1, 0-based)
8	4 \underline{nm}	Array of \underline{nm} absolute stream positions to first ROW record (→6.83) of each <i>Row Block</i>

Record INDEX, BIFF3-BIFF4:

Offset	Size	Contents
0	4	Absolute stream position of the first NAME record (→6.66). In BIFF4W workbook streams this is the offset relative to the position of the SHEETHDR record (→6.90) of the current sheet.
4	2	Index to first used row (\underline{rf} , 0-based)
6	2	Index to first row of unused tail of sheet (\underline{rl} , last used row + 1, 0-based)
8	4	Absolute stream position of the first XF record (→6.115). In BIFF4W workbook streams this is the offset relative to the position of the SHEETHDR record (→6.90) of the current sheet.
12	4 \underline{nm}	Array of \underline{nm} absolute stream positions to first ROW record (→6.83) of each <i>Row Block</i> . In BIFF4W workbook streams these offsets are relative to the position of the SHEETHDR record (→6.90) of the current sheet.

Record INDEX, BIFF5-BIFF7:

Offset	Size	Contents
0	4	Not used
4	2	Index to first used row (\underline{rf} , 0-based)
6	2	Index to first row of unused tail of sheet (\underline{rl} , last used row + 1, 0-based)
8	4	Absolute stream position of the DEFCOLWIDTH record (→6.29) of the current sheet. If this record does not exist, the offset points to the record at the position where the DEFCOLWIDTH record would occur.
12	4 \underline{nm}	Array of \underline{nm} absolute stream positions to the DBCELL record (→6.26) of each <i>Row Block</i>

Record INDEX, BIFF8:

Offset	Size	Contents
0	4	Not used
4	4	Index to first used row (<u>r.f.</u> , 0-based)
8	4	Index to first row of unused tail of sheet (<u>r.l.</u> , last used row + 1, 0-based)
12	4	Absolute stream position of the DEFCOLWIDTH record (→6.29) of the current sheet. If this record does not exist, the offset points to the record at the position where the DEFCOLWIDTH record would occur.
16	4· <u>nm</u>	Array of <u>nm</u> absolute stream positions to the DBCELL record (→6.26) of each <i>Row Block</i>

6.56 INTEGER

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0002 _H	—	—	—	—	—	—	—

This record represents a cell that contains an unsigned 16-bit integer value. If a value cannot be stored as a 16-bit integer, a NUMBER record (→6.68) must be written. From BIFF3 on this record is replaced by the RK record (→6.82).

Record INTEGER, BIFF2:

Offset	Size	Contents
0	2	Index to row
2	2	Index to column
4	3	Cell attributes (→3.12)
7	2	Unsigned 16-bit integer value

6.57 ITERATION

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0011 _H	0011 _H	0011 _H	0011 _H	0011 _H	0011 _H	0011 _H	0011 _H

This record is part of the *Calculation Settings Block* (→5.3). It stores if iterations are allowed while calculating recursive formulas.

Record ITERATION, BIFF2-BIFF8:

Offset	Size	Contents
0	2	0 = Iterations off; 1 = Iterations on

6.58 IXFE – Index to XF

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0044 _H	—	—	—	—	—	—	—

This record occurs in front of every cell record (for instance BLANK, INTEGER, NUMBER, LABEL, FORMULA) that references to an XF record (→6.115) with an index greater than 62. The XF index field of the cell record consists only of 6 bits. The maximum value 63 is used to indicate a preceding IXFE record containing the real XF index. See →3.12 for more details.

Record IXFE, BIFF2:

Offset	Size	Contents
0	2	Index to XF record (→6.115)

6.59 LABEL

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0004 _H	0204 _H	0204 _H	0204 _H	0204 _H	0204 _H	[0204 _H]	[0204 _H]

This record represents a cell that contains a string. In BIFF8 it is replaced by the LABELSST record (→6.61). Nevertheless Excel can import a LABEL record contained in a BIFF8 file.

Record LABEL, BIFF2:

Offset	Size	Contents
0	2	Index to row
2	2	Index to column
4	3	Cell attributes (→3.12)
7	var.	Byte string, 8-bit string length (→3.3)

Record LABEL, BIFF3-BIFF7:

Offset	Size	Contents
0	2	Index to row
2	2	Index to column
4	2	Index to XF record (→6.115)
6	var.	Byte string, 16-bit string length (→3.3)

Record LABEL, BIFF8:

Offset	Size	Contents
0	2	Index to row
2	2	Index to column
4	2	Index to XF record (→6.115)
6	var.	Unicode string, 16-bit string length (→3.4)

6.60 LABELRANGES

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	—	—	015F _H	015F _H

This record contains the addresses of all row and column label ranges in the current sheet.

Record LABELRANGES, BIFF8:

Offset	Size	Contents
0	var.	Cell range address list with all row label ranges (→3.13.2)
var.	var.	Cell range address list with all column label ranges (→3.13.2)

6.61 LABELSST

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	—	—	00FD _H	00FD _H

This record represents a cell that contains a string. It replaces the LABEL record (→6.59) and RSTRING record (→6.84) used in BIFF2-BIFF7. See →5.11 for more information about shared string tables.

Record LABELSST, BIFF8:

Offset	Size	Contents
0	2	Index to row
2	2	Index to column
4	2	Index to XF record (→6.115)
6	4	Index into SST record (→6.96)

6.62 LEFTMARGIN

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0026 _H	0026 _H	0026 _H	0026 _H	0026 _H	0026 _H	0026 _H	0026 _H

This record is part of the *Page Settings Block* (→5.4). It contains the left page margin of the current worksheet.

Record LEFTMARGIN, BIFF2-BIFF8:

Offset	Size	Contents
0	8	Left page margin in inches (IEEE 754 floating-point value, 64-bit double precision)

6.63 MERGEDCELLS

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	—	—	00E5 _H	00E5 _H

This record contains the addresses of merged cell ranges in the current sheet.

Record MERGEDCELLS, BIFF8:

Offset	Size	Contents
0	var.	Cell range address list with merged ranges (→3.13.2)

If the record size exceeds the limit, it is *not* continued with a CONTINUE record, but another self-contained MERGEDCELLS record is started. The limit of 8224 bytes per record results in a maximum number of 1027 merged ranges.

Example: A sheet contains 1040 merged cell ranges. The first MERGEDCELLS record contains a list of 1027 range addresses (the leading number of ranges is 1027 too). Following a second MERGEDCELLS record with the remaining 13 merged ranges.

6.64 MULBLANK – Multiple BLANK

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	00BE _H	00BE _H	00BE _H	00BE _H

This record represents a cell range of empty cells. All cells are located in the same row.

Record MULBLANK, BIFF5-BIFF8:

Offset	Size	Contents
0	2	Index to row
2	2	Index to first column (<u>f_C</u>)
4	2· <u>nc</u>	List of <u>nc</u> = <u>lc</u> - <u>fc</u> +1 16-bit indexes to XF records (→6.115)
4+2· <u>nc</u>	2	Index to last column (<u>lc</u>)

6.65 MULRK – Multiple RK

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	00BD _H	00BD _H	00BD _H	00BD _H

This record represents a cell range containing RK value cells. All cells are located in the same row.

Record MULRK, BIFF5-BIFF8:

Offset	Size	Contents									
0	2	Index to row									
2	2	Index to first column (<u>f_C</u>)									
4	6· <u>nc</u>	List of <u>nc</u> = <u>lc</u> - <u>fc</u> +1 XF/RK structures. Each XF/RK contains:									
		<table border="1"> <thead> <tr> <th>Offset</th> <th>Size</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>2</td> <td>Index to XF record (→6.115)</td> </tr> <tr> <td>2</td> <td>4</td> <td>RK value (→3.6)</td> </tr> </tbody> </table>	Offset	Size	Contents	0	2	Index to XF record (→6.115)	2	4	RK value (→3.6)
Offset	Size	Contents									
0	2	Index to XF record (→6.115)									
2	4	RK value (→3.6)									
4+6· <u>nc</u>	2	Index to last column (<u>lc</u>)									

6.66 NAME

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0018 _H	0218 _H	0218 _H	0218 _H	0018 _H	0018 _H	0018 _H	0018 _H

This record is part of a *Link Table* (→5.10). It contains the name and the token array of an internal defined name. Token arrays of defined names contain tokens with aberrant token classes (→4.2).

Record NAME, BIFF2:

Offset	Size	Contents									
0	1	Option flags: <table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>02_H</td> <td>1 = Function macro or command macro</td> </tr> <tr> <td>2</td> <td>04_H</td> <td>1 = Complex function (array formula or user defined)</td> </tr> </tbody> </table>	Bit	Mask	Contents	1	02 _H	1 = Function macro or command macro	2	04 _H	1 = Complex function (array formula or user defined)
Bit	Mask	Contents									
1	02 _H	1 = Function macro or command macro									
2	04 _H	1 = Complex function (array formula or user defined)									
1	1	If name is function macro or command macro (see option flags above): 01 _H = Function macro, 02 _H = Command macro									
2	1	Keyboard shortcut (only for command macro names)									
3	1	Length of the name (character count, <u>l</u> <u>n</u>)									
4	1	Size of the formula data (<u>s</u> <u>z</u>)									
5	<u>l</u> <u>n</u>	Character array of the name									
5+ <u>l</u> <u>n</u>	<u>s</u> <u>z</u>	Formula data (RPN token array without size field, →4)									
5+ <u>l</u> <u>n</u> + <u>s</u> <u>z</u>	1	Duplicate of the formula data size field (<u>s</u> <u>z</u>)									

Record NAME, BIFF3-BIFF4:

Offset	Size	Contents
0	2	Option flags, see below
2	1	Keyboard shortcut (only for command macro names, see below)
3	1	Length of the name (character count, <u>l</u> <u>n</u>)
4	2	Size of the formula data (<u>s</u> <u>z</u>)
6	<u>l</u> <u>n</u>	Character array of the name
6+ <u>l</u> <u>n</u>	<u>s</u> <u>z</u>	Formula data (RPN token array without size field, →4)

Record NAME, BIFF5/BIFF7:

Offset	Size	Contents
0	2	Option flags, see below
2	1	Keyboard shortcut (only for command macro names, see below)
3	1	Length of the name (character count, <u>l</u> <u>n</u>)
4	2	Size of the formula data (<u>s</u> <u>z</u>)
6	2	0 = Global name, otherwise index to EXTERNSHEET record (<i>one-based</i>)
8	2	0 = Global name, otherwise index to sheet (<i>one-based</i>)
10	1	Length of menu text (character count, <u>l</u> <u>m</u>)
11	1	Length of description text (character count, <u>l</u> <u>d</u>)
12	1	Length of help topic text (character count, <u>l</u> <u>h</u>)
13	1	Length of status bar text (character count, <u>l</u> <u>s</u>)
14	<u>l</u> <u>n</u>	Character array of the name
14+ <u>l</u> <u>n</u>	<u>s</u> <u>z</u>	Formula data (RPN token array without size field, →4)
14+ <u>l</u> <u>n</u> + <u>s</u> <u>z</u>	<u>l</u> <u>m</u>	Character array of menu text
var.	<u>l</u> <u>d</u>	Character array of description text
var.	<u>l</u> <u>h</u>	Character array of help topic text
var.	<u>l</u> <u>s</u>	Character array of status bar text

Record NAME, BIFF8:

Offset	Size	Contents
0	2	Option flags, see below
2	1	Keyboard shortcut (only for command macro names, see below)
3	1	Length of the name (character count, <u>l</u> <u>n</u>)
4	2	Size of the formula data (<u>s</u> <u>z</u>)
6	2	Not used
8	2	0 = Global name, otherwise index to sheet (<i>one-based</i>)
10	1	Length of menu text (character count, <u>l</u> <u>m</u>)
11	1	Length of description text (character count, <u>l</u> <u>d</u>)
12	1	Length of help topic text (character count, <u>l</u> <u>h</u>)
13	1	Length of status bar text (character count, <u>l</u> <u>s</u>)
14	var.	Name (Unicode string without length field, →3.4)
var.	<u>s</u> <u>z</u>	Formula data (RPN token array without size field, →4)
[var.]	var.	(optional, only if <u>l</u> <u>m</u> > 0) Menu text (Unicode string without length field, →3.4)
[var.]	var.	(optional, only if <u>l</u> <u>d</u> > 0) Description text (Unicode string without length field, →3.4)
[var.]	var.	(optional, only if <u>l</u> <u>h</u> > 0) Help topic text (Unicode string without length field, →3.4)
[var.]	var.	(optional, only if <u>l</u> <u>s</u> > 0) Status bar text (Unicode string without length field, →3.4)

• Option Flags

Bit	Mask	Flag name	Contents
0	0001 _H	<u>h</u> <u>i</u> <u>d</u> <u>d</u> <u>e</u> <u>n</u>	0 = Visible 1 = Hidden
1	0002 _H	<u>f</u> <u>u</u> <u>n</u> <u>c</u>	0 = Command macro 1 = Function macro
2	0004 _H	<u>v</u> <u>b</u> <u>a</u> <u>s</u> <u>i</u> <u>c</u>	0 = Sheet macro 1 = VisualBasic macro
3	0008 _H	<u>m</u> <u>a</u> <u>c</u> <u>r</u> <u>o</u>	0 = Standard name 1 = Macro name (see below)
4	0010 _H	<u>c</u> <u>o</u> <u>m</u> <u>p</u> <u>l</u> <u>e</u> <u>x</u>	0 = Simple formula 1 = Complex formula (array formula or user defined)
5	0020 _H	<u>b</u> <u>u</u> <u>i</u> <u>l</u> <u>t</u> <u>i</u> <u>n</u>	0 = User-defined name 1 = Built-in name (see below)
11-6	0FC0 _H	<u>f</u> <u>u</u> <u>n</u> <u>c</u> <u>g</u> <u>r</u> <u>o</u> <u>u</u> <u>p</u>	Function group (BIFF4-BIFF8), only if <u>m</u> <u>a</u> <u>c</u> <u>r</u> <u>o</u> = 1, must be >0 then 1 = Financial 8 = Logical 2 = Date & Time 9 = Information 3 = Math & Trig 10 = Commands 4 = Statistical 11 = Customizing 5 = Lookup & Reference 12 = Macro Control 6 = Database 13 = DDE/External 7 = Text 14 = User Defined
12	1000 _H	<u>b</u> <u>i</u> <u>n</u> <u>a</u> <u>r</u> <u>y</u>	0 = Formula definition 1 = Binary data (BIFF5-BIFF8)

• Macro Names

If the macro flag in the option flags field is set to 1 (see above), the defined name specifies a macro function or procedure. The type of the macro is specified by the flags func, vbasic, and funcgroup (all these flags must be 0, if the defined name is not a macro).

Command macros (macro = 1, func = 0) are shown in the “Select Macro” dialog of Excel. Function macros (macro = 1, func = 1) are shown in the “Insert Function” dialog of Excel. They will be inserted into the function group specified by funcgroup. If the flag vbasic is set, the name refers to a macro in the VisualBasic project, otherwise to a sheet macro.

• Built-In Names

In BIFF2, the built-in names are written with their full name and differ in localised versions of Excel. From BIFF3 on only an index to a built-in names is stored. If the built-in flag of the option flags field is set (see above), the name string contains only one character with the following meaning:

Character	Built-in name	Character	Built-in name
<00 _H >	Consolidate_Area	<07 _H >	Pint_Titles
<01 _H >	Auto_Open	<08 _H >	Recorder
<02 _H >	Auto_Close	<09 _H >	Data_Form
<03 _H >	Extract	<0A _H >	Auto_Activate (BIFF5-BIFF8)
<04 _H >	Database	<0B _H >	Auto_Deactivate (BIFF5-BIFF8)
<05 _H >	Criteria	<0C _H >	Sheet_Title (BIFF5-BIFF8)
<06 _H >	Print_Area	<0D _H >	_FilterDatabase (BIFF8)

- ! In BIFF5/BIFF7 the name “_FilterDatabase” (used to define filtered cell ranges) occurs as full name, not with its index. The built-in flag of the option flags field is not set for this name.

6.67 NOTE

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
001C _H	001C _H	001C _H	001C _H	001C _H	001C _H	001C _H	001C _H

This record represents a cell annotation.

Record NOTE, BIFF2-BIFF7:

Offset	Size	Contents
0	2	Index to row
2	2	Index to column
4	2	Total length of the annotation string (character count, <u>l_n</u>)
6	var.	Character array (8-bit characters). This character array will contain at most 2048 characters. If an annotation is longer than 2048 characters, the remaining string will be written in one or more following NOTE records (see below). Nevertheless the string length field <u>l_n</u> in this record contains the <i>total length</i> of the annotation text.

Record NOTE for continuation of long notes, BIFF2-BIFF7:

Offset	Size	Contents
0	2	always FFFF _H
2	2	Not used
4	2	Length of <i>this</i> part of the annotation string (character count, 1...2048, <u>l_n</u>)
6	var.	Character array containing <u>l_n</u> characters (8-bit characters)

Example: An annotation contains 5000 characters. The string will be split into 3 parts: the first two NOTE records contain 2048 characters each, and the third NOTE record contains the remaining 904 characters. The first NOTE record contains the total length of the string (5000), the second and third record contain the local length (2048 respectively 904).

Record NOTE, BIFF8:

2do

6.68 NUMBER

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0003 _H	0203 _H	0203 _H	0203 _H	0203 _H	0203 _H	0203 _H	0203 _H

This record represents a cell that contains a floating-point value.

Record NUMBER, BIFF2:

Offset	Size	Contents
0	2	Index to row
2	2	Index to column
4	3	Cell attributes (→3.12)
7	8	IEEE 754 floating-point value (64-bit double precision) ³⁰

Record NUMBER, BIFF3-BIFF8:

Offset	Size	Contents
0	2	Index to row
2	2	Index to column
4	2	Index to XF record (→6.115)
6	8	IEEE 754 floating-point value (64-bit double precision) ³⁰

6.69 OBJECTPROTECT

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	0063 _H	0063 _H	0063 _H	0063 _H	0063 _H	0063 _H	0063 _H

This record is part of the worksheet/workbook protection (→5.18). It determines whether the objects of the current sheet are protected. Object protection is not active, if this record is omitted.

Record OBJECTPROTECT, BIFF3-BIFF8:

Offset	Size	Contents
0	2	0 = Objects not protected; 1 = Objects protected

³⁰ For details about the internal structure of floating-point values see http://en.wikipedia.org/wiki/IEEE_floating-point_standard.

6.70 PALETTE

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	0092 _H	0092 _H	0092 _H	0092 _H	0092 _H	0092 _H	0092 _H

This record contains the definition of all user-defined colours available for cell and object formatting.

Record PALETTE, BIFF3-BIFF8:

Offset	Size	Contents
0	2	Number of following colours (<u>nm</u>). Contains 16 in BIFF3-BIFF4 and 56 in BIFF5-BIFF8.
2	4 <u>nm</u>	List of <u>nm</u> RGB colours (→3.5)

The following table shows how colour indexes are used in other records:

Colour index	Resulting colour or internal list index
00 _H	Built-in Black (R = 00 _H , G = 00 _H , B = 00 _H)
01 _H	Built-in White (R = FF _H , G = FF _H , B = FF _H)
02 _H	Built-in Red (R = FF _H , G = 00 _H , B = 00 _H)
03 _H	Built-in Green (R = 00 _H , G = FF _H , B = 00 _H)
04 _H	Built-in Blue (R = 00 _H , G = 00 _H , B = FF _H)
05 _H	Built-in Yellow (R = FF _H , G = FF _H , B = 00 _H)
06 _H	Built-in Magenta (R = FF _H , G = 00 _H , B = FF _H)
07 _H	Built-in Cyan (R = 00 _H , G = FF _H , B = FF _H)
08 _H	First user-defined colour from the PALETTE record (entry 0 from record colour list)
⋮	⋮
17 _H (BIFF3-BIFF4) 3F _H (BIFF5-BIFF8)	Last user-defined colour from the PALETTE record (entry 15 or 55 from record colour list)
18 _H (BIFF3-BIFF4) 40 _H (BIFF5-BIFF8)	System window text colour for border lines (used in records XF →6.115, CF →6.16, and WINDOW2 (BIFF8 only), →6.109)
19 _H (BIFF3-BIFF4) 41 _H (BIFF5-BIFF8)	System window background colour for pattern background (used in records XF, and CF)
43 _H	System face colour (dialogue background colour)
4D _H	System window text colour for chart border lines
4E _H	System window background colour for chart areas
4F _H	Automatic colour for chart border lines (seems to be always Black)
50 _H	System ToolTip background colour (used in note objects)
51 _H	System ToolTip text colour (used in note objects)
7FFF _H	System window text colour for fonts (used in records FONT →6.43, EFONT →6.35, and CF)

6.71 PANE

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0041 _H	0041 _H	0041 _H	0041 _H	0041 _H	0041 _H	0041 _H	0041 _H

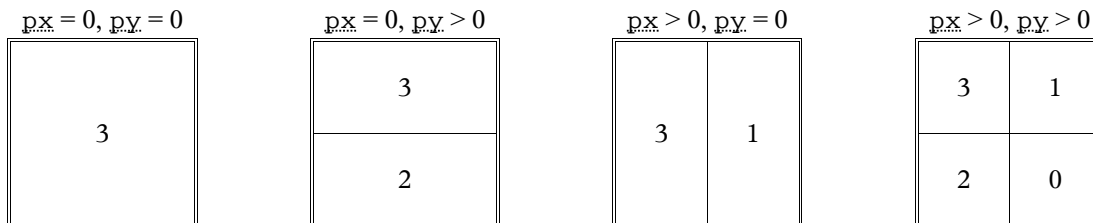
This record stores the position of window panes. It is part of the *Sheet View Settings Block* (→5.5). If the sheet does not contain any splits, this record will not occur.

A sheet can be split in two different ways, with unfrozen panes or with frozen panes. A flag in the WINDOW2 record (→6.109) specifies, if the panes are frozen, which affects the contents of this record.

Record PANE, BIFF2-BIFF8:

Offset	Size	Contents
0	2	Position of the vertical split (p_x , 0 = No vertical split): <i>Unfrozen pane</i> : Width of the left pane(s) (in twips = $\frac{1}{20}$ of a point) <i>Frozen pane</i> : Number of visible columns in left pane(s)
2	2	Position of the horizontal split (p_y , 0 = No horizontal split): <i>Unfrozen pane</i> : Height of the top pane(s) (in twips = $\frac{1}{20}$ of a point) <i>Frozen pane</i> : Number of visible rows in top pane(s)
4	2	Index to first visible row in bottom pane(s)
6	2	Index to first visible column in right pane(s)
8	1	Identifier of pane with active cell cursor (see below)
[9]	1	Not used (BIFF5-BIFF8 only, not written in BIFF2-BIFF4)

If the panes are frozen, pane 0 is always active, regardless of the cursor position. The correct identifiers for all possible combinations of visible panes are shown in the following pictures.



6.72 PASSWORD

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0013 _H	0013 _H	0013 _H	0013 _H	0013 _H	0013 _H	0013 _H	0013 _H

This record is part of the worksheet/workbook protection (→5.18). It stores a 16-bit hash value, calculated from the worksheet or workbook protection password.

Record PASSWORD, BIFF2-BIFF8:

Offset	Size	Contents
0	2	16-bit hash value of the password (→5.18.4)

6.73 PHONETIC

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	—	—	00EF _H	00EF _H

This record contains default settings for the “Asian Phonetic Settings” dialog and the addresses of all cells which show Asian phonetic text.

Record PHONETIC, BIFF8:

Offset	Size	Contents												
0	2	Index to FONT record (→6.43) used for Asian phonetic text of new cells												
2	2	Additional settings used for Asian phonetic text of new cells:												
		<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>1-0</td> <td>0003_H</td> <td>Type of Japanese phonetic text: 00₂ = Katakana (narrow) 10₂ = Hiragana 01₂ = Katakana (wide)</td> </tr> <tr> <td>3-2</td> <td>000C_H</td> <td>Alignment of all portions of the Asian phonetic text: 00₂ = Not specified (Japanese only) 10₂ = Centered 01₂ = Left (Top for vertical text) 11₂ = Distributed</td> </tr> <tr> <td>5-4</td> <td>0030_H</td> <td>11₂ (always set)</td> </tr> </tbody> </table>	Bit	Mask	Contents	1-0	0003 _H	Type of Japanese phonetic text: 00 ₂ = Katakana (narrow) 10 ₂ = Hiragana 01 ₂ = Katakana (wide)	3-2	000C _H	Alignment of all portions of the Asian phonetic text: 00 ₂ = Not specified (Japanese only) 10 ₂ = Centered 01 ₂ = Left (Top for vertical text) 11 ₂ = Distributed	5-4	0030 _H	11 ₂ (always set)
Bit	Mask	Contents												
1-0	0003 _H	Type of Japanese phonetic text: 00 ₂ = Katakana (narrow) 10 ₂ = Hiragana 01 ₂ = Katakana (wide)												
3-2	000C _H	Alignment of all portions of the Asian phonetic text: 00 ₂ = Not specified (Japanese only) 10 ₂ = Centered 01 ₂ = Left (Top for vertical text) 11 ₂ = Distributed												
5-4	0030 _H	11 ₂ (always set)												
4	var.	Cell range address list with all cells with visible Asian phonetic text (→3.13.2)												

6.74 PRECISION

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
000E _H	000E _H	000E _H	000E _H	000E _H	000E _H	000E _H	000E _H

This record stores if formulas use the real cell values for calculation or the values displayed on the screen. In BIFF2-BIFF4 this record is part of the *Calculation Settings Block* (→5.3). In BIFF5-BIFF8 it is stored in the *Workbook Globals Substream*.

Record PRECISION, BIFF2-BIFF8:

Offset	Size	Contents
0	2	0 = Use displayed values; 1 = Use real cell values

6.75 PRINTGRIDLINES

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
002B _H	002B _H	002B _H	002B _H	002B _H	002B _H	002B _H	002B _H

This record stores if sheet grid lines will be printed.

Record PRINTGRIDLINES, BIFF2-BIFF8:

Offset	Size	Contents
0	2	0 = Do not print sheet grid lines; 1 = Print sheet grid lines

6.76 PRINTHEADERS

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
002A _H	002A _H	002A _H	002A _H	002A _H	002A _H	002A _H	002A _H

This record stores if the row and column headers (the areas with row numbers and column letters) will be printed.

Record PRINTHEADERS, BIFF2-BIFF8:

Offset	Size	Contents
0	2	0 = Do not print row/column headers; 1 = Print row/column headers

6.77 PROTECT

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0012 _H	0012 _H	0012 _H	0012 _H	0012 _H	0012 _H	0012 _H	0012 _H

This record is part of the worksheet/workbook protection (→5.18). It specifies whether a worksheet or a workbook is protected against modification. Protection is not active, if this record is omitted.

Record PROTECT, BIFF2-BIFF8:

Offset	Size	Contents
0	2	0 = Not protected; 1 = Protected

6.78 QUICKTIP

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	—	—	0800 _H	0800 _H

This record contains the cell range and text for a tool tip. It occurs in conjunction with the HLINK record for hyperlinks (→6.53) in the *Hyperlink Table* (→5.13). This record is only available in Excel 9.0 (Excel 2000) and later.

Record QUICKTIP, BIFF8:

Offset	Size	Contents
0	2	0800 _H (repeated record identifier)
2	8	Cell range address of all cells containing the tool tip (→3.13.1)
10	var.	Character array of the tool tip, no Unicode string header, always 16-bit characters, zero-terminated

6.79 RANGEPROTECTION

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	—	—	—	0868 _H

This record is part of the worksheet/workbook protection (→5.18). It stores information about special protected ranges in a protected sheet.

Record RANGEPROTECTION, BIFF8X:

Offset	Size	Contents
0	2	0868 _H (repeated record identifier)
2		2do

6.80 REFMODE

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
000F _H	000F _H	000F _H	000F _H	000F _H	000F _H	000F _H	000F _H

This record is part of the *Calculation Settings Block* (→5.3). It stores which method is used to show cell addresses in formulas.

- The “RC” mode uses numeric indexes for rows and columns, for example “R(1)C(-1)”, or “R1C1:R2C2”.
- The “A1” mode uses characters for columns and numbers for rows, for example “B1”, or “\$A\$1:\$B\$2”.

Record REFMODE, BIFF2-BIFF8:

Offset	Size	Contents
0	2	0 = RC mode; 1 = A1 mode

6.81 RIGHTMARGIN

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0027 _H	0027 _H	0027 _H	0027 _H	0027 _H	0027 _H	0027 _H	0027 _H

This record is part of the *Page Settings Block* (→5.4). It contains the right page margin of the current worksheet.

Record RIGHTMARGIN, BIFF2-BIFF8:

Offset	Size	Contents
0	8	Right page margin in inches (IEEE 754 floating-point value, 64-bit double precision)

6.82 RK

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	027E _H	027E _H	027E _H	027E _H	027E _H	027E _H	027E _H

This record represents a cell that contains an RK value (encoded integer or floating-point value). If a floating-point value cannot be encoded to an RK value, a NUMBER record (→6.68) will be written. This record replaces the record INTEGER (→6.56) written in BIFF2.

Record RK, BIFF3-BIFF8:

Offset	Size	Contents
0	2	Index to row
2	2	Index to column
4	2	Index to XF record (→6.115)
6	4	RK value (→3.6)

6.83 ROW

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0008 _H	0208 _H	0208 _H	0208 _H	0208 _H	0208 _H	0208 _H	0208 _H

This record contains the properties of a single row in a sheet. Rows and cells in a sheet are divided into blocks of 32 rows. For details about *Row Blocks* see →5.7.

Record ROW, BIFF2:

Offset	Size	Contents									
0	2	Index of this row									
2	2	Index to column of the first cell which is described by a cell record									
4	2	Index to column of the last cell which is described by a cell record, increased by 1									
6	2	<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>14-0</td> <td>7FFF_H</td> <td>Height of the row, in twips = $\frac{1}{20}$ of a point</td> </tr> <tr> <td>15</td> <td>8000_H</td> <td>0 = Row has custom height; 1 = Row has default height</td> </tr> </tbody> </table>	Bit	Mask	Contents	14-0	7FFF _H	Height of the row, in twips = $\frac{1}{20}$ of a point	15	8000 _H	0 = Row has custom height; 1 = Row has default height
Bit	Mask	Contents									
14-0	7FFF _H	Height of the row, in twips = $\frac{1}{20}$ of a point									
15	8000 _H	0 = Row has custom height; 1 = Row has default height									
8	2	Not used									
10	1	0 = No defaults written; 1 = Default row attribute field and XF index occur below (<u>f.l</u>)									
11	2	Relative offset to calculate stream position of the first cell record for this row (→5.7.1)									
[13]	3	(written only if <u>f.l</u> = 1) Default row attributes (→3.12)									
[16]	2	(written only if <u>f.l</u> = 1) Index to XF record (→6.115)									

Record ROW, BIFF3-BIFF8:

Offset	Size	Contents																														
0	2	Index of this row																														
2	2	Index to column of the first cell which is described by a cell record																														
4	2	Index to column of the last cell which is described by a cell record, increased by 1																														
6	2	<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>14-0</td> <td>7FFF_H</td> <td>Height of the row, in twips = $\frac{1}{20}$ of a point</td> </tr> <tr> <td>15</td> <td>8000_H</td> <td>0 = Row has custom height; 1 = Row has default height</td> </tr> </tbody> </table>	Bit	Mask	Contents	14-0	7FFF _H	Height of the row, in twips = $\frac{1}{20}$ of a point	15	8000 _H	0 = Row has custom height; 1 = Row has default height																					
Bit	Mask	Contents																														
14-0	7FFF _H	Height of the row, in twips = $\frac{1}{20}$ of a point																														
15	8000 _H	0 = Row has custom height; 1 = Row has default height																														
8	2	Not used																														
10	2	In BIFF3-BIFF4 this field contains a relative offset to calculate stream position of the first cell record for this row (→5.7.1). In BIFF5-BIFF8 this field is not used anymore, but the DBCELL record (→6.26) instead.																														
12	4	Option flags and default row formatting: <table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>2-0</td> <td>00000007_H</td> <td>Outline level of the row</td> </tr> <tr> <td>4</td> <td>00000010_H</td> <td>1 = Outline group starts or ends here (depending on where the outline buttons are located, see WSBOOL record, →6.113), <i>and</i> is collapsed</td> </tr> <tr> <td>5</td> <td>00000020_H</td> <td>1 = Row is hidden (manually, or by a filter or outline group)</td> </tr> <tr> <td>6</td> <td>00000040_H</td> <td>1 = Row height and default font height do not match</td> </tr> <tr> <td>7</td> <td>00000080_H</td> <td>1 = Row has explicit default format (<u>f.l</u>)</td> </tr> <tr> <td>8</td> <td>00000100_H</td> <td>Always 1</td> </tr> <tr> <td>27-16</td> <td>0FFF0000_H</td> <td>If <u>f.l</u> = 1: Index to default XF record (→6.115)</td> </tr> <tr> <td>28</td> <td>10000000_H</td> <td>1 = Additional space above the row. This flag is set, if the upper border of at least one cell in this row or if the lower border of at least one cell in the row above is formatted with a thick line style. Thin and medium line styles are not taken into account.</td> </tr> <tr> <td>29</td> <td>20000000_H</td> <td>1 = Additional space below the row. This flag is set, if the lower border of at least one cell in this row or if the upper border of at least one cell in the row below is formatted with a medium or thick line style. Thin line styles are not taken into account.</td> </tr> </tbody> </table>	Bit	Mask	Contents	2-0	00000007 _H	Outline level of the row	4	00000010 _H	1 = Outline group starts or ends here (depending on where the outline buttons are located, see WSBOOL record, →6.113), <i>and</i> is collapsed	5	00000020 _H	1 = Row is hidden (manually, or by a filter or outline group)	6	00000040 _H	1 = Row height and default font height do not match	7	00000080 _H	1 = Row has explicit default format (<u>f.l</u>)	8	00000100 _H	Always 1	27-16	0FFF0000 _H	If <u>f.l</u> = 1: Index to default XF record (→6.115)	28	10000000 _H	1 = Additional space above the row. This flag is set, if the upper border of at least one cell in this row or if the lower border of at least one cell in the row above is formatted with a thick line style. Thin and medium line styles are not taken into account.	29	20000000 _H	1 = Additional space below the row. This flag is set, if the lower border of at least one cell in this row or if the upper border of at least one cell in the row below is formatted with a medium or thick line style. Thin line styles are not taken into account.
Bit	Mask	Contents																														
2-0	00000007 _H	Outline level of the row																														
4	00000010 _H	1 = Outline group starts or ends here (depending on where the outline buttons are located, see WSBOOL record, →6.113), <i>and</i> is collapsed																														
5	00000020 _H	1 = Row is hidden (manually, or by a filter or outline group)																														
6	00000040 _H	1 = Row height and default font height do not match																														
7	00000080 _H	1 = Row has explicit default format (<u>f.l</u>)																														
8	00000100 _H	Always 1																														
27-16	0FFF0000 _H	If <u>f.l</u> = 1: Index to default XF record (→6.115)																														
28	10000000 _H	1 = Additional space above the row. This flag is set, if the upper border of at least one cell in this row or if the lower border of at least one cell in the row above is formatted with a thick line style. Thin and medium line styles are not taken into account.																														
29	20000000 _H	1 = Additional space below the row. This flag is set, if the lower border of at least one cell in this row or if the upper border of at least one cell in the row below is formatted with a medium or thick line style. Thin line styles are not taken into account.																														

6.84 RSTRING

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	00D6 _H	00D6 _H	[00D6 _H]	[00D6 _H]

This record stores a formatted text cell (Rich-Text). In BIFF8 it is replaced by the LABELSST record (→6.96). Nevertheless Excel uses this record if it copies formatted text cells to the clipboard.

Record RSTRING, BIFF5/BIFF7:

Offset	Size	Contents
0	2	Index to row
2	2	Index to column
4	2	Index to XF record (→6.115)
6	<u>sz</u>	Byte string, 16-bit string length (→3.3)
6+ <u>sz</u>	1	Number of Rich-Text formatting runs (<u>rt</u>)
7+ <u>sz</u>	2· <u>rt</u>	List of <u>rt</u> formatting runs (→3.2)

Record RSTRING, BIFF8:

Offset	Size	Contents
0	2	Index to row
2	2	Index to column
4	2	Index to XF record (→6.115)
6	<u>sz</u>	Unformatted Unicode string, 16-bit string length (→3.4)
6+ <u>sz</u>	2	Number of Rich-Text formatting runs (<u>rt</u>)
8+ <u>sz</u>	4· <u>rt</u>	List of <u>rt</u> formatting runs (→3.2)

6.85 SAVERECALC

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	005F _H	005F _H	005F _H	005F _H	005F _H	005F _H	005F _H

This record is part of the *Calculation Settings Block* (→5.3). It contains the “Recalculate before save” option in Excel's calculation settings dialogue.

Record SAVERECALC, BIFF3-BIFF8:

Offset	Size	Contents
0	2	0 = Do not recalculate; 1 = Recalculate before saving the document

6.86 SCENPROTECT

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	00DD _H	00DD _H	00DD _H	00DD _H

This record is part of the worksheet/workbook protection (→5.18). It determines whether the scenarios of the current sheet are protected. Scenario protection is not active, if this record is omitted.

Record SCENPROTECT, BIFF5-BIFF8:

Offset	Size	Contents
0	2	0 = Scenarios not protected; 1 = Scenarios protected

6.87 SCL

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	00A0 _H	00A0 _H	00A0 _H	00A0 _H	00A0 _H	00A0 _H

This record stores the magnification of the active view of the current worksheet. In BIFF8 this can be either the normal view or the page break preview. This is determined in the WINDOW2 record (→6.109). The SCL record is part of the *Sheet View Settings Block* (→5.5).

Record SCL, BIFF4-BIFF8:

Offset	Size	Contents
0	2	Numerator of the view magnification fraction (<u>num</u>)
2	2	Denominator of the view magnification fraction (<u>den</u>)

The magnification is stored as reduced fraction. The magnification results from num/den.

6.88 SELECTION

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
001D _H	001D _H	001D _H	001D _H	001D _H	001D _H	001D _H	001D _H

This record contains the addresses of all selected cell ranges and the position of the active cell for a pane in the current sheet. It is part of the *Sheet View Settings Block* (→5.5). There is one SELECTION record for each pane in the sheet.

Record SELECTION, BIFF2-BIFF8:

Offset	Size	Contents
0	1	Pane identifier (see PANE record, →6.71)
1	2	Index to row of the active cell
3	2	Index to column of the active cell
5	2	Index into the following cell range list to the entry that contains the active cell
7	var.	Cell range address list containing all selected cell ranges (→3.13.2). Column indexes are always 8-bit values, also in BIFF8.

6.89 SETUP

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	00A1 _H	00A1 _H	00A1 _H	00A1 _H	00A1 _H	00A1 _H

This record is part of the *Page Settings Block* (→5.4). It stores the page format settings of the current sheet. The pages may be scaled in percent or by using an absolute number of pages. This setting is located in the WSBOOL record (→6.113). If pages are scaled in percent, the scaling factor in this record is used, otherwise the “Fit to pages” values. One of the “Fit to pages” values may be 0. In this case the sheet is scaled to fit only to the other value.

• Record Contents

Record SETUP, BIFF4:

Offset	Size	Contents															
0	2	Paper size (see below)															
2	2	Scaling factor in percent															
4	2	Start page number															
6	2	Fit worksheet width to this number of pages (0 = use as many as needed)															
8	2	Fit worksheet height to this number of pages (0 = use as many as needed)															
10	2	Option flags:															
		<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0001_H</td> <td>0 = Print pages in columns 1 = Print pages in rows</td> </tr> <tr> <td>1</td> <td>0002_H</td> <td>0 = Landscape 1 = Portrait</td> </tr> <tr> <td>2</td> <td>0004_H</td> <td>1 = Paper size, scaling factor, and paper orientation (portrait/landscape) are not initialised</td> </tr> <tr> <td>3</td> <td>0008_H</td> <td>0 = Print coloured 1 = Print black and white</td> </tr> </tbody> </table>	Bit	Mask	Contents	0	0001 _H	0 = Print pages in columns 1 = Print pages in rows	1	0002 _H	0 = Landscape 1 = Portrait	2	0004 _H	1 = Paper size, scaling factor, and paper orientation (portrait/landscape) are not initialised	3	0008 _H	0 = Print coloured 1 = Print black and white
Bit	Mask	Contents															
0	0001 _H	0 = Print pages in columns 1 = Print pages in rows															
1	0002 _H	0 = Landscape 1 = Portrait															
2	0004 _H	1 = Paper size, scaling factor, and paper orientation (portrait/landscape) are not initialised															
3	0008 _H	0 = Print coloured 1 = Print black and white															

Record SETUP, BIFF5-BIFF8:

Offset	Size	Contents																																				
0	10	Equal to BIFF4, see above																																				
10	2	Option flags:																																				
		<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0001_H</td> <td>0 = Print pages in columns 1 = Print pages in rows</td> </tr> <tr> <td>1</td> <td>0002_H</td> <td>0 = Landscape 1 = Portrait</td> </tr> <tr> <td>2</td> <td>0004_H</td> <td>1 = Paper size, scaling factor, paper orientation (portrait/landscape), print resolution and number of copies are not initialised</td> </tr> <tr> <td>3</td> <td>0008_H</td> <td>0 = Print coloured 1 = Print black and white</td> </tr> <tr> <td>4</td> <td>0010_H</td> <td>0 = Default print quality 1 = Draft quality</td> </tr> <tr> <td>5</td> <td>0020_H</td> <td>0 = Do not print cell notes 1 = Print cell notes</td> </tr> <tr> <td>6</td> <td>0040_H</td> <td>0 = Paper orientation setting is valid 1 = Paper orientation setting not initialised</td> </tr> <tr> <td>7</td> <td>0080_H</td> <td>0 = Automatic page numbers 1 = Use start page number</td> </tr> <tr> <td colspan="3">The following flags are valid for BIFF8 only:</td> </tr> <tr> <td>9</td> <td>0200_H</td> <td>0 = Print notes as displayed 1 = Print notes at end of sheet</td> </tr> <tr> <td>11-10</td> <td>0C00_H</td> <td>00₂ = Print errors as displayed 01₂ = Do not print errors 10₂ = Print errors as “-” 11₂ = Print errors as “#N/A!”</td> </tr> </tbody> </table>	Bit	Mask	Contents	0	0001 _H	0 = Print pages in columns 1 = Print pages in rows	1	0002 _H	0 = Landscape 1 = Portrait	2	0004 _H	1 = Paper size, scaling factor, paper orientation (portrait/landscape), print resolution and number of copies are not initialised	3	0008 _H	0 = Print coloured 1 = Print black and white	4	0010 _H	0 = Default print quality 1 = Draft quality	5	0020 _H	0 = Do not print cell notes 1 = Print cell notes	6	0040 _H	0 = Paper orientation setting is valid 1 = Paper orientation setting not initialised	7	0080 _H	0 = Automatic page numbers 1 = Use start page number	The following flags are valid for BIFF8 only:			9	0200 _H	0 = Print notes as displayed 1 = Print notes at end of sheet	11-10	0C00 _H	00 ₂ = Print errors as displayed 01 ₂ = Do not print errors 10 ₂ = Print errors as “-” 11 ₂ = Print errors as “#N/A!”
Bit	Mask	Contents																																				
0	0001 _H	0 = Print pages in columns 1 = Print pages in rows																																				
1	0002 _H	0 = Landscape 1 = Portrait																																				
2	0004 _H	1 = Paper size, scaling factor, paper orientation (portrait/landscape), print resolution and number of copies are not initialised																																				
3	0008 _H	0 = Print coloured 1 = Print black and white																																				
4	0010 _H	0 = Default print quality 1 = Draft quality																																				
5	0020 _H	0 = Do not print cell notes 1 = Print cell notes																																				
6	0040 _H	0 = Paper orientation setting is valid 1 = Paper orientation setting not initialised																																				
7	0080 _H	0 = Automatic page numbers 1 = Use start page number																																				
The following flags are valid for BIFF8 only:																																						
9	0200 _H	0 = Print notes as displayed 1 = Print notes at end of sheet																																				
11-10	0C00 _H	00 ₂ = Print errors as displayed 01 ₂ = Do not print errors 10 ₂ = Print errors as “-” 11 ₂ = Print errors as “#N/A!”																																				
12	2	Print resolution in dpi																																				
14	2	Vertical print resolution in dpi																																				
16	8	Header margin (IEEE 754 floating-point value, 64-bit double precision)																																				
24	8	Footer margin (IEEE 754 floating-point value, 64-bit double precision)																																				
32	2	Number of copies to print																																				

• Paper Size Table

Index	Paper type	Paper size	Index	Paper type	Paper size
0	Undefined		48	Undefined	
1	Letter	8 ¹ / ₂ " × 11"	49	Undefined	
2	Letter small	8 ¹ / ₂ " × 11"	50	Letter Extra	9 ¹ / ₂ " × 12"
3	Tabloid	11" × 17"	51	Legal Extra	9 ¹ / ₂ " × 15"
4	Ledger	17" × 11"	52	Tabloid Extra	11 ¹¹ / ₁₆ " × 18"
5	Legal	8 ¹ / ₂ " × 14"	53	A4 Extra	235mm × 322mm
6	Statement	5 ¹ / ₂ " × 8 ¹ / ₂ "	54	Letter Transverse	8 ¹ / ₂ " × 11"
7	Executive	7 ¹ / ₄ " × 10 ¹ / ₂ "	55	A4 Transverse	210mm × 297mm
8	A3	297mm × 420mm	56	Letter Extra Transv.	9 ¹ / ₂ " × 12"
9	A4	210mm × 297mm	57	Super A/A4	227mm × 356mm
10	A4 small	210mm × 297mm	58	Super B/A3	305mm × 487mm
11	A5	148mm × 210mm	59	Letter Plus	8 ¹ / ₂ " × 12 ¹¹ / ₁₆ "
12	B4 (JIS)	257mm × 364mm	60	A4 Plus	210mm × 330mm
13	B5 (JIS)	182mm × 257mm	61	A5 Transverse	148mm × 210mm
14	Folio	8 ¹ / ₂ " × 13"	62	B5 (JIS) Transverse	182mm × 257mm
15	Quarto	215mm × 275mm	63	A3 Extra	322mm × 445mm
16	10×14	10" × 14"	64	A5 Extra	174mm × 235mm
17	11×17	11" × 17"	65	B5 (ISO) Extra	201mm × 276mm
18	Note	8 ¹ / ₂ " × 11"	66	A2	420mm × 594mm
19	Envelope #9	3 ⁷ / ₈ " × 8 ⁷ / ₈ "	67	A3 Transverse	297mm × 420mm
20	Envelope #10	4 ¹ / ₈ " × 9 ¹ / ₂ "	68	A3 Extra Transverse	322mm × 445mm
21	Envelope #11	4 ¹ / ₂ " × 10 ³ / ₈ "	69	Dbl. Japanese Postcard	200mm × 148mm
22	Envelope #12	4 ³ / ₄ " × 11"	70	A6	105mm × 148mm
23	Envelope #14	5" × 11 ¹ / ₂ "	71		
24	C	17" × 22"	72		
25	D	22" × 34"	73		
26	E	34" × 44"	74		
27	Envelope DL	110mm × 220mm	75	Letter Rotated	11" × 8 ¹ / ₂ "
28	Envelope C5	162mm × 229mm	76	A3 Rotated	420mm × 297mm
29	Envelope C3	324mm × 458mm	77	A4 Rotated	297mm × 210mm
30	Envelope C4	229mm × 324mm	78	A5 Rotated	210mm × 148mm
31	Envelope C6	114mm × 162mm	79	B4 (JIS) Rotated	364mm × 257mm
32	Envelope C6/C5	114mm × 229mm	80	B5 (JIS) Rotated	257mm × 182mm
33	B4 (ISO)	250mm × 353mm	81	Japanese Postcard Rot.	148mm × 100mm
34	B5 (ISO)	176mm × 250mm	82	Dbl. Jap. Postcard Rot.	148mm × 200mm
35	B6 (ISO)	125mm × 176mm	83	A6 Rotated	148mm × 105mm
36	Envelope Italy	110mm × 230mm	84		
37	Envelope Monarch	3 ⁷ / ₈ " × 7 ¹ / ₂ "	85		
38	6 ³ / ₄ Envelope	3 ⁵ / ₈ " × 6 ¹ / ₂ "	86		
39	US Standard Fanfold	14 ⁷ / ₈ " × 11"	87		
40	German Std. Fanfold	8 ¹ / ₂ " × 12"	88	B6 (JIS)	128mm × 182mm
41	German Legal Fanfold	8 ¹ / ₂ " × 13"	89	B6 (JIS) Rotated	182mm × 128mm
42	B4 (ISO)	250mm × 353mm	90	12×11	12" × 11"
43	Japanese Postcard	100mm × 148mm			
44	9×11	9" × 11"			
45	10×11	10" × 11"			
46	15×11	15" × 11"			
47	Envelope Invite	220mm × 220mm			

6.90 SHEETHDR

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	008F _H	—	—	—	—

This record occurs only in BIFF4W workbook files. It precedes a BIFF4S *Worksheet Substream* that describes a single sheet in the BIFF4W workbook. It contains the sheet name and the byte length of the following substream data. Adding this substream length to the stream position of the following BOF record gives the position of the next SHEETHDR record. See →5.1.2 for details about the BIFF4W workbook stream.

Record SHEETHDR, BIFF4W:

Offset	Size	Contents
0	4	Byte length of the following BIFF4S <i>Worksheet Substream</i>
4	var.	Name of the sheet (byte string, 8-bit string length, →3.3)

6.91 SHEETLAYOUT

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	—	—	—	0862 _H

This record stores the colour of the tab below the sheet containing the sheet name.

Record SHEETLAYOUT, BIFF8X:

Offset	Size	Contents
0	2	0862 _H (repeated record identifier)
2	10	Not used
12	4	Unknown data: 14 _H 00 _H 00 _H 00 _H
16	2	Colour index (→6.70) for sheet name tab
18	2	Not used

6.92 SHEETPROTECTION

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	—	—	—	0867 _H

This record is part of the worksheet/workbook protection (→5.18). It stores additional options for sheet protection.

Record SHEETPROTECTION, BIFF8X:

Offset	Size	Contents
0	2	0867 _H (repeated record identifier)
2	9	Not used
11	8	Unknown data: 02 _H 00 _H 01 _H 00 _H FF _H FF _H FF _H FF _H
19	2	Option flags, see below (default: 4400 _H)
21	2	Not used

The following flags specify, which actions are allowed while the sheet is protected. The state of object and scenario protection is also contained in the records OBJECTPROTECT (→6.69) and SCENPROTECT (→6.86). These records are stored for compatibility to Excel 97 and Excel 2000.

Option flags for sheet protection (a set bit specifies that the action is allowed):

Bit	Mask	Contents	Bit	Mask	Contents
0	0001 _H	Edit objects	8	0100 _H	Delete columns
1	0002 _H	Edit scenarios	9	0200 _H	Delete rows
2	0004 _H	Change cell formatting	10	0400 _H	Select locked cells
3	0008 _H	Change column formatting	11	0800 _H	Sort a cell range
4	0010 _H	Change row formatting	12	1000 _H	Edit auto filters
5	0020 _H	Insert columns	13	2000 _H	Edit PivotTables
6	0040 _H	Insert rows	14	4000 _H	Select unlocked cells
7	0080 _H	Insert hyperlinks			

6.93 SHEETSOFFSET

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	008E _H	—	—	—	—

This record occurs only in BIFF4W workbook files. It is located in the *Workbook Globals Substream* and contains the stream position of the first SHEETHDR record (→6.90). See →5.1.2 for details about the BIFF4W workbook stream.

Record SHEETSOFFSET, BIFF4W:

Offset	Size	Contents
0	4	Absolute stream position of the first SHEETHDR record (from beginning of the stream)

6.94 SHRFMLA – Shared Formula

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	04BC _H	04BC _H	04BC _H	04BC _H

This record stores the token array of a shared formula. Shared formulas are similar to array formulas, they store a formula used in a range of cells. The SHRFMLA record is not a real cell record, but follows the first FORMULA record (→6.46) of the cell range. For more information about shared formulas see →5.8.

Record SHRFMLA, BIFF5-BIFF8:

Offset	Size	Contents
0	6	Cell range address of the area used by the shared formula (→3.13.1). Column indexes are always 8-bit values, also in BIFF8.
6	1	Not used
7	1	Number of existing FORMULA records for this shared formula
8	var.	Token array of the shared formula (→4)

6.95 SORT

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	0090 _H	0090 _H	0090 _H	0090 _H

This record stores the last settings from the “Sort” dialogue for each sheet. These settings are not attached to a cell range in the sheet, that means, it is not possible to determine the cell range sorted with the settings of this record.

Record SORT, BIFF5/BIFF7:

Offset	Size	Contents
0	2	Option flags, see below
2	1	Length of first sort key (character count, <u>1.1</u>)
3	1	Length of second sort key (character count, <u>1.2</u>)
4	1	Length of third sort key (character count, <u>1.3</u>)
5	<u>1.1</u>	Character array of first sort key
[5+ <u>1.1</u>]	<u>1.2</u>	(optional) Character array of description text
[5+ <u>1.1</u> + <u>1.2</u>]	<u>1.3</u>	(optional) Character array of help topic text
var.	1	Not used

Record SORT, BIFF8:

Offset	Size	Contents
0	2	Option flags, see below
2	1	Length of first sort key (character count, <u>1.1</u>)
3	1	Length of second sort key (character count, <u>1.2</u>)
4	1	Length of third sort key (character count, <u>1.3</u>)
5	var.	First sort key (Unicode string without length field, →3.4)
[var.]	var.	(optional, only if <u>1.2</u> > 0) Second sort key (Unicode string without length field, →3.4)
[var.]	var.	(optional, only if <u>1.3</u> > 0) Third sort key (Unicode string without length field, →3.4)
var.	1	Not used

Option flags, BIFF5-BIFF8:

Bit	Mask	Contents
0	0001 _H	0 = Sort rows (top-to-bottom) 1 = Sort columns (left-to-right)
1	0002 _H	0 = Sort first key in ascending order 1 = Sort first key in descending order
2	0004 _H	0 = Sort second key in ascending order 1 = Sort second key in descending order
3	0008 _H	0 = Sort third key in ascending order 1 = Sort third key in descending order
4	0010 _H	0 = Sort case-insensitive 1 = Sort case-sensitive
9-5	03E0 _H	<i>One-based</i> index into the table of defined sort lists, or 0 for sorting without a list

6.96 SST – Shared String Table

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	—	—	00FC _H	00FC _H

This record contains a list of all strings used anywhere in the workbook. Each string occurs only once. The workbook uses indexes into the list to reference the strings. See →5.11 for more information.

Record SST, BIFF8:

Offset	Size	Contents
0	4	Total number of strings in the workbook (see below)
4	4	Number of following strings (<u>nm</u>)
8	var.	List of <u>nm</u> Unicode strings, 16-bit string length (→3.4)

The first field of the SST record counts the total occurrence of strings in the workbook. For instance, the string “AAA” is used 3 times and the string “BBB” is used 2 times. The first field contains 5 and the second field contains 2, followed by the two strings.

6.97 STANDARDWIDTH

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	0099 _H	0099 _H	0099 _H	0099 _H	0099 _H	0099 _H

This record specifies the default column width for columns that have a set bit in the GCW record (BIFF4-BIFF7, →6.47), or columns that do not have a corresponding COLINFO record (BIFF8, →6.18).

Record STANDARDWIDTH, BIFF4-BIFF8:

Offset	Size	Contents
4	2	Default width of the columns in $1/256$ of the width of the zero character, using default font (first FONT record in the file)

6.98 STRING

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0007 _H	0207 _H	0207 _H	0207 _H	0207 _H	0207 _H	0207 _H	0207 _H

This record stores the result of a string formula. It occurs directly after a string formula (→6.46).

Record STRING, BIFF2:

Offset	Size	Contents
0	var.	Byte string, 8-bit string length (→3.3)

Record STRING, BIFF3-BIFF7:

Offset	Size	Contents
0	var.	Byte string, 16-bit string length (→3.3)

In BIFF8 files no STRING record occurs, if the result string is empty.

Record STRING, BIFF8:

Offset	Size	Contents
0	var.	Non-empty Unicode string, 16-bit string length (→3.4)

6.100 SUPBOOK – External Workbook

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	—	—	01AE _H	01AE _H

This record mainly stores the URL of an external document and a list of sheet names inside this document. Furthermore it is used to store DDE and OLE object links, or to indicate an internal 3D reference or an add-in function. See →5.10.3 for details about external references in BIFF8.

6.100.1 External References

A SUPBOOK record for external references stores the URL and a list of sheet names.

Record SUPBOOK for external references, BIFF8:

Offset	Size	Contents
0	2	Number of sheet names (<u>nm</u>)
2	var.	Encoded URL without sheet name (→3.9.1). Unicode string, 16-bit string length (→3.4).
var.	var.	List of <u>nm</u> sheet names (Unicode strings with 16-bit string length, →3.4)

6.100.2 Internal References

In each file occurs a SUPBOOK that is used for internal 3D references. It stores the number of sheets of the own document.

Record SUPBOOK for 3D references, BIFF8:

Offset	Size	Contents
0	2	Number of sheets in this document
2	2	01 _H 04 _H (relict of BIFF5/BIFF7, the byte string “<04 _H >”, see →3.9.1)

6.100.3 Add-In Functions

Add-in function names are stored in EXTERNNAME records following this SUPBOOK record.

Record SUPBOOK for add-in functions, BIFF8:

Offset	Size	Contents
0	2	0001 _H
2	2	01 _H 3A _H (relict of BIFF5/BIFF7, the byte string “#”, see EXTERNSHEET record, →6.39)

6.100.4 DDE Links, OLE Object Links

The SUPBOOK record of a DDE link or an OLE object link contains the name of the server application (DDE) or the class name (OLE) and the name of a source document. These names are encoded in one string.

Record SUPBOOK for DDE links and OLE object links, BIFF8:

Offset	Size	Contents
0	2	0000 _H
2	var.	Encoded source document name (→3.9.2). Unicode string, 16-bit string length (→3.4).

6.101 TABLEOP – Multiple Operation Table

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0036 _H	0236 _H	0236 _H	0236 _H	0236 _H	0236 _H	0236 _H	0236 _H

This record stores information about a multiple operation table in the sheet. It follows the first FORMULA record (→6.46) of the cell range containing the operation table. For more information about multiple operations see →5.9.

In BIFF2, a multiple operation with two data source ranges is stored in the TABLEOP2 record, see →6.102. From BIFF3 on, the TABLEOP record is able to represent this kind of multiple operation too.

Record TABLEOP, BIFF2:

Offset	Size	Contents
0	6	Cell range address of the multiple operation table range (→3.13.1)
6	1	0 = Do not recalculate the table, 1 = Always recalculate the table
7	1	0 = Input data is in the column left of the table, formulas are in the row above the table 1 = Input data is in the row above the table, formulas are in the column left of the table
8	2	Index to row of the input cell
10	2	Index to column of the input cell

Record TABLEOP, BIFF3-BIFF8:

Offset	Size	Contents												
0	6	Cell range address of the multiple operation table range (→3.13.1). Column indexes are always 8-bit values, also in BIFF8.												
6	2	Option flags: <table border="1" data-bbox="491 1010 1487 1339"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0001_H</td> <td>1 = Always recalculate array formula</td> </tr> <tr> <td>1</td> <td>0002_H</td> <td>1 = Calculate array formula on open</td> </tr> <tr> <td>3-2</td> <td>000C_H</td> <td>Multiple operation table mode: <ul style="list-style-type: none"> • 00₂ = Input data is in the column left of the table, formulas are in the row above the table • 01₂ = Input data is in the row above the table, formulas are in the column left of the table • 1x₂ = Table uses row and column input data (x = not used) </td> </tr> </tbody> </table>	Bit	Mask	Contents	0	0001 _H	1 = Always recalculate array formula	1	0002 _H	1 = Calculate array formula on open	3-2	000C _H	Multiple operation table mode: <ul style="list-style-type: none"> • 00₂ = Input data is in the column left of the table, formulas are in the row above the table • 01₂ = Input data is in the row above the table, formulas are in the column left of the table • 1x₂ = Table uses row and column input data (x = not used)
Bit	Mask	Contents												
0	0001 _H	1 = Always recalculate array formula												
1	0002 _H	1 = Calculate array formula on open												
3-2	000C _H	Multiple operation table mode: <ul style="list-style-type: none"> • 00₂ = Input data is in the column left of the table, formulas are in the row above the table • 01₂ = Input data is in the row above the table, formulas are in the column left of the table • 1x₂ = Table uses row and column input data (x = not used) 												
8	2	Index to row of input cell (in mode 1x ₂ index to row of input cell for row input)												
10	2	Index to column of input cell (in mode 1x ₂ index to column of input cell for row input)												
12	2	In mode 1x ₂ index to row of input cell for column input; else not used												
14	2	In mode 1x ₂ index to column of input cell for column input; else not used												

6.102 TABLEOP2 – Multiple Operation Table

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0037 _H	—	—	—	—	—	—	—

In BIFF2 this record stores information about a multiple operation table in a sheet using two independent input ranges. It follows the first FORMULA record (→6.46) of the cell range containing the operation table. For more information about multiple operations see →5.9. From BIFF3 on, the TABLEOP record (→6.101) contains this kind of multiple operation too.

Record TABLEOP2, BIFF2:

Offset	Size	Contents
0	6	Cell range address of the multiple operation table range (→3.13.1)
6	1	0 = Do not recalculate the table, 1 = Always recalculate the table
7	1	Not used
8	2	Index to row of input cell for row input
10	2	Index to column of input cell for row input
12	2	Index to row of input cell for column input
14	2	Index to column of input cell for column input

6.103 TOPMARGIN

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0028 _H	0028 _H	0028 _H	0028 _H	0028 _H	0028 _H	0028 _H	0028 _H

This record is part of the *Page Settings Block* (→5.4). It contains the top page margin of the current worksheet.

Record TOPMARGIN, BIFF2-BIFF8:

Offset	Size	Contents
0	8	Top page margin in inches (IEEE 754 floating-point value, 64-bit double precision)

6.104 UNCALCED

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	005E _H	005E _H	005E _H	005E _H	005E _H	005E _H	005E _H

If this record occurs in the *Worksheet Substream*, it indicates that the formulas have not been recalculated before the document was saved.

Record UNCALCED, BIFF2-BIFF8:

Offset	Size	Contents
0	2	Not used

6.105 USESELFS

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	—	—	—	—	—	0160 _H	0160 _H

This record specifies if the formulas in the workbook can use “natural language formulas”. This type of formula can refer to cells by its content or the content of the column or row header cell.

Record USESELFS, BIFF8:

Offset	Size	Contents
0	2	0 = Do not use natural language formulas 1 = Use natural language formulas

6.106 VCENTER

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	0084 _H	0084 _H	0084 _H	0084 _H	0084 _H	0084 _H	0084 _H

This record is part of the *Page Settings Block* (→5.4). It specifies if the sheet is centred vertically when printed.

Record VCENTER, BIFF3-BIFF8:

Offset	Size	Contents
0	2	0 = Print sheet aligned at top page border 1 = Print sheet vertically centred

6.107 VERTICALPAGEBREAKS

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
001A _H	001A _H	001A _H	001A _H	001A _H	001A _H	001A _H	001A _H

This record is part of the *Page Settings Block* (→5.4). It contains all vertical manual page breaks.

Record VERTICALPAGEBREAKS, BIFF2-BIFF7:

Offset	Size	Contents
0	2	Number of following column indexes (<u>nm</u>)
2	2· <u>nm</u>	List of <u>nm</u> column indexes. Each index specifies the first column after the page break.

Record VERTICALPAGEBREAKS, BIFF8:

Offset	Size	Contents												
0	2	Number of following column index structures (<u>nm</u>)												
2	6· <u>nm</u>	List of <u>nm</u> column index structures. Each column index structure contains: <table border="1" data-bbox="416 1238 1402 1388"> <thead> <tr> <th>Offset</th> <th>Size</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>2</td> <td>Index to first column following the page break</td> </tr> <tr> <td>2</td> <td>2</td> <td>Index to first row of this page break</td> </tr> <tr> <td>4</td> <td>2</td> <td>Index to last row of this page break</td> </tr> </tbody> </table>	Offset	Size	Contents	0	2	Index to first column following the page break	2	2	Index to first row of this page break	4	2	Index to last row of this page break
Offset	Size	Contents												
0	2	Index to first column following the page break												
2	2	Index to first row of this page break												
4	2	Index to last row of this page break												

The column indexes in the lists must be ordered ascending. If in BIFF8 a column contains several page breaks, they must be ordered ascending by start row index.

6.108 WINDOW1

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
003D _H	003D _H	003D _H	003D _H	003D _H	003D _H	003D _H	003D _H

This record contains general settings for the document window and global workbook settings (BIFF5-BIFF8). In BIFF4W this record occurs in every worksheet.

Record WINDOW1, BIFF2-BIFF4:

Offset	Size	Contents
0	2	Horizontal position of the document window (in twips = $\frac{1}{20}$ of a point)
2	2	Vertical position of the document window (in twips = $\frac{1}{20}$ of a point)
4	2	Width of the document window (in twips = $\frac{1}{20}$ of a point)
6	2	Height of the document window (in twips = $\frac{1}{20}$ of a point)
8	2	0 = Window is visible; 1 = Window is hidden

Record WINDOW1, BIFF5-BIFF8:

Offset	Size	Contents																		
0	2	Horizontal position of the document window (in twips = $\frac{1}{20}$ of a point)																		
2	2	Vertical position of the document window (in twips = $\frac{1}{20}$ of a point)																		
4	2	Width of the document window (in twips = $\frac{1}{20}$ of a point)																		
6	2	Height of the document window (in twips = $\frac{1}{20}$ of a point)																		
8	2	Option flags: <table border="1" data-bbox="491 994 1471 1218"> <thead> <tr> <th>Bits</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0001_H</td> <td>0 = Window is visible 1 = Window is hidden</td> </tr> <tr> <td>1</td> <td>0002_H</td> <td>0 = Window is open 1 = Window is minimised</td> </tr> <tr> <td>3</td> <td>0008_H</td> <td>0 = Horizontal scroll bar hidden 1 = Horizontal scroll bar visible</td> </tr> <tr> <td>4</td> <td>0010_H</td> <td>0 = Vertical scroll bar hidden 1 = Vertical scroll bar visible</td> </tr> <tr> <td>5</td> <td>0020_H</td> <td>0 = Worksheet tab bar hidden 1 = Worksheet tab bar visible</td> </tr> </tbody> </table>	Bits	Mask	Contents	0	0001 _H	0 = Window is visible 1 = Window is hidden	1	0002 _H	0 = Window is open 1 = Window is minimised	3	0008 _H	0 = Horizontal scroll bar hidden 1 = Horizontal scroll bar visible	4	0010 _H	0 = Vertical scroll bar hidden 1 = Vertical scroll bar visible	5	0020 _H	0 = Worksheet tab bar hidden 1 = Worksheet tab bar visible
Bits	Mask	Contents																		
0	0001 _H	0 = Window is visible 1 = Window is hidden																		
1	0002 _H	0 = Window is open 1 = Window is minimised																		
3	0008 _H	0 = Horizontal scroll bar hidden 1 = Horizontal scroll bar visible																		
4	0010 _H	0 = Vertical scroll bar hidden 1 = Vertical scroll bar visible																		
5	0020 _H	0 = Worksheet tab bar hidden 1 = Worksheet tab bar visible																		
10	2	Index to active (displayed) worksheet																		
12	2	Index of first visible tab in the worksheet tab bar																		
14	2	Number of selected worksheets (highlighted in the worksheet tab bar)																		
16	2	Width of worksheet tab bar (in $\frac{1}{1000}$ of window width). The remaining space is used by the horizontal scrollbar.																		

6.109 WINDOW2

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
003E _H	023E _H	023E _H	023E _H	023E _H	023E _H	023E _H	023E _H

This record contains additional settings for the document window (BIFF2-BIFF4S) or for the window of a specific worksheet (BIFF4W-BIFF8). It is part of the *Sheet View Settings Block* (→5.5).

• Contents of the WINDOW2 Record

Record WINDOW2, BIFF2:

Offset	Size	Contents
0	1	0 = Show formula results 1 = Show formulas
1	1	0 = Do not show grid lines 1 = Show grid lines
2	1	0 = Do not show sheet headers 1 = Show sheet headers
3	1	0 = Panes are not frozen 1 = Panes are frozen (<i>f.reeze</i>)
4	1	0 = Show zero values as empty cells 1 = Show zero values
5	2	Index to first visible row
7	2	Index to first visible column
9	1	0 = Use manual grid line colour (below) 1 = Use automatic grid line colour
10	4	Grid line RGB colour (→3.5)

Record WINDOW2, BIFF3-BIFF7:

Offset	Size	Contents
0	2	Option flags (see below)
2	2	Index to first visible row
4	2	Index to first visible column
6	4	Grid line RGB colour (→3.5)

Record WINDOW2, BIFF8:

Offset	Size	Contents
0	2	Option flags (see below)
2	2	Index to first visible row
4	2	Index to first visible column
6	2	Colour index of grid line colour (→6.70). Note that in BIFF2-BIFF7 an RGB colour is written instead.
8	2	Not used
10	2	Cached magnification factor in page break preview (in percent); 0 = Default (60%)
12	2	Cached magnification factor in normal view (in percent); 0 = Default (100%)
14	4	Not used

In BIFF8 this record stores used magnification factors for page break preview and normal view. These values are used to restore the magnification, when the view is changed. The real magnification of the currently active view is stored in the SCL record (→6.87). The type of the active view is stored in the option flags field (see below).

• Option Flags

Option flags, BIFF3-BIFF8:

Bits	Mask	Contents	
0	0001 _H	0 = Show formula results	1 = Show formulas
1	0002 _H	0 = Do not show grid lines	1 = Show grid lines
2	0004 _H	0 = Do not show sheet headers	1 = Show sheet headers
3	0008 _H	0 = Panes are not frozen	1 = Panes are frozen (<u>f</u> reeze)
4	0010 _H	0 = Show zero values as empty cells	1 = Show zero values
5	0020 _H	0 = Manual grid line colour	1 = Automatic grid line colour
6	0040 _H	0 = Columns from left to right	1 = Columns from right to left
7	0080 _H	0 = Do not show outline symbols	1 = Show outline symbols
8	0100 _H	0 = Keep splits if pane freeze is removed	1 = Remove splits if pane freeze is removed
9	0200 _H	0 = Sheet not selected	1 = Sheet selected (BIFF5-BIFF8)
10	0400 _H	0 = Sheet not visible	1 = Sheet visible (BIFF5-BIFF8)
11	0800 _H	0 = Show in normal view	1 = Show in page break preview (BIFF8)

The freeze flag specifies, if a following PANE record (→6.71) describes unfrozen or frozen panes.

6.110 WINDOWPROTECT

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0019 _H	0019 _H	0019 _H	0019 _H	0019 _H	0019 _H	0019 _H	0019 _H

This record is part of the worksheet/workbook protection (→5.18). It determines whether the window configuration of this document is protected. Window protection is not active, if this record is omitted.

Record WINDOWPROTECT, BIFF2-BIFF8:

Offset	Size	Contents
0	2	0 = Window settings not protected; 1 = Window settings protected

6.111 WRITEACCESS

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	005C _H	005C _H	005C _H	005C _H	005C _H	005C _H	005C _H

This record is part of the file protection (→5.19). It contains the name of the user that has saved the file. The user name is always stored as an equal-sized string. All unused characters after the name are filled with space characters. It is not required to write the mentioned string length. Every other length will be accepted too.

Record WRITEACCESS, BIFF3-BIFF4:

Offset	Size	Contents
0	32	User name, byte string, 8-bit string length, 31 characters (→3.3)

Record WRITEACCESS, BIFF5/BIFF7:

Offset	Size	Contents
0	54	User name, byte string, 8-bit string length, 53 characters (→3.3)

Record WRITEACCESS, BIFF8:

Offset	Size	Contents
0	var.	User name, Unicode string, 16-bit string length, 109 characters (→3.4)

6.112 WRITEPROT

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	0086 _H	0086 _H	0086 _H	0086 _H	0086 _H	0086 _H	0086 _H

This record is part of the file protection (→5.19). It does not contain any data. If present it specifies that the file is write protected. The write protection password is stored in the FILESHARING record (→6.42).

6.113 WSBOOL

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	0081 _H	0081 _H	0081 _H	0081 _H	0081 _H	0081 _H	0081 _H

This record stores a 16-bit value with Boolean options for the current sheet. From BIFF5 on the “Save external linked values” option is moved to the record BOOKBOOL (→6.9).

Option flags of record WSBOOL, BIFF3-BIFF8:

Bit	Mask	Contents
0	0001 _H	0 = Do not show automatic page breaks 1 = Show automatic page breaks
4	0010 _H	0 = Standard sheet 1 = Dialogue sheet (BIFF5-BIFF8)
5	0020 _H	0 = No automatic styles in outlines 1 = Apply automatic styles to outlines
6	0040 _H	0 = Outline buttons above outline group 1 = Outline buttons below outline group
7	0080 _H	0 = Outline buttons left of outline group 1 = Outline buttons right of outline group
8	0100 _H	0 = Scale printout in percent (→6.89) 1 = Fit printout to number of pages (→6.89)
9	0200 _H	0 = Save external linked values (BIFF3-BIFF4 only, →5.10) 1 = <i>Do not</i> save external linked values (BIFF3-BIFF4 only, →5.10)
10	0400 _H	0 = Do not show row outline symbols 1 = Show row outline symbols
11	0800 _H	0 = Do not show column outline symbols 1 = Show column outline symbols
13-12	3000 _H	These flags specify the arrangement of windows. They are stored in BIFF4 only. 00 ₂ = Arrange windows tiled 01 ₂ = Arrange windows horizontal 10 ₂ = Arrange windows vertical 11 ₂ = Arrange windows cascaded

The following flags are valid for BIFF4-BIFF8 only:

14	4000 _H	0 = Standard expression evaluation 1 = Alternative expression evaluation
15	8000 _H	0 = Standard formula entries 1 = Alternative formula entries

6.114 XCT – CRN Count

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
—	0059 _H	0059 _H	0059 _H	0059 _H	0059 _H	0059 _H	0059 _H

This record stores the number of immediately following CRN records. These records are used to store the cell contents of external references. See →5.10 for details about external references.

Record XCT, BIFF3-BIFF7:

Offset	Size	Contents
0	2	Number of following CRN records (→6.24)

Record XCT, BIFF8:

Offset	Size	Contents
0	2	Number of following CRN records (→6.24)
2	2	Index into sheet table of the involved SUPBOOK record (→6.100)

6.115 XF – Extended Format

BIFF2	BIFF3	BIFF4S	BIFF4W	BIFF5	BIFF7	BIFF8	BIFF8X
0043 _H	0243 _H	0443 _H	0443 _H	00E0 _H	00E0 _H	00E0 _H	00E0 _H

This record contains formatting information for cells, rows, columns or styles.

6.115.1 XF Substructures

From BIFF3 on, some of the elements occur unchanged in every BIFF version. These elements are described in the following using a specific name for each element. In the description of the record structure the names are used to reference to these tables.

• XF_TYPE_PROT – XF Type and Cell Protection (3 Bits), BIFF3-BIFF8

These 3 bits are part of a specific data byte.

Bit	Mask	Contents
0	01 _H	1 = Cell is locked
1	02 _H	1 = Formula is hidden
2	04 _H	0 = Cell XF; 1 = Style XF

• XF_USED_ATTRIB – Attributes Used from Parent Style XF (6 Bits), BIFF3-BIFF8

Each bit describes the validity of a specific group of attributes. In cell XFs a cleared bit means the attributes of the parent style XF are used (but only if the attributes are valid there), a set bit means the attributes of this XF are used. In style XFs a cleared bit means the attribute setting is valid, a set bit means the attribute should be ignored.

Bit	Mask	Contents
0	01 _H	Flag for number format
1	02 _H	Flag for font
2	04 _H	Flag for horizontal and vertical alignment, text wrap, indentation, orientation, rotation, and text direction
3	08 _H	Flag for border lines
4	10 _H	Flag for background area style
5	20 _H	Flag for cell protection (cell locked and formula hidden)

• XF_HOR_ALIGN – Horizontal Alignment (3 Bits), BIFF2-BIFF8

The horizontal alignment consists of 3 bits and is part of a specific data byte.

Value	Horizontal alignment
00 _H	General
01 _H	Left
02 _H	Centred
03 _H	Right
04 _H	Filled
05 _H	Justified (BIFF4-BIFF8X)
06 _H	Centred across selection (BIFF4-BIFF8X)
07 _H	Distributed (BIFF8X)

• **XF_VERT_ALIGN – Vertical Alignment (2 or 3 Bits), BIFF4-BIFF8**

The vertical alignment consists of 2 bits (BIFF4) or 3 bits (BIFF5-BIFF8) and is part of a specific data byte. Vertical alignment is not available in BIFF2 and BIFF3.

Value	Vertical alignment
00 _H	Top
01 _H	Centred
02 _H	Bottom
03 _H	Justified (BIFF5-BIFF8X)
04 _H	Distributed (BIFF8X)

• **XF_ORIENTATION – Text Orientation (2 Bits), BIFF4-BIFF7**

In the BIFF versions BIFF4-BIFF7, text can be rotated in steps of 90 degrees or stacked. The orientation mode consists of 2 bits and is part of a specific data byte. In BIFF8 a rotation angle occurs instead of these flags.

Value	Text orientation
00 _H	Not rotated
01 _H	Letters are stacked top-to-bottom, but not rotated
02 _H	Text is rotated 90 degrees counterclockwise
03 _H	Text is rotated 90 degrees clockwise

• **XF_ROTATION – Text Rotation Angle (1 Byte), BIFF8**

Value	Text rotation
0	Not rotated
1-90	1 to 90 degrees counterclockwise
91-180	1 to 90 degrees clockwise
255	Letters are stacked top-to-bottom, but not rotated

• **XF_BORDER_34 – Cell Border Style (4 Bytes), BIFF3-BIFF4**

Cell borders contain a line style and a line colour for each line of the border.

Bit	Mask	Contents
2-0	00000007 _H	Top line style (→3.10)
7-3	00000F8 _H	Colour index (→6.70) for top line colour
10-8	00000700 _H	Left line style (→3.10)
15-11	0000F800 _H	Colour index (→6.70) for left line colour
18-16	00070000 _H	Bottom line style (→3.10)
23-19	00F80000 _H	Colour index (→6.70) for bottom line colour
26-24	07000000 _H	Right line style (→3.10)
31-27	F8000000 _H	Colour index (→6.70) for right line colour

• **XF_AREA_34 – Cell Background Area Style (2 Bytes), BIFF3-BIFF4**

A cell background area style contains an area pattern and a foreground and background colour.

Bit	Mask	Contents
5-0	003F _H	Fill pattern (→3.11)
10-6	07C0 _H	Colour index (→6.70) for pattern colour
15-11	F800 _H	Colour index (→6.70) for pattern background

6.115.2 XF Record Contents

Record XF, BIFF2:

Offset	Size	Contents																					
0	1	Index to FONT record (→6.43)																					
1	1	Not used																					
2	1	<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>5-0</td> <td>3F_H</td> <td>Index to FORMAT record (→6.45)</td> </tr> <tr> <td>6</td> <td>40_H</td> <td>1 = Cell is locked</td> </tr> <tr> <td>7</td> <td>80_H</td> <td>1 = Formula is hidden</td> </tr> </tbody> </table>	Bit	Mask	Contents	5-0	3F _H	Index to FORMAT record (→6.45)	6	40 _H	1 = Cell is locked	7	80 _H	1 = Formula is hidden									
		Bit	Mask	Contents																			
		5-0	3F _H	Index to FORMAT record (→6.45)																			
		6	40 _H	1 = Cell is locked																			
7	80 _H	1 = Formula is hidden																					
3	1	<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>2-0</td> <td>07_H</td> <td>XF_HOR_ALIGN – Horizontal alignment (see above)</td> </tr> <tr> <td>3</td> <td>08_H</td> <td>1 = Cell has left black border</td> </tr> <tr> <td>4</td> <td>10_H</td> <td>1 = Cell has right black border</td> </tr> <tr> <td>5</td> <td>20_H</td> <td>1 = Cell has top black border</td> </tr> <tr> <td>6</td> <td>40_H</td> <td>1 = Cell has bottom black border</td> </tr> <tr> <td>7</td> <td>80_H</td> <td>1 = Cell has shaded background</td> </tr> </tbody> </table>	Bit	Mask	Contents	2-0	07 _H	XF_HOR_ALIGN – Horizontal alignment (see above)	3	08 _H	1 = Cell has left black border	4	10 _H	1 = Cell has right black border	5	20 _H	1 = Cell has top black border	6	40 _H	1 = Cell has bottom black border	7	80 _H	1 = Cell has shaded background
		Bit	Mask	Contents																			
		2-0	07 _H	XF_HOR_ALIGN – Horizontal alignment (see above)																			
		3	08 _H	1 = Cell has left black border																			
		4	10 _H	1 = Cell has right black border																			
		5	20 _H	1 = Cell has top black border																			
		6	40 _H	1 = Cell has bottom black border																			
7	80 _H	1 = Cell has shaded background																					

Record XF, BIFF3:

Offset	Size	Contents												
0	1	Index to FONT record (→6.43)												
1	1	Index to FORMAT record (→6.45)												
2	1	XF_TYPE_PROT – XF type and cell protection (see above)												
3	1	<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>7-2</td> <td>FC_H</td> <td>XF_USED_ATTRIB – Used attributes (see above)</td> </tr> </tbody> </table>	Bit	Mask	Contents	7-2	FC _H	XF_USED_ATTRIB – Used attributes (see above)						
		Bit	Mask	Contents										
7-2	FC _H	XF_USED_ATTRIB – Used attributes (see above)												
4	2	<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>2-0</td> <td>0007_H</td> <td>XF_HOR_ALIGN – Horizontal alignment (see above)</td> </tr> <tr> <td>3</td> <td>0008_H</td> <td>1 = Text is wrapped at right border</td> </tr> <tr> <td>15-4</td> <td>FFF0_H</td> <td>Index to parent style XF (always FFF_H in style XFs)</td> </tr> </tbody> </table>	Bit	Mask	Contents	2-0	0007 _H	XF_HOR_ALIGN – Horizontal alignment (see above)	3	0008 _H	1 = Text is wrapped at right border	15-4	FFF0 _H	Index to parent style XF (always FFF _H in style XFs)
		Bit	Mask	Contents										
		2-0	0007 _H	XF_HOR_ALIGN – Horizontal alignment (see above)										
3	0008 _H	1 = Text is wrapped at right border												
15-4	FFF0 _H	Index to parent style XF (always FFF _H in style XFs)												
6	2	XF_AREA_34 – Cell background area (see above)												
8	4	XF_BORDER_34 – Cell border lines (see above)												

Record XF, BIFF4:

Offset	Size	Contents															
0	1	Index to FONT record (→6.43)															
1	1	Index to FORMAT record (→6.45)															
2	2	<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>2-0</td> <td>0007_H</td> <td>XF_TYPE_PROT – XF type, cell protection (see above)</td> </tr> <tr> <td>15-4</td> <td>FFF0_H</td> <td>Index to parent style XF (always FFF_H in style XFs)</td> </tr> </tbody> </table>	Bit	Mask	Contents	2-0	0007 _H	XF_TYPE_PROT – XF type, cell protection (see above)	15-4	FFF0 _H	Index to parent style XF (always FFF _H in style XFs)						
		Bit	Mask	Contents													
		2-0	0007 _H	XF_TYPE_PROT – XF type, cell protection (see above)													
15-4	FFF0 _H	Index to parent style XF (always FFF _H in style XFs)															
4	1	<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>2-0</td> <td>07_H</td> <td>XF_HOR_ALIGN – Horizontal alignment (see above)</td> </tr> <tr> <td>3</td> <td>08_H</td> <td>1 = Text is wrapped at right border</td> </tr> <tr> <td>5-4</td> <td>30_H</td> <td>XF_VERT_ALIGN – Vertical alignment (see above)</td> </tr> <tr> <td>7-6</td> <td>C0_H</td> <td>XF_ORIENTATION – Text orientation (see above)</td> </tr> </tbody> </table>	Bit	Mask	Contents	2-0	07 _H	XF_HOR_ALIGN – Horizontal alignment (see above)	3	08 _H	1 = Text is wrapped at right border	5-4	30 _H	XF_VERT_ALIGN – Vertical alignment (see above)	7-6	C0 _H	XF_ORIENTATION – Text orientation (see above)
		Bit	Mask	Contents													
		2-0	07 _H	XF_HOR_ALIGN – Horizontal alignment (see above)													
		3	08 _H	1 = Text is wrapped at right border													
		5-4	30 _H	XF_VERT_ALIGN – Vertical alignment (see above)													
7-6	C0 _H	XF_ORIENTATION – Text orientation (see above)															
5	1	<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>7-2</td> <td>FC_H</td> <td>XF_USED_ATTRIB – Used attributes (see above)</td> </tr> </tbody> </table>	Bit	Mask	Contents	7-2	FC _H	XF_USED_ATTRIB – Used attributes (see above)									
		Bit	Mask	Contents													
7-2	FC _H	XF_USED_ATTRIB – Used attributes (see above)															
6	2	XF_AREA_34 – Cell background area (see above)															
8	4	XF_BORDER_34 – Cell border lines (see above)															

Record XF, BIFF5/BIFF7:

Offset	Size	Contents																					
0	2	Index to FONT record (→6.43)																					
2	2	Index to FORMAT record (→6.45)																					
4	2	<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>2-0</td> <td>0007_H</td> <td>XF_TYPE_PROT – XF type, cell protection (see above)</td> </tr> <tr> <td>15-4</td> <td>FFF0_H</td> <td>Index to parent style XF (always FFF_H in style XFs)</td> </tr> </tbody> </table>	Bit	Mask	Contents	2-0	0007 _H	XF_TYPE_PROT – XF type, cell protection (see above)	15-4	FFF0 _H	Index to parent style XF (always FFF _H in style XFs)												
Bit	Mask	Contents																					
2-0	0007 _H	XF_TYPE_PROT – XF type, cell protection (see above)																					
15-4	FFF0 _H	Index to parent style XF (always FFF _H in style XFs)																					
6	1	<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>2-0</td> <td>07_H</td> <td>XF_HOR_ALIGN – Horizontal alignment (see above)</td> </tr> <tr> <td>3</td> <td>08_H</td> <td>1 = Text is wrapped at right border</td> </tr> <tr> <td>6-4</td> <td>70_H</td> <td>XF_VERT_ALIGN – Vertical alignment (see above)</td> </tr> </tbody> </table>	Bit	Mask	Contents	2-0	07 _H	XF_HOR_ALIGN – Horizontal alignment (see above)	3	08 _H	1 = Text is wrapped at right border	6-4	70 _H	XF_VERT_ALIGN – Vertical alignment (see above)									
Bit	Mask	Contents																					
2-0	07 _H	XF_HOR_ALIGN – Horizontal alignment (see above)																					
3	08 _H	1 = Text is wrapped at right border																					
6-4	70 _H	XF_VERT_ALIGN – Vertical alignment (see above)																					
7	1	<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>1-0</td> <td>03_H</td> <td>XF_ORIENTATION – Text orientation (see above)</td> </tr> <tr> <td>7-2</td> <td>FC_H</td> <td>XF_USED_ATTRIB – Used attributes (see above)</td> </tr> </tbody> </table>	Bit	Mask	Contents	1-0	03 _H	XF_ORIENTATION – Text orientation (see above)	7-2	FC _H	XF_USED_ATTRIB – Used attributes (see above)												
Bit	Mask	Contents																					
1-0	03 _H	XF_ORIENTATION – Text orientation (see above)																					
7-2	FC _H	XF_USED_ATTRIB – Used attributes (see above)																					
8	4	Cell border lines and background area: <table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>6-0</td> <td>0000007F_H</td> <td>Colour index (→6.70) for pattern colour</td> </tr> <tr> <td>13-7</td> <td>00003F80_H</td> <td>Colour index (→6.70) for pattern background</td> </tr> <tr> <td>21-16</td> <td>003F0000_H</td> <td>Fill pattern (→3.11)</td> </tr> <tr> <td>24-22</td> <td>01C00000_H</td> <td>Bottom line style (→3.10)</td> </tr> <tr> <td>31-25</td> <td>FE000000_H</td> <td>Colour index (→6.70) for bottom line colour</td> </tr> </tbody> </table>	Bit	Mask	Contents	6-0	0000007F _H	Colour index (→6.70) for pattern colour	13-7	00003F80 _H	Colour index (→6.70) for pattern background	21-16	003F0000 _H	Fill pattern (→3.11)	24-22	01C00000 _H	Bottom line style (→3.10)	31-25	FE000000 _H	Colour index (→6.70) for bottom line colour			
Bit	Mask	Contents																					
6-0	0000007F _H	Colour index (→6.70) for pattern colour																					
13-7	00003F80 _H	Colour index (→6.70) for pattern background																					
21-16	003F0000 _H	Fill pattern (→3.11)																					
24-22	01C00000 _H	Bottom line style (→3.10)																					
31-25	FE000000 _H	Colour index (→6.70) for bottom line colour																					
12	4	<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>2-0</td> <td>00000007_H</td> <td>Top line style (→3.10)</td> </tr> <tr> <td>5-3</td> <td>00000038_H</td> <td>Left line style (→3.10)</td> </tr> <tr> <td>8-6</td> <td>000001C0_H</td> <td>Right line style (→3.10)</td> </tr> <tr> <td>15-9</td> <td>0000FE00_H</td> <td>Colour index (→6.70) for top line colour</td> </tr> <tr> <td>22-16</td> <td>007F0000_H</td> <td>Colour index (→6.70) for left line colour</td> </tr> <tr> <td>29-23</td> <td>3F800000_H</td> <td>Colour index (→6.70) for right line colour</td> </tr> </tbody> </table>	Bit	Mask	Contents	2-0	00000007 _H	Top line style (→3.10)	5-3	00000038 _H	Left line style (→3.10)	8-6	000001C0 _H	Right line style (→3.10)	15-9	0000FE00 _H	Colour index (→6.70) for top line colour	22-16	007F0000 _H	Colour index (→6.70) for left line colour	29-23	3F800000 _H	Colour index (→6.70) for right line colour
Bit	Mask	Contents																					
2-0	00000007 _H	Top line style (→3.10)																					
5-3	00000038 _H	Left line style (→3.10)																					
8-6	000001C0 _H	Right line style (→3.10)																					
15-9	0000FE00 _H	Colour index (→6.70) for top line colour																					
22-16	007F0000 _H	Colour index (→6.70) for left line colour																					
29-23	3F800000 _H	Colour index (→6.70) for right line colour																					

Record XF, BIFF8:

Offset	Size	Contents																											
0	2	Index to FONT record (→6.43)																											
2	2	Index to FORMAT record (→6.45)																											
4	2	<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>2-0</td> <td>0007_H</td> <td>XF_TYPE_PROT – XF type, cell protection (see above)</td> </tr> <tr> <td>15-4</td> <td>FFF0_H</td> <td>Index to parent style XF (always FFF_H in style XFs)</td> </tr> </tbody> </table>	Bit	Mask	Contents	2-0	0007 _H	XF_TYPE_PROT – XF type, cell protection (see above)	15-4	FFF0 _H	Index to parent style XF (always FFF _H in style XFs)																		
Bit	Mask	Contents																											
2-0	0007 _H	XF_TYPE_PROT – XF type, cell protection (see above)																											
15-4	FFF0 _H	Index to parent style XF (always FFF _H in style XFs)																											
6	1	<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>2-0</td> <td>07_H</td> <td>XF_HOR_ALIGN – Horizontal alignment (see above)</td> </tr> <tr> <td>3</td> <td>08_H</td> <td>1 = Text is wrapped at right border</td> </tr> <tr> <td>6-4</td> <td>70_H</td> <td>XF_VERT_ALIGN – Vertical alignment (see above)</td> </tr> </tbody> </table>	Bit	Mask	Contents	2-0	07 _H	XF_HOR_ALIGN – Horizontal alignment (see above)	3	08 _H	1 = Text is wrapped at right border	6-4	70 _H	XF_VERT_ALIGN – Vertical alignment (see above)															
Bit	Mask	Contents																											
2-0	07 _H	XF_HOR_ALIGN – Horizontal alignment (see above)																											
3	08 _H	1 = Text is wrapped at right border																											
6-4	70 _H	XF_VERT_ALIGN – Vertical alignment (see above)																											
7	1	XF_ROTATION: Text rotation angle (see above)																											
8	1	<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>3-0</td> <td>0F_H</td> <td>Indent level</td> </tr> <tr> <td>4</td> <td>10_H</td> <td>1 = Shrink content to fit into cell</td> </tr> <tr> <td>7-6</td> <td>C0_H</td> <td>Text direction (BIFF8X only) 00₂ = According to context 01₂ = Left-to-right 10₂ = Right-to-left</td> </tr> </tbody> </table>	Bit	Mask	Contents	3-0	0F _H	Indent level	4	10 _H	1 = Shrink content to fit into cell	7-6	C0 _H	Text direction (BIFF8X only) 00 ₂ = According to context 01 ₂ = Left-to-right 10 ₂ = Right-to-left															
Bit	Mask	Contents																											
3-0	0F _H	Indent level																											
4	10 _H	1 = Shrink content to fit into cell																											
7-6	C0 _H	Text direction (BIFF8X only) 00 ₂ = According to context 01 ₂ = Left-to-right 10 ₂ = Right-to-left																											
9	1	<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>7-2</td> <td>FC_H</td> <td>XF_USED_ATTRIB – Used attributes (see above)</td> </tr> </tbody> </table>	Bit	Mask	Contents	7-2	FC _H	XF_USED_ATTRIB – Used attributes (see above)																					
Bit	Mask	Contents																											
7-2	FC _H	XF_USED_ATTRIB – Used attributes (see above)																											
10	4	Cell border lines and background area: <table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>3-0</td> <td>0000000F_H</td> <td>Left line style (→3.10)</td> </tr> <tr> <td>7-4</td> <td>000000F0_H</td> <td>Right line style (→3.10)</td> </tr> <tr> <td>11-8</td> <td>00000F00_H</td> <td>Top line style (→3.10)</td> </tr> <tr> <td>15-12</td> <td>0000F000_H</td> <td>Bottom line style (→3.10)</td> </tr> <tr> <td>22-16</td> <td>007F0000_H</td> <td>Colour index (→6.70) for left line colour</td> </tr> <tr> <td>29-23</td> <td>3F800000_H</td> <td>Colour index (→6.70) for right line colour</td> </tr> <tr> <td>30</td> <td>40000000_H</td> <td>1 = Diagonal line from top left to right bottom</td> </tr> <tr> <td>31</td> <td>80000000_H</td> <td>1 = Diagonal line from bottom left to right top</td> </tr> </tbody> </table>	Bit	Mask	Contents	3-0	0000000F _H	Left line style (→3.10)	7-4	000000F0 _H	Right line style (→3.10)	11-8	00000F00 _H	Top line style (→3.10)	15-12	0000F000 _H	Bottom line style (→3.10)	22-16	007F0000 _H	Colour index (→6.70) for left line colour	29-23	3F800000 _H	Colour index (→6.70) for right line colour	30	40000000 _H	1 = Diagonal line from top left to right bottom	31	80000000 _H	1 = Diagonal line from bottom left to right top
Bit	Mask	Contents																											
3-0	0000000F _H	Left line style (→3.10)																											
7-4	000000F0 _H	Right line style (→3.10)																											
11-8	00000F00 _H	Top line style (→3.10)																											
15-12	0000F000 _H	Bottom line style (→3.10)																											
22-16	007F0000 _H	Colour index (→6.70) for left line colour																											
29-23	3F800000 _H	Colour index (→6.70) for right line colour																											
30	40000000 _H	1 = Diagonal line from top left to right bottom																											
31	80000000 _H	1 = Diagonal line from bottom left to right top																											
14	4	<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>6-0</td> <td>0000007F_H</td> <td>Colour index (→6.70) for top line colour</td> </tr> <tr> <td>13-7</td> <td>00003F80_H</td> <td>Colour index (→6.70) for bottom line colour</td> </tr> <tr> <td>20-14</td> <td>001FC000_H</td> <td>Colour index (→6.70) for diagonal line colour</td> </tr> <tr> <td>24-21</td> <td>01E00000_H</td> <td>Diagonal line style (→3.10)</td> </tr> <tr> <td>31-26</td> <td>FC000000_H</td> <td>Fill pattern (→3.11)</td> </tr> </tbody> </table>	Bit	Mask	Contents	6-0	0000007F _H	Colour index (→6.70) for top line colour	13-7	00003F80 _H	Colour index (→6.70) for bottom line colour	20-14	001FC000 _H	Colour index (→6.70) for diagonal line colour	24-21	01E00000 _H	Diagonal line style (→3.10)	31-26	FC000000 _H	Fill pattern (→3.11)									
Bit	Mask	Contents																											
6-0	0000007F _H	Colour index (→6.70) for top line colour																											
13-7	00003F80 _H	Colour index (→6.70) for bottom line colour																											
20-14	001FC000 _H	Colour index (→6.70) for diagonal line colour																											
24-21	01E00000 _H	Diagonal line style (→3.10)																											
31-26	FC000000 _H	Fill pattern (→3.11)																											
18	2	<table border="1"> <thead> <tr> <th>Bit</th> <th>Mask</th> <th>Contents</th> </tr> </thead> <tbody> <tr> <td>6-0</td> <td>007F_H</td> <td>Colour index (→6.70) for pattern colour</td> </tr> <tr> <td>13-7</td> <td>3F80_H</td> <td>Colour index (→6.70) for pattern background</td> </tr> </tbody> </table>	Bit	Mask	Contents	6-0	007F _H	Colour index (→6.70) for pattern colour	13-7	3F80 _H	Colour index (→6.70) for pattern background																		
Bit	Mask	Contents																											
6-0	007F _H	Colour index (→6.70) for pattern colour																											
13-7	3F80 _H	Colour index (→6.70) for pattern background																											

7 Drawing Objects, Escher Layer

2do

8 Charts

2do

9 PivotTables

2do

9.0.1 Record Order in a PivotTable Cache Stream

•	SXDB
○	SXDBEX
••	• SXFIELD
	○ SXDBTYPE
	SXDOUBLE
	SXBOOLEAN
	•• SXERROR
	•• SXSTRING
	SXDATETIME
	SXEMPTY
••	SXINDEXLIST
•	EOF

9.0.2 Record Order in a PivotTable

•	SXVIEW
••	• SXVD
	•• SXVI
	○ SXVDEX
•	SXIVD
○	SXIVD
○	SXPI
○○	SXDI
•	SXLI
•	SXLI
•	SXEX

10 Change Tracking

2do