

# Release Notes



## Havok Animation

**Version : 6.0.0 RC1**

*including all development since version 5.5.0 RC1*

---

### Animation Demos

---

#### Demos

---

<b>HKA-595</b>	Implemented	<b>Optimization for multiple skins on the same rig.</b>	6.0.0 Beta2
----------------	-------------	---	-------------

The 'Move Meshes Into Skeleton Space' is a new checkbox option in the Havok Content Tools CreateSkin filter. It will transform the mesh vertices (and all associated data - normals, binormals etc.) into the space of the root bone of the skeleton before the binding matrices are calculated. This will mean that the set of 'meshToBone' transforms will be identical for each mesh binding (or independent of the mesh) on export, in the case that you are exporting multiple mesh bindings. Thus at runtime the 'meshToWorld' transforms required for skinning will need to be computed only once for a given pose, and can be reused by all meshes, which may be more efficient.

---

<b>HKA-999</b>	Implemented	<b>Demo illustrating motion markup outside of behavior.</b>	6.0.0 RC1
----------------	-------------	---	-----------

A demo illustrating annotations created by the Footstep Analysis Filter has been added under Animation/Api/Playback/FootstepAnalysis

---

<b>HKA-1053</b>	Implemented	<b>Demo illustrating stable physical attachments (e.g. gun belts, backpacks) to animated characters..</b>	6.0.0 RC1
-----------------	-------------	---	-----------

A new demo has been added here:  
Demos\Physics\UseCase\PhysicalAttachments\CharacterAttachments

This shows an animated character with a rigid body ragdoll following the animation to which several examples of 'attachments' have been added. Each example has several variants which show the 'out of the box' solutions's problems (stretching, jitter, energy, excessive motion) and then 1 or 2 steps to reduce that e.g. with damping, or with damping + chain constraints.

---

### Animation Runtime

---



## Bugs

<b>HKA-1016</b> Fixed	<b>DMA alignment exception with float tracks (both sampleonly and sampleandblend).</b>	6.0.0 Beta1
<b>Behavior Change</b>	<p>Invalid DMAs were being performed if the number of float tracks was not a multiple of 4.</p> <p>N.B. The float tracks output array must now be of size equal to a multiple of 16 bytes in order for the DMA of the results to this array to be performed.</p> <p>The hkaPose class has been modified to 'overallocate' (set possibly larger capacity) to ensure this, so users using this class need make no changes.</p> <p>However, users who allocate their own output arrays must ensure that they allocate enough space for the DMA. The helper functions: <code>hkaMultithreadedAnimationUtils::allocateFloatSlotsArrayRoundedUpToMultipleOf16()</code> and <code>hkaMultithreadedAnimationUtils::deallocateFloatSlotsArray()</code> may help the user do this.</p>	
<b>HKA-924</b> Fixed	<b>Add interface to clear cache keys for skeletal animation</b>	6.0.0 RC1
	<p>The keys used for cache access use the address in memory of the animation and hence may not actually be unique over time if animations are moved in memory or deleted/loaded. A new helper method <code>hkaAnimation::clearAllCacheKeys(hkaChunkCache* cache)</code> has been added to help 'unload' an animation instance fully from the cache.</p>	
<b>HKA-982</b> Fixed	<b>Remove default constructors for <code>hkaSampleAnimationJob::AnimationData::AnimationData()</code> and <code>hkaAnimationSampleAndCombineJob::ControlData::ControlData()</code> on SPU.</b>	6.0.0 Beta1
<b>Behavior Change</b>	<p>This permits the user who is implementing their own SPU animation jobs to make these members of a class with virtual functions to be DMA'd to the SPU and (since the in-place constructor can be called safely on the SPU without overwriting the data in these classes).</p>	
<b>HKA-1020</b> Fixed	<b><code>hkaPose::syncLocalSpace()</code> can fail under certain circumstances.</b>	6.0.0 Beta1
<b>Behavior Change</b>	<p>Previously <code>syncLocalSpace()</code> assumed that bones with dirty local space (clean model space) will also have parents with clean model space. This was not guaranteed to be true if <code>pose.setBoneModelSpace( *, *, hkaPose::PROPAGATE )</code> was called. This has now been fixed so that <code>syncLocalSpace()</code> will sync the parent's model space if required.</p>	

<b>HKA-1025</b> Fixed	<b>hkaMirroredAnimation::samplePartialWithDataChunks() is not multithreaded-safe.</b>  This is now safe (and moreover the default job processing in a multithreaded environment does not actually call the 'chunks' version of the sampling).	6.0.0 Beta2
<b>HKA-1044</b> Fixed	<b>Local-to-model conversion of partially sampled animations on the SPU will leave unsampled bones uninitialized.</b>  The sampling, blending and conversion to model was done only on the max bones specified (the partial sample set), leaving any other bones in the output destination pose untouched (probably uninitialized). A new member m_numSkeletonBones has been added to hkaAnimationSampleAndCombineJob which is set by default to the number of bones in the skeleton. Now the code on SPU will always fill in the local the reference pose for all unsampled bones in the range 0 to m_numSkeletonBones-1 , perform the local to model conversion and DMA over all m_numSkeletonBones bones. This ensures the default behavior results in no uninitialized bones or allows the user to set m_numSkeletonBones to m_maxBones if they want the old behavior.	6.0.0 RC1
<b>HKA-784</b> Fixed  <b>Behavior Change</b>	<b>hkPose::makeAllChildrenLocalSpace() doesn't correctly handle skeletons with multiple roots.</b>  Now fixed.	6.0.0 Beta1
<b>HKA-967</b> Fixed	<b>Unused methods in hkaSplineSkeletalAnimation needed when linking for SPU.</b>  The linker was confusing methods not needed. These have now been #ifdef'd out for this build.	6.0.0 RC1
<b>HKA-759</b> Fixed  <b>Behavior Change</b>	<b>hkSkeletonUtils::getDescendants() doesn't work on the root bone.</b>  Now works for any roots.	6.0.0 Beta1

## New Features

<b>HKA-1001</b> Implemented	<b>Create motion markup infrastructure to support footfall analysis</b>  The Footstep Analysis Filter has been added to the content tools. The results are displayed in the preview window.	6.0.0 Beta2
-----------------------------	---	-------------

---

**HKA-1023** Implemented **Add CPU versions of animation jobs.** 6.0.0 Beta2

hkaSampleAnimationJobs and hkaAnimationSampleAndCombineJobs can now be processed on the CPU. This allows multithreaded platforms to use the existing Havok job queue for animation jobs without any additional work. See the SampleAndBlendMultithreadingDemo and SampleOnlyMultithreadingDemo for examples.

## Improvements

---

**HKA-657** Implemented **hkaAnimationBinding, hkaMeshBinding and hkaBoneAttachment should support associations to each other by name.** 6.0.0 Beta1

These 3 classes now have the member variable const char\* m\_originalSkeletonName which is set by the Content Tools. It can also be set by the user. This allows association of serialized objects with a given skeleton by name.

---

**HKA-877** Implemented **SPU Animation decompression should be able to continuously process jobs.** 6.0.0 Beta2

**Behavior Change** Havok animation job management now allows the user to add jobs asynchronously by setting the wait policy of the job queue to avoid termination when the queue is empty. The user can then use either the old job semaphore or new job flag to be notified of job completion. The use of this new feature is illustrated in the Animation\Api\Multithreading\SampleAndBlend and Animation\Api\Multithreading\SampleOnly demos.

### Interface Change

---

**HKA-998** Implemented **Handle mirroring of annotation tracks** 6.0.0 Beta1

**Behavior Change** hkaMirroredSkeleton now supports the mirroring of annotation tracks. The user may now specify which annotation names map to one another for mirroring. Expected use cases include mapping of a "rightFootDown" annotation to a "leftFootDown" annotation.

### Interface Change

<b>HKA-1002</b>	Implemented	<b>Improve debugging of stack overflow on SPU when decompressing animations and reduce default chunk buffer size.</b>	6.0.0 RC1
<p>There is now a HK_CRITICAL_ASSERT in the stack allocations on SPU which will fire if the stack overflows.</p> <p>Also, the default buffer size was reduced slightly to 70K - Note that this is still conservative and clients should use the <code>hkaAnimation::getMaxSizeOfCombinedDataChunks()</code> for sample-only jobs , or <code>hkaMultithreadedAnimationUtils::getMaxSizeRequiredForSampleAndCombineJobBuffer()</code> and <code>getConservativeMaxSizeRequiredForSampleAndCombineJobBuffer()</code> for sample-and-combine jobs to get a better estimate of how large this actually needs to be.</p>			
<b>HKA-1048</b>	Implemented	<b>Improve ease of mirroring setup</b>	6.0.0 RC1
<p><b>Behavior Change</b> Animation mirroring is now much simpler to setup. Required user inputs are a mapping between "left" and "right" bones, an example symmetric pose (such as the reference pose) and the orientation of the mirror plane in model space.</p> <p><b>Interface Change</b></p>			
HKA-876			
<b>HKA-959</b>	Implemented	<b>Add more PPU-side asserts for invalid/corrupt job data when using SPU sampling or blending.</b>	6.0.0 Beta2
<p><code>isValid()</code> methods have been added to the sampling jobs which will test all data for validity including alignment and size for possible DMAs. In Debug these will assert on invalid data, and in Release they will return a <code>hkBool</code> indicating validity..</p>			
<b>HKA-974</b>	Implemented	<b>Rename animation classes for clarity</b>	6.0.0 Beta1
<p>The following renames have been performed:</p> <p><code>hkaSkeletalAnimation</code>-&gt;<code>hkaAnimation</code>  <code>hkaInterleavedSkeletalAnimation</code>-&gt;<code>hkaInterleavedUncompressedAnimation</code>  <code>hkaDeltaCompressedSkeletalAnimation</code>-&gt;<code>hkaDeltaCompressedAnimation</code>  <code>hkaWaveletSkeletalAnimation</code>-&gt;<code>hkaWaveletCompressedAnimation</code>  <code>hksSplineSkeletalAnimation</code>-&gt;<code>hkaSplineCompressedAnimation</code>  <code>hkaMirroredSkeletalAnimation</code>-&gt;<code>hkaMirroredAnimation</code></p> <p><b>Interface Change</b></p>			

---

<b>HKA-987</b>	Implemented	<b>Ensure hkaAnimationSampleAndCombineJob is consistent between allocate time and create time.</b>	6.0.0 RC1
----------------	-------------	--	-----------

hkaMultithreadedAnimationUtils::allocateSampleAndCombineJob() now allocates to the full size of the number of controls in the animated skeleton and this is checked in hkaMultithreadedAnimationUtils::createSampleAndCombineJob() to ensure the user has not added more controls.  
It is now also possible to reuse the job structure by allocating only once and deleting when the animated skeleton is deleted, assuming that the number of animation controls does not change.

---

<b>HKA-1011</b>	Implemented	<b>hkaSkeletalAnimation::getAnnotations() should be investigated for rigor in handling time intervals (both wrapped and non-wrapped)</b>	6.0.0 Beta2
-----------------	-------------	--	-------------

hkaSkeletalAnimation::getAnnotations() now properly handles time wrapping at the end of an animation (previously an annotation on the last frame could be found twice).

---

<b>HKA-1026</b>	Implemented	<b>hkaSplineCompressedAnimation::recompose should be optimized for PS3</b>	6.0.0 RC1
-----------------	-------------	--	-----------

hkaSplineCompressedAnimation::recompose has been optimized for the PS3 using intrinsic functions.

---

<b>HKA-1031</b>	Implemented	<b>hkaAnimation::getAnnotations() and hkaAnimatedSkeleton::getAnnotations() should allow the user to specify the maximum number of annotations requested</b>	6.0.0 Beta2
-----------------	-------------	--	-------------

The functions hkaAnimation::getAnnotations() and hkaAnimatedSkeleton::getAnnotations() now allow the user to specify the maximum number of annotations requested for efficiency reasons.

### Interface Change

---

<b>HKA-1033</b>	Implemented	<b>SampleAndBlendMultithreadingDemo should be straightforward to add customer assets to.</b>	6.0.0 Beta2
-----------------	-------------	--	-------------

It is now clear how to add customer assets to the SampleAndBlendMultithreadingDemo for testing and timings.

---

<b>HKA-1039</b>	Implemented	<b>Ensure Animation fully supports NULL/Identity bindings.</b>	6.0.0 RC1
-----------------	-------------	--	-----------

As a result of EXP-477 the user may now optionally 'prune' the index arrays in the animation bindings (hkaAnimationBinding objects) on export from the Content Tools if the binding is equal to the identity (track indices map exactly to bone indices). This will set m\_transformTrackToBoneIndices or m\_floatTrackToFloatSlotIndices to HK\_NULL and m\_numTransformTrackToBoneIndices or m\_numFloatTrackToFloatSlotIndices to 0 for transform or float components of the binding respectively. This is supported on the runtime side by identifying this case and using an implicit identity mapping. All user code which previously assumed these array were non-empty will have to handle this case if the user intends to prune bindings on export or at runtime.

---

<b>HKA-1027</b>	Implemented	<b>Add destructor for hkaSampleAnimationJob::AnimationData and hkaAnimationSampleAndCombineJob::ControlData</b>	6.0.0 Beta2
-----------------	-------------	---	-------------

We added empty destructors so that these can be used with hkObjectArrays without compiler warnings.

---

<b>HKA-1028</b>	Implemented	<b>hkaAnimationSampleAndCombineJob::ControlData::isValid() and hkaSampleAnimationJob::AnimationData::isValid() should be const.</b>	6.0.0 Beta2
-----------------	-------------	---	-------------

These methods are now const.

## Optimizations

---

<b>HKA-986</b>	Implemented	<b>Remove some LHS from hkaPose::syncModelSpace() , hkaSkeletonUtils::transformModelPoseToLocal/LocalToModel() and hkaSkeletonMapper::mapPose().</b>	6.0.0 RC1
----------------	-------------	--	-----------

Several LHS from these methods have been eliminated.

---

<b>HKA-993</b>	Implemented	<b>Prefetch mapping information in hkaSkeletonMapper::mapPose()</b>	6.0.0 Beta2
----------------	-------------	---	-------------

We now prefetch the simplemappings as we iterate through them for an expected performance improvement.

---

<b>HKA-995</b>	Implemented	<b>hkaSkeletonMapper::mapPose should call inline version of hkVector4::setTransformedPos().</b>	6.0.0 Beta2
----------------	-------------	---	-------------

The inlined function offers a small performance improvement.

---

<b>HKA-1007</b>	Implemented	<b>Rewrite hkaPose accessors to be less prone to user error.</b>	6.0.0 Beta2
-----------------	-------------	--	-------------

Accessor methods renamed to give clearer indication of what is being synchronized internally with each call.

### Interface Change

---

<b>HKA-1019</b>	Implemented	<b>Remove LHS from hkaSkeletonUtils::getModelSpaceScale() .</b>	6.0.0 RC1
-----------------	-------------	---	-----------

A load-hit-store was removed from this method.

---

<b>HKA-1051</b>	Implemented	<b>hkaSkeletonUtils::normalizeRotations() should normalize 4 quaternions at once.</b>	6.0.0 RC1
-----------------	-------------	---	-----------

This method now uses vector operations to speed up the normalization.

## Compression

---

### Bugs

---

<b>HKA-1035</b>	Fixed	<b>Wavelet and delta endian-swizzling broken for float tracks.</b>	5.5.1 RC1
-----------------	-------	--	-----------

Float tracks are now interpreted properly on architectures of different endian.

### Improvements

---

<b>HKA-1040</b>	Implemented	<b>Add asserts to hkaWaveletSkeletalAnimation::initialize and hkaDeltaCompressedSkeletalAnimation::initialize</b>	6.0.0 RC1
-----------------	-------------	---	-----------

Track Analysis (called by hkaWaveletSkeletalAnimation and hkaDeltaCompressedSkeletalAnimation initialization) now asserts if both absolute and relative tolerances are used simultaneously.

## Controllers

---

### Bugs



<b>HKA-914</b>	Fixed	<b>hkaAnimationControls with less than 8 tracks cause DMA alignment errors on PS3 when used with pool memory.</b>	6.0.0 RC1
Animation controls now ensure that the arrays are oversized to at least 16 bytes in capacity which will ensure that the pool memory manager allocated aligned to 16bytes (allocations less than 16 bytes are only 8-byte aligned by the pool memory manger).			
<b>HKA-997</b>	Fixed	<b>hkaDefaultAnimationControl's easeIn/easeOut method implementation does not match description in Havok Documentation</b>	6.0.0 RC1
hkaDefaultAnimationControl now allows easeIn or easeOut to be called multiple times without resetting. This is the behavior described in the documentation.			

## Documentation

---

### Bugs

<b>HKA-997</b>	Fixed	<b>hkaDefaultAnimationControl's easeIn/easeOut method implementation does not match description in Havok Documentation</b>	6.0.0 RC1
hkaDefaultAnimationControl now allows easeIn or easeOut to be called multiple times without resetting. This is the behavior described in the documentation.			

### Documentation Changes

<b>HKA-988</b>	Implemented	<b>Animation documentation needs more multithreading explanation.</b>	6.0.0 RC1
Havok Animation uses the same task, job and job queue structure shared by other Havok components and this is documented in the Multithreading section of the Animation userguide. The uses of each job type are illustrated in the Animation\Api\Multithreading\SampleAndBlendMultithreadingDemo and Animation\Api\Multithreading\SampleOnlyMultithreadingDemo respectively.			

<b>HKA-983</b>	Implemented	<b>Highlight importance/requirement of constrained pair collision filtering in ragdoll docs.</b>	6.0.0 RC1
<p>setPoseModelSpace() is intended for (re)initialization but not intended to be called continuously during simulation as it can be slow and it will zero all body velocities. To influence the pose of ('drive') an hkaRagdollInstance over time during simulation, use the driveToPose() functionality of the hkaRagdollRigidBodyController, or call hkpKeyFrameUtility::applyHardKeyFrame on each rigid body instead.</p>			

## Ragdoll

---

### Improvements

<b>HKA-697</b>	Implemented	<b>hkRagdollInstance::setPoseModelSpace() should warn if the ragdoll is in the world.</b>	6.0.0 RC1
<p>This function should not be called every frame - it will repose the ragdoll and clear all velocities. For proper ragdoll driving clients should use controllers, and this method now warns about this.</p>			
<b>HKA-984</b>	Implemented	<b>Document hkaRagdollInstance::addToWorld() updateFilter parameter.</b>	6.0.0 RC1
<p>Added reference manual documentation.</p>			