File Format Guidelines

Summary

This document contains an operational description of the knowledge levels as defined in the preservation plan. The knowledge levels are prerequisite for implementing full preservation in the future. The three knowledge levels (stored, identified, known) as introduced in the preservation plan will be defined on a technical level here.

The purpose of this document is to provide an operational description of certain strategic principles contained in the preservation plan. It also serves to inform the designated community under which conditions file formats are preserved.

Knowledge levels

Our digital repository contains a great number of different file formats[1], but currently verification and idenfication of these formats happens on a limited scale. For instance, when a number of Excel files was analysed in the course of a project a lot of files happened to be non-Excel formats or they contained file format errors. This finding highlighted the importance of identifying and verifying the file formats in our collection in a more systematic and consistent way.[2]

To enhance our information on file formats they must be identified, checked and verified. Also technical metadata must be extracted and stored. To implement these processes more knowledge on file formats is needed. Because of the large amount of different file formats, it is not conceivable that can be achieved at once. This is why the knowledge levels provide an approach to enhance the required knowledge on file formats in stages. In this way clear information can be provided to the designated community on what file formats are at which stage in the process. In practice some file formats may never reach the ‘known’-level because this is not a requirement for all file types.

The processes needed to enhance knowledge about file formats can be considered separately from the regular checks that happen during the ingest phase. For instance a file format that is on the lowest level of ‘stored’ will still get a 0-byte check and a verification based on checksum. These checks are part of the bit preservation level which is the minimum for all file formats. The

[1] There are about 1130 file extensions

[2] File formats are for the most part identified based on file extension, but this is unreliable. A better way of identifying files is by using ‘signatures’. A signature is a data pattern within the file itself that can be used to determine the format. Tools are available for identifying files based on signature.

level of ‘known’ file format is required for the preservation level of ‘full preservation’. This means in practice that more elaborate checks are performed on certain file formats. For a ‘known’ file format validation software is used to identify files that may contain errors. This does not mean we don’t store these files, but information on errors found is stored along with the file. This information will inform preservation watch and risk analysis processes. In this way research can be done on actions needed to guarantee long term preservation.

Enhancing the knowledge on file formats will thus be done in stages. Some file formats may be identified without being validated. When all the conditions related to a certain knowledge level are met, the appropriate knowledge level will be assigned to a certain file format. This status will be stored for each file format in the repository. In this way it will also be defined which steps need to be performed to be able to move on to the next level. Which conditions are appropriate for a certain knowledge level is detailed below.

1. Knowledge level ‘stored file format’

‘Stored’ file formats are only checked for bit corruption by verifying the checksum. Little is known about a file format at this stage since the file format is not identified.[3] There is no formal identification done using software like Droid Also no PRONOM ID is stored for file formats at this stage. A MIME-type may be known but this is not considerd reliable for identification purposes.

2. Knowledge level ‘identified file format’

An file format can be considered ‘identified’ if a PRONOM ID is assigned to the file. This also means a tool is available that can identify the specific file format and link it to the PRONOM register.
Identification using PRONOM is considered a best practice within the digital preservation community. This is because a PRONOM ID gives more specific[5] information about a file than just the MIME-type. This information is essential for long term preservation which is why it can be considered the baseline for identification.

This means it can be considered the first step in gaining control of the file formats for long term preservation, though it is not enough to be able to perform ‘full preservation’. It is in-between bit preservation and full preservation and can therefore be considered bit preservation+.

3. Knowledge level ‘known file format’

[3] File format extension is not considered reliable for identification purposes.

[5] For instance if the file format is PDF versus if the file format is PDF version 1.4

A ‘known’ file format can be fully preserved since this entails identification, validation and extraction of technical metadata. This information combined will serve as input for interpreting the files so guidelines can be written documenting risks and mitigating actions, if applicable.

Raising the level from ‘identified’ to ‘known’ will be done in stages. In some cases technical metadata extraction may be done without validation. This means the file format will still be considered ‘identified’ and not ‘known’.

The conditions for assigning the highest level of ‘known’ are:

A. All the conditions that apply to the ‘identified’-level
B. Technical metadata extraction
C. Validation of the file format
    a. Classification of the output of the validation software

D. Documentation of the file format containing at least:
    a. Software environment
    b. Readability of the file format on premise and browser accessibility for formats that are publically available
    c. Part of Technology watch
    d. Minimal file size is known
    e. Risk analysis is being done on the file format with a focus on long term preservation

If all this information is available, the file format can be considered preserved for the long-term. This also means all the conditions are met for being able to perform ‘full’ preservation.

Compressed archive formats

When considering the above approach of knowledge levels special attention needs to be given to compressed packaging formats[4], like zip files. These file formats are used to limit the size needed for storage. Currently files in these packages are not extracted but stored in the form in which they are received. Since it is harder to analyse files stored in packages, in the future we will extract files and store them as separate files instead of storing them in a packaged file.

[4] See addendum 2 for a list of compressed package formats within the digital repository

Addendum 1: List of stored, identified and known file formats

Preface

This addendum contains a list of file formats within the repository categorized according to current knowledge level.

Known file formats

There are currently no file formats that fulfill all the conditions required for this knowledge level.

Identified file formats

There are currenly no file formats that completely fulfill the conditions of the ‘identified’ level because assigning a PRONOM identifier is not yet possible in the system.

There are a number of file formats that we have experience with:

File format	**Conditions**			
	Identification	**Technical metadata extraction**	**Validation**	**Documentation**
pdf	Yes, Version	No	No	No
ePub	Yes, Version	Yes, epubcheck	Yes	Draft guidelines on validation errors
jp2	No	Yes, jplyzer	Yes, jplyzer	Information available from the R&D department

Stored file formats

The list below contains information on file formats stored in the repository, based on file format extension. This is considered an inreliable identification method which is why this list may contain errors. After migration to our new repository this list will be updated after identifying the files using a better method which will result in a more reliable list.

Research based on this list has been done by Johan van der Knijff.[5]

gif	xml	jpg	csv	tif	doc	arc
sml	raw	oa3	htm	wav	mp3	docx
txt	bmp	swf	xls	lmx	zip	class
epq	js	mp4	xlsx	rtf	suppl	mov
png	abg	avi	ppt	dat	mpg	aif
cab	page	exe	pptx	dll	ini	dib
x32	db	fig	inf	drv	phd	asc
swa	onx	kmz	ani	dxr	css	eps
vxd	eks	format	tar	lcp	epv	icns
flo	kml	bmr	wmv	hed	bak	bin
jar	pge	pm	api	flp	template	pic
ico	mht	hdr	cat	lst	fba	gz
fb0	wri	cxt	mno	px	al	nls
val	pfb	pfm	tst	grf	xg0	yg0
cct	abt	ex_	hlp	log	lib	ins
grx	xg1	yg1	mol	ps	utx	xtu
pdb	rar	ddd	did	cer	nld	dig
wma	mdb	pl	pdd	ttf	nwk	dcr
dbf	new	wld	ccj	fbd	avx	fas
inx	h	dir	psd	r	boot	cdx
pkg	stp	pod	cnt	fpt	dl_	epm
wassessment	wquestion	ocx	bundle	svg	config	id
vus	dxf	idx	mb	cst	pdx	plist

[5] Johan van der Knijff, 2015 en 2016, Quickscan file formats

gid	tag	tre	lng	vbx	htc	fbf
lid	a4r	fbe	sgm	url	xg2	yg2
lic	std	fasta	rsd	rst	bat	lsr
syd	syx	cmp	msf	nex	tgz	m
bs	dwg	exp	x	msi	ndx	tpl
m1v	fla	x16	bed	trn	z	tar
plc	scm	bz2	000	cfg	dbr	bik
thd	cif	grp	tlb	hw4	tab	ddb
fbc	properties	hwn	otf	xg3	yg3	fbg
stc	fast	tbk	tsv	1	pse	apl
reg	iss	ent	jbf	mmm	chm	fmt
xg4	yg4	dot	asm	m4v	wmz	cnf
ods	fbp	lb0	ldb	c	fbs	src
xg5	yg5	aicon6	dbd	dic	qtc	mst
enc	lix	lsf	cs	wpl	mpp	phy
pps	qdat	vmp	xg6	yg6	fa	mso
aln	cys	f6p	rcm	fbj	fst	lnk

py	xrf	cdt	fdt	ifp	l01	l02
nib	n01	n02	pft	conf	dah	dal
cyto	f6s	oc_	xg7	xg8	yg7	yg8
dtd	dxx	init	its	prj	and	ctf
cur	lba	wcm	wpd	cac	hts	prc
a05	a09	sif	bag	dbk	ibk	old
ssh	xg9	xpi	yg9	a07	fbx	gtr
mml	pir	psp	rm	wrd	xsl	a01

a10	enl	fot	gpr	inc	nav	olb
sas	wig	fbv	map	msg	01	a08
di_	dil	fbu	mdl	nb	sfk	cgi
daw	fbm	hqx	md8	rsrc	six	tan
tbl	u32	xib	acv	cpr	csp	fbh
gpk	ipj	ix	mnu	mop	out	rcs
sps	wmf	xga	xgb	xgc	xgd	xge
yga	ygb	ygc	ygd	yge	7z	atr
ex2	rt_	asf	bngl	btr	cyp	dmg
dsp	fbn	f60	hsc	lck	mat	nxs
qt	zdt	386	art	as	dict	dis
emf	ind	mac	par	res	sdf	tmb
trt	typ	xmo	xsd	arv	bst	ckb
ckw	cty	do	epb	fam	fo_	fts
hmm	idl	lex	md_	mdx	ovr	sid
sld	tit	tt_	tv	vb_	wip	afa
apm	a03	cp2	dct	dlx	fby	fit
fon	fpr	frm	gff	gtf	hpf	ihe
irt	isu	oud	sta	the	vst	v12
wdz	xyz	9	bdef	btl	en1	en3
fbb	fbi	fdb	fna	gpd	mcd	ohe
ort	owl	ply	ptt	rdata	r32	sbk
shs	ssi	sto	tex	tfw	toc	uc
w32	xgf	ygf	118	abs	au	bas
cel	chr	cmd	cpp	cwt	en5	fbq
fbr	flt	fsa	f6q	gml	hsh	hwt

img	lsp	mca	mid	mom	msk	net
noi	opx	osx	ppz	shlb	shlib	smi
tl_	txtn	vbs	vdb	2	3	4
warc	aa	acm	ac1	ac3	addi	akc
ande	bkg	cdi	cfm	cga	cpl	cps
cp1	dab	dbe	dc1	dc3	dft	d2i
d32	ega	eml	fbt	fbw	fbz	f6r
gbf	gp	he0	he2	he4	his	hr2
hw2	iaa	ic1	ic2	ic3	idb	idh
i2n	java	ld	lo	lo1	lo3	mega
mod	mrg	mswm	mw1	mw2	mx	_n_
ngm	null	nw1	obj	oc1	oc3	odp
pag	pcx	pem	prm	pub	pzl	rdat
read	ref	rsm	sbml	sfl	sh	spf
srg	thb	thh	thw	tnt	tree	ttx
twl	tw1	tw2	ufl	uni	vga	wa_
wp5	xtr	0	ali	bbk	blb	bsc
cdr	daf	dbz	ddf	dls	dwt	end
esp	fld	frx	gb	ghe	gms	gmx
gps	grt	hl_	hli	hwa	in_	isk
kix	ls	lvl	lxt	meg	mmp	mpd
mpl	mws	m8cs	nec	nfo	obd	ol_
plm	pp1	re_	rsr	seq	spk	sql
start	stockholm	sys	x86	eps	nwk	3gp
38_	adm	ahe	alb	all	ana	asp

gz	bhe	bloc	bmf	bmk	bnm	bok
brt	bsp	b6p	cdb	ckt	classic	de_
dendro	diff	drl	dsw	dta	edm	eng
expression	faa	fastq	fbk	fda	fdi	fpi
ftp	gct	ged	gex	gi_	gpml	graphml
hhc	hhk	hhs	hta	htv	hwi	hwr
hwy	idp	jp_	jrs	jsp	jtv	krl
krs	layout	lp	lwf	mad	mail	man
matr	~mc	meta	mhe	mm	mpc	mrt

mta	mts	mw	mzid	m12	nexml	obo
odc	ode	odi	odt	off	opa	_pa
paml	path	ped	pgt	ph	phylip	pif
plt	pml	pot	ppsx	qda	rel	rxc
sav	scop	slb	slw	smm	sq1	stf
stx	sty	tdt	td1	toe	trl	trx
types	vcf	vgw	vmo	web	wrl	w02
w03	xa	xfasta	xlsm	01	64	aaa
aab	accdb	ace	ade	ado	af2	ai
amu	apf	apr	apr2	aps	apt	arb
asu	asx	aw	bash	bbb	bio	blib
brd	brh	brl	cath	ccd	cf_	ch_
chip	clg	cli	cls	clus	cm_	cmdx
cmmcd	cmt	cod	col	com	core	cp
cpd	cr	ctm	cvb	c21	c32	daa
dac	dad	dae	dai	daj	data	dau

dav	dax	day	daz	dba	dbb	dbc
dbg	dbh	dbi	dbj	dbl	dbm	dbn
dbo	dbp	dbq	dbs	dbt	dbu	dbv
dbw	dbx	dby	db1	db4	db5	dca
dcb	dcc	dcd	dcl	def	del	_dh
dia	_dj	_dl	drw	dr1	dr4	dr5
ds	dsj	dsl	dtx	dum	dump	_dv
egt	embl	emz	en_	en2	en7	epg
eug	ext	fat	fbo	fdx	flg	flx
fnd	fnt	fpkm-tracking	functions	f4v	gal	gbk
gdb	gdl	genbank	gens	gff3	gls	gmt
gnum	gobpm	gosfunction	gpc	gvw	hdb	hi_
hst	hs6	htmla	hyp	ia	ias	ijm
ion	is	jac	jjj	key	kgg	kin
kit	kls	lbb	ld_	lhs	lisp	list
local	lsrh	lx1	lx2	lzh	mak	manifest
mase	ma4	mbo	mbp	mbq	mbr	mbs
mbt	mbu	mbv	mbw	mbx	mby	mbz
mcb	mcc	mch	mct	me	mfw	mmd
mng	mnt	mor	mpg4	ms_	msb	msh
mus	mxp	nab	name	nbp	newi	nex
nexu	nexus	nh	nhx	nlog	nlogo	no_
not	npr	nt_	ntt	nvw	nw	n3
obi	odb	ol2	opf	pak	pau	pct
perl	pgn	phb	phyloxml	phyt	pict	pid

pk	pop	pos	ppj	prf	prt	ptn
pux	p01	qst	qtif	qtx	raw1	rda
rlx	rnw	rpt	scb	scc	sdq	seq58
sit	slv	ss	stl	str	_sv	sxw
sym	syt	s1	tage	tbr	tcl	td_
tde	tdf	td2	tnc	tok	top	tsk
tt	twf	tx_	_u_f_l	uml	usn	utd
vb	vdp	voi	vxr	wll	wpr	wtr
xgmml	xmfa	zon	ztz	FEXT	003	004
mol2	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	5	6	7	8	pse
onx	kmz	kml				

Addendum 2: compressed package formats

zip
rar
7z tar
gz
lzh
bz2
tgz
sit