nParticles cache description

by S. Cieri, C. D'Angelis

100cells
www.100cells.com
info@100cells.com

Maya Cache file format

The Maya Cache file format (*.mc*) is used in Maya for a variety of caching purposes.
Apparently it is the format which Autodesk is aiming to, as Maya's unified standard for caches. Unfortunately, though, it is not yet a unified standard; neither it is properly documented.

nCache for nParticles

nCaches are a sub-class of Maya Caches, which store simulation data for nucleus objects or fluid effects; nParticles are a specific kind of nucleus object.
Within this document, only the file structure of nCaches for nParticles will be described. Such description is to be taken with reserve, as it's merely resulting out of observations (and deductions) of .mc binary files and Python scripts provided by Autodesk.
Most likely, this will be the first step of a wider research, aimed to generalize the format description (so that it might include every kind of Maya Cache).

Format Specification (generalized)

Maya cache consists of an header and an arbitrary number of channels. Each channel stores data points in time (not necessarily regularly spaced, and not necessarily co-incident in time with data in other channels).

On disk, the Maya cache is made up of a XML description file, and one or more data files.
The description file provides an overview of the .mc file; it is (almost) exaustively described within Maya's official documentation, then it won't be discussed here.

The data files contain the actual data for the channels, and can be organized in a single file per cache, or one file per frame. Here the first case will be described, while the latter (which is pretty intuitive, but yet to be checked) will be presented in a next, revised version of this document or a compendium.

The file format allows for the use of an arbitrary number of channels.
Channel names are restricted and case sensitive.
For Maya to work with an .mc file, such file must at minimum contain a Position channel.

All data is in little-endian byte order.

nParticles cache description (one file)

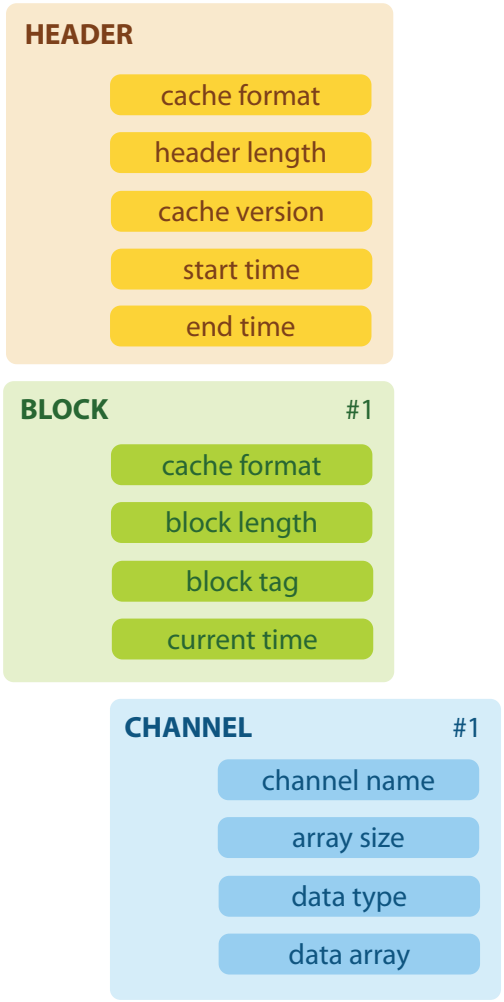
by S. Cieri, C. D'Angelis

100cells
www.100cells.com
info@100cells.com

Format Specification for nParticles, in single .mc file
The multi-frame .mc file format for nParticles consists of:

- A header for general file information.
- A channel definitions section, detailing the channel-wise data included in each particle.
- A block of binary data for the particles.

The time unit used by nCaches is a "tick", which is 1/6000 seconds.



nParticles cache description (one file)

by S. Cieri, C. D'Angelis

100cells
www.100cells.com
info@100cells.com

HEADER

48 bytes

cache format	4 bytes	"FOR4"	char*	cache format
header length	4 bytes	.. ..	(hex) unsigned int	header data length (bytes)
cache version	8 bytes	"CACHVRSN"	char*	tag for cache version
	4 bytes	"00 00 00 04"	(hex) unsigned int	cache version data length (bytes)
	4 bytes	"30 2E 31 00"	(hex)	cache version
start time	4 bytes	"STIM"	char*	tag for start time
	4 bytes	"00 00 00 04"	(hex) unsigned int	start time data length (bytes)
	4 bytes	"00 00 00 00"	(hex) unsigned int	start value**
end time	4 bytes	"ETIM"	char*	tag for end time
	4 bytes	"00 00 00 04"	(hex) unsigned int	end time data length (bytes)
	4 bytes	"00 00 00 01"	(hex) unsigned int	end value**

BLOCK

24 + n bytes

cache format	4 bytes	"FOR4"	char*	cache format
block length	4 bytes	.. ..	(hex) unsigned int	block data length (bytes)
block tag	4 bytes	"MYCH"	char*	tag for channels' block
current time	4 bytes	"TIME"	char*	tag for current time
	4 bytes	"00 00 00 04"	(hex) unsigned int	time data length (bytes)
	4 bytes	.. ..	(hex) int	current time (ticks)
	n bytes	[channels' data, as explained below]		

CHANNEL

28 + n bytes

channel name	4 bytes	"CHNM"	char*	tag for channel name
	4 bytes	.. ..	int (hex)	channel name data length (name length + 1)
	n bytes		char*	channel name, padded with modulus of 4 padding = 4 - len(name)%4
array size	4 bytes	"SIZE"	char*	tag for channel's array size
	4 bytes	"00 00 00 04"	(hex) unsigned int	array size data length (bytes)
	4 bytes	.. ..	(hex) unsigned int	array size (number of elements)
data type	4 bytes		char*	tag for data type
		"DBLA"		double array
		"FVCA"		float vector array (Maya 8.5 and later)
		"DVCA"		double vector array (Maya 8.0 and earlier)
data array	4 bytes	.. ..	(hex) int	array size (bytes)
	n bytes		(hex) float | double	array data

nParticles cache description (one file per frame)

by S. Cieri, C. D'Angelis

100cells
www.100cells.com
info@100cells.com

Format Specification for nParticles, in one .mc file per frame

The single-frame .mc file format for nParticles consists of:

- A header for general file information.
- A channel definitions section, detailing the channel-wise data included in each particle.
- A block of binary data for the particles.

The time unit used by nCaches is a "tick", which is 1/6000 seconds.

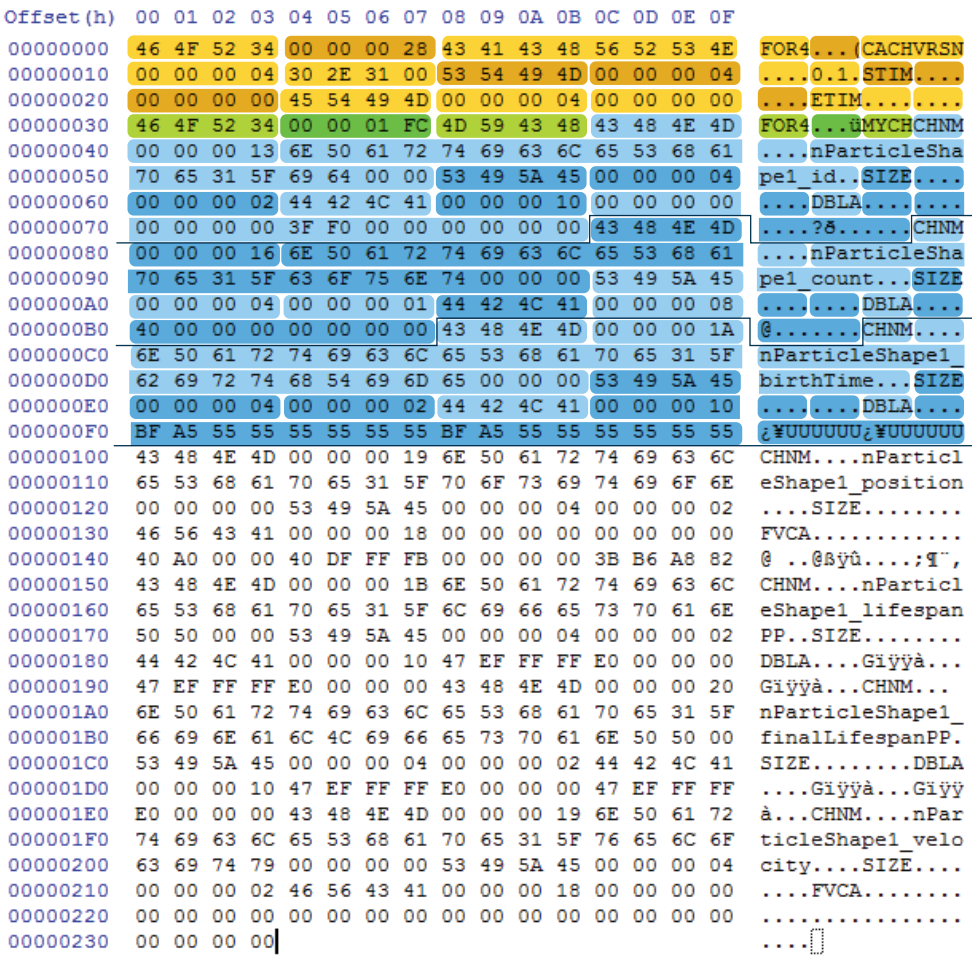
HEADER
- cache format
- header length
- cache version
- start (current) time
- end (current) time

BLOCK
- cache format
- block length
- block tag

CHANNEL #1
- channel name
- array size
- data type
- data array

CHANNEL #2

CHANNEL #3



```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000   46 4F 52 34 00 00 00 28 43 41 43 48 56 52 53 4E  FOR4...(CACHVRSN
00000010   00 00 00 04 30 2E 31 00 53 54 49 4D 00 00 00 04  ....0.1.STIM....
00000020   00 00 00 00 45 54 49 4D 00 00 00 04 00 00 00 00  ....ETIM........
00000030   46 4F 52 34 00 00 01 FC 4D 59 43 48 43 48 4E 4D  FOR4...üMYCHCHNM
00000040   00 00 00 13 6E 50 61 72 74 69 63 6C 65 53 68 61  ....nParticleSha
00000050   70 65 31 5F 69 64 00 00 53 49 5A 45 00 00 00 04  pe1_id..SIZE....
00000060   00 00 00 02 44 42 4C 41 00 00 00 10 00 00 00 00  ....DBLA........
00000070   00 00 00 00 3F F0 00 00 00 00 00 00 43 48 4E 4D  ....?ð......CHNM
00000080   00 00 00 16 6E 50 61 72 74 69 63 6C 65 53 68 61  ....nParticleSha
00000090   70 65 31 5F 63 6F 75 6E 74 00 00 00 53 49 5A 45  pe1_count...SIZE
000000A0   00 00 00 04 00 00 00 01 44 42 4C 41 00 00 00 08  ........DBLA....
000000B0   40 00 00 00 00 00 00 00 43 48 4E 4D 00 00 00 1A  @.......CHNM....
000000C0   6E 50 61 72 74 69 63 6C 65 53 68 61 70 65 31 5F  nParticleShape1_
000000D0   62 69 72 74 68 54 69 6D 65 00 00 00 53 49 5A 45  birthTime...SIZE
000000E0   00 00 00 04 00 00 00 02 44 42 4C 41 00 00 00 10  ........DBLA....
000000F0   BF A5 55 55 55 55 55 55 BF A5 55 55 55 55 55 55  ¿¥UUUUUU¿¥UUUUUU
00000100   43 48 4E 4D 00 00 00 19 6E 50 61 72 74 69 63 6C  CHNM....nPartic
00000110   65 53 68 61 70 65 31 5F 70 6F 73 69 74 69 6F 6E  eShape1_position
00000120   00 00 00 00 53 49 5A 45 00 00 00 04 00 00 00 02  ....SIZE........
00000130   46 56 43 41 00 00 00 18 00 00 00 00 00 00 00 00  FVCA............
00000140   40 A0 00 00 40 DF FF FB 00 00 00 00 3B B6 A8 82  @ ..@ßÿû....;¶¨,
00000150   43 48 4E 4D 00 00 00 1B 6E 50 61 72 74 69 63 6C  CHNM....nPartic
00000160   65 53 68 61 70 65 31 5F 6C 69 66 65 73 70 61 6E  eShape1_lifespan
00000170   50 50 00 00 53 49 5A 45 00 00 00 04 00 00 00 02  PP..SIZE........
00000180   44 42 4C 41 00 00 00 10 47 EF FF FF E0 00 00 00  DBLA....Gïÿÿà...
00000190   47 EF FF FF E0 00 00 00 43 48 4E 4D 00 00 00 20  Gïÿÿà...CHNM...
000001A0   6E 50 61 72 74 69 63 6C 65 53 68 61 70 65 31 5F  nParticleShape1_
000001B0   66 69 6E 61 6C 4C 69 66 65 73 70 61 6E 50 50 00  finalLifespanPP.
000001C0   53 49 5A 45 00 00 00 04 00 00 00 02 44 42 4C 41  SIZE........DBLA
000001D0   00 00 00 10 47 EF FF FF E0 00 00 00 47 EF FF FF  ....Gïÿÿà...Gïÿÿ
000001E0   E0 00 00 00 43 48 4E 4D 00 00 00 19 6E 50 61 72  à...CHNM....nPar
000001F0   74 69 63 6C 65 53 68 61 70 65 31 5F 76 65 6C 6F  ticleShape1_velo
00000200   63 69 74 79 00 00 00 00 53 49 5A 45 00 00 00 04  city....SIZE....
00000210   00 00 00 02 46 56 43 41 00 00 00 18 00 00 00 00  ....FVCA........
00000220   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000230   00 00 00 00                                      ....
```

nParticles cache description (one file per frame)

by S. Cieri, C. D'Angelis

100cells
www.100cells.com
info@100cells.com

HEADER

48 bytes

cache format	4 bytes	"FOR4"	char*	cache format
header length	4 bytes	.. ..	(hex) unsigned int	header data length (bytes)
cache version	8 bytes	"CACHVRSN"	char*	tag for cache version
	4 bytes	"00 00 00 04"	(hex) unsigned int	cache version data length (bytes)
	4 bytes	"30 2E 31 00"	(hex)	cache version
start time	4 bytes	"STIM"	char*	tag for start time
	4 bytes	"00 00 00 04"	(hex) unsigned int	start time data length (bytes)
	4 bytes	.. ..	(hex) unsigned int	current frame**
end time	4 bytes	"ETIM"	char*	tag for end time
	4 bytes	"00 00 00 04"	(hex) unsigned int	end time data length (bytes)
	4 bytes	.. ..	(hex) unsigned int	current frame**

BLOCK

12 + n bytes

cache format	4 bytes	"FOR4"	char*	cache format
block length	4 bytes	.. ..	(hex) unsigned int	block data length (bytes)
block tag	4 bytes	"MYCH"	char*	tag for channels' block
	n bytes	[channels' data, as explained below]		

CHANNEL

28 + n bytes

channel name	4 bytes	"CHNM"	char*	tag for channel name
	4 bytes	.. ..	int (hex)	channel name data length (name length + 1)
	n bytes		char*	channel name, padded with modulus of 4 padding = 4 - len(name)%4
array size	4 bytes	"SIZE"	char*	tag for channel's array size
	4 bytes	"00 00 00 04"	(he x) unsigned int	array size data length (bytes)
	4 bytes	.. ..	(hex) unsigned int	array size (number of elements)
data type	4 bytes		char*	tag for data type
		"DBLA"		double array
		"FVCA"		float vector array (Maya 8.5 and later)
		"DVCA"		double vector array (Maya 8.0 and earlier)
data array	4 bytes	.. ..	(hex) int	array size (bytes)
	n bytes		(hex) float | double	array data