ACCESSDATA SUPPLEMENTAL APPENDIX

Registry Offsets

REGF BLOCK OFFSETS

Offsets	Description	Comment
0–3	regf header	0x72656766 – regf
12–19	Date/time of modification	64-bit Windows date/time stamp
48–	Path and filename	
508–511	XOR checksum	Checksum of data in the sector

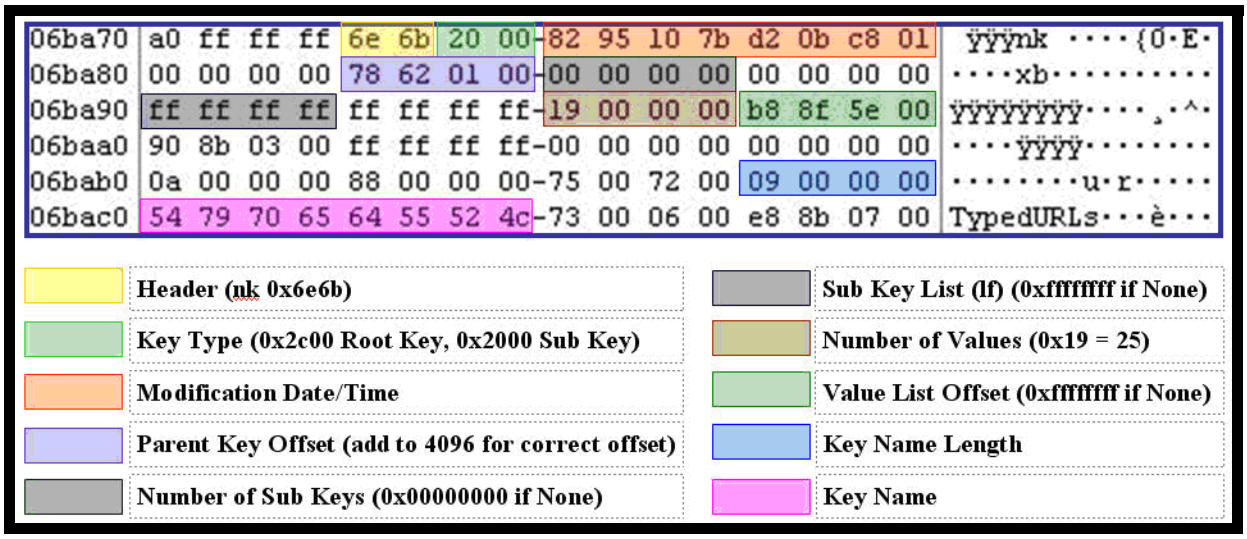
HBIN HEADER BLOCK OFFSETS (FIRST 32 BYTES)

Offsets	Description	Comment
0–3	hbin header	0x6862696e – hbin
4–7	Pointer to first hbin block	0x00100000, 0x00200000, 0x00300000, and so on
8–11	Pointer to next hbin block	Always 0x00100000
20–27	Date/time of modification	64-bit Windows date/time stamp

NK CELL (KEY NODE) OFFSETS

Offsets	Description	Comment
0–3	Entry length (header)	Stores cell size in negative number
4–5	Cell type	nk header – 0x6e6b
6–7	Key type	0x2c00 = Root Key, 0x2000 = Subkey
8–15	Date/time of modification	64-bit Windows date/time stamp
20–23	Offset to parent	
24–27	Number of subkeys	

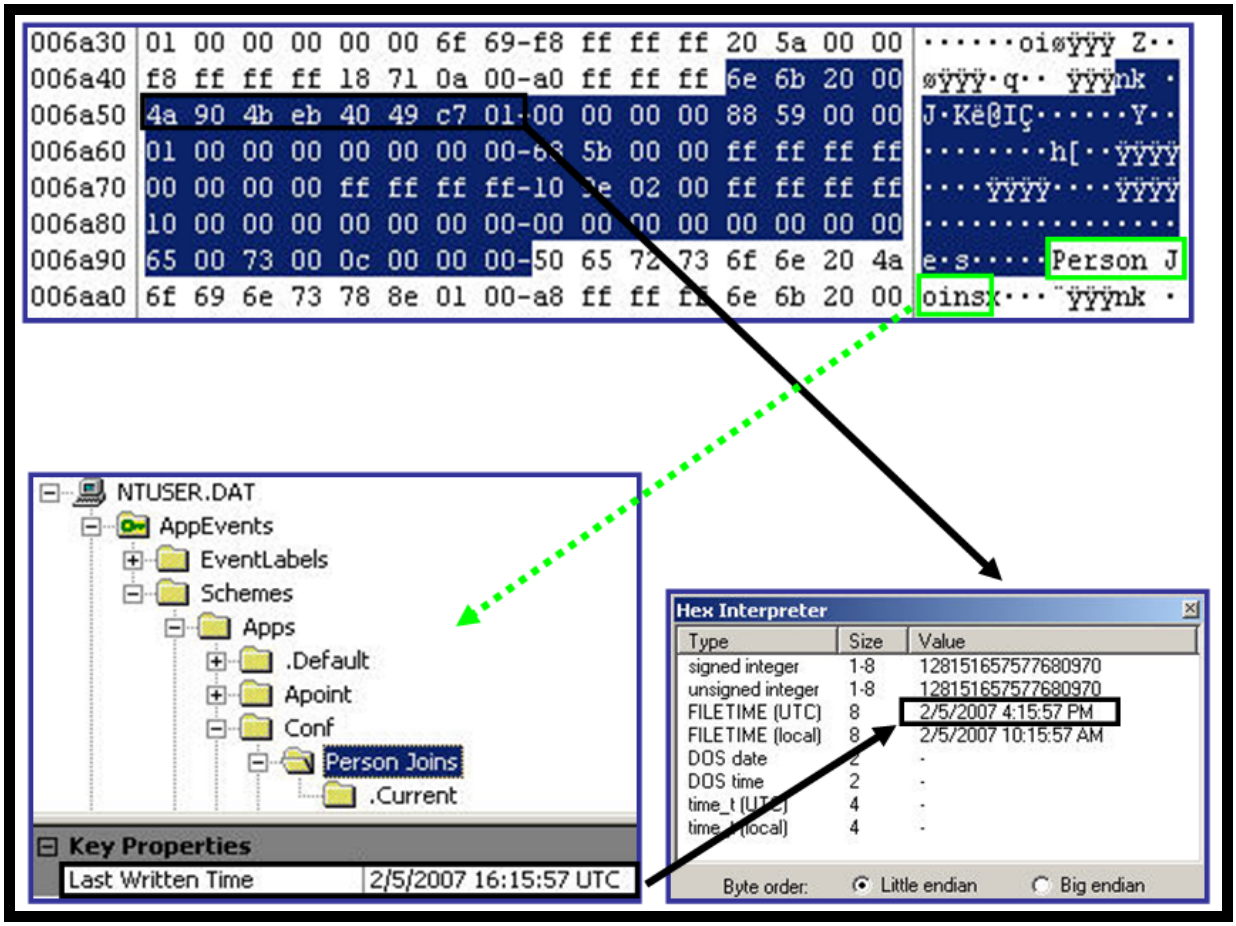
Offsets	Description	Comment
32–35	Subkey list (lf/lh)	If none present = 0xffffffff
40–43	Number of values	
44–47	Offset to value list	Add 4096 / if none = 0xffffffff
48–51	Permissions offset	sk header
52–55	Class entry offset	If none present = 0xffffffff
76–77	Key name length	
80–	Key name	Variable length



Regular Expression to Locate NK Headers/Data

This regular expression looks for key names preceded by the nk header and carves them out. It will highlight from the header to the beginning of the key's name.

Registry Cells=nk[\x2c|\x20]\x00.{7}\x01.{64}



LF Header Offsets—Subkey Lists)

Offsets	Description	Comment
0–3	Entry length	0x6862696e – hbin
4–5	lf header	0x6c66 or 0x6c68 (lh = XP) XP uses an "lf" header in Default, Software, System, and Userdiff hives. XP also uses a hash to identify the name through a lookup rather than using the actual name.

Offsets	Description	Comment
6–7	Number of subkeys	
8–11	Offsets to subkeys	Add 4,096
12–15	First four characters of subkey name	
		Offsets to other subkeys and first four characters will follow for number listed in offsets 6–7

VK Header Offsets

Values can be of two types: a value cell that contains actual data and a value cell that points to data. Values can also be named or unnamed. If no name is assigned to a value, this is the "default" seen in Regedit and Registry Viewer.

Named Value That Contains Data

Offsets	Description	Comment
0–3	Entry length	Stores cell size in negative number
4–5	vk header	0x766b
8–9	Length of data	Dataset size in the value
10–11	Data type	0x0000 = Pointer, 0x0080 = Resident data
16–19	Value type	01 = REG_SZ 02 = REG_EXPAND_SZ 03 = REG_BINARY 04 = REG_DWORD 07 = REG_MULTI_SZ
20–23	Value names present	0x00000000 = No named value 0x01000000 = Named value
24–	Value name	Variable length data

Named Value That Points to Data

Offsets	Description	Comment
0–3	Entry length	Stores cell size in negative number
4–5	vk header	0x766b
8–9	Length of data	Dataset size in the value
10–11	Data type	0x0000 = Pointer, 0x0080 = Resident data
12–15	Offset to linked data	Add 4,096
16–19	Value type	01 = REG_SZ 02 = REG_EXPAND_SZ 03 = REG_BINARY 04 = REG_DWORD 07 = REG_MULTI_SZ
20–23	Value names present	0x00000000 = No named value 0x01000000 = Named value
24–	Value name	Variable length data

Unnamed Value That Contains Data

Offsets	Description	Comment
0–3	Entry length	Stores cell size in negative number
4–5	vk header	0x766b
8–9	Length of data	Dataset size in the value
10–11	Data type	0x0000 = Pointer, 0x0080 = Resident data
12–15	Offset to linked data	Add 4,096
16–19	Value type	01 = REG_SZ 02 = REG_EXPAND_SZ 03 = REG_BINARY 04 = REG_DWORD 07 = REG_MULTI_SZ
20–23	Value names present	0x00000000 = No named value 0x01000000 = Named value
24–	Value name	Variable length data

Unnamed Value That Points to Data

Offsets	Description	Comment
0–3	Entry length	Stores cell size in negative number
4–5	vk header	0x766b
8–9	Length of data	Dataset size in the value
10–11	Data type	0x0000 = Pointer, 0x0080 = Resident data
12–15	Offset to linked data	Add 4,096
16–19	Value type	01 = REG_SZ 02 = REG_EXPAND_SZ 03 = REG_BINARY 04 = REG_DWORD 07 = REG_MULTI_SZ
20–23	Value names present	0x00000000 = No named value 0x01000000 = Named value
24–	Value name	Variable length data

VALUE LISTS (NO HEADER)

Offsets	Description	Comment
0–3	Entry length	Stores cell size in negative number
4–7	Offsets to values	There can be multiples

SAM FILE OFFSETS

F Value Offsets

Offsets	Description	Comment
8–15	Date and time of last login	64-bit Windows date/time stamp 0x0000000000000000 if empty
24–31	Password reset date.time	64-bit Windows date/time stamp 0x0000000000000000 if empty

Offsets	Description	Comment
32–39	Expiration date/time	64-bit Windows date/time stamp 0x0000000000000000 if empty or 0xfffffffffffffff7 if empty
40–47	Last failed login	64-bit Windows date/time stamp 0x0000000000000000 if empty
48–51	RID	Relative Identifier portion of the SID
56	Account status/password set	Account Status left nibble: • 0 = Account active • 1 = Account not active Password Set right nibble • 0 = Password required • 4 = Password not set
60–61	Country code	Default 0000, US 0001, Canada 0002
64–65	Invalid login count	
66–67	Login count	

V Value Offsets

Offsets	Description	Comment
12–23	Pointer to login name	12-byte dataset
24–35	Pointer to name	12-byte dataset
35–47	Pointer to comment	12-byte dataset
156–167	Pointer to LAN password hash	12-byte dataset
166–177	Pointer to NT password hash	12-byte dataset

Divide the 12-byte dataset into three sets of four bytes each. The first four bytes is the pointer to the beginning of the designated data plus 204 bytes. The middle four bytes defines the size of the data. The last four bytes are not used. For example, the dataset 0xbc000000400000000000000, is divided up to:

- 0xbc000000 = Pointer to data start (0xbc = 188 + 204 = 392 as the beginning offset)
- 0x04000000 = Size of the data (four bytes)
- 0x00000000 = Not used

GROUP OFFSETS

Group offsets are located at:

SAM\SAM\Domains\Builtin\Aliases\00000###

Example:

Subkey Name: 00000220

Note that the number in this example converts to 544. The numbers are in hex format to identify the particular group.

Offsets	Description	Comment
16–27	Pointer to group name	12-byte dataset
28–39	Pointer to group description	12-byte dataset
35–47	Pointer to group members	12-byte dataset

Divide the 12-byte dataset into three sets of four bytes each. The first four bytes is the pointer to the beginning of the designated data plus 204 bytes. The middle four bytes defines the size of the data. The last four bytes are not used. For example, the dataset 0x980000001c00000000000000, is divided up to:

- 0x98000000 = Pointer to the group name (0xbc = 152 + 52 = 204 as beginning offset)
- 0x1c000000 = Size of the data (28 bytes)
- 0x00000000 = Not used

USER ASSIST OFFSETS

Offsets	Description	Comment
0-3	Session Number	
4-7	Use Count	Begins at 5 so first use will show a 6
8-15	Last launched date and time	64-bit Windows Date/Time Stamp