Release Notes

Havok Physics

Version : 6.0.0 RC1

including all development since version 5.5.0 RC1

Character Controller

Bugs

HVK-4417 Fixed **Rigid Body Character Controller tries to remove same vertical contact point twice** 6.0.0 Beta

Fixed a bug in the hkpCharacterRigidBodyListener::contactPointRemovedCallback where the pointer to hkpSimpleContactConstraintAtom was possibly invalid. Now, the proper atom is retrieved and modified.

HVK-4547 Fixed **hkpCharacterRigidBody::checkSupport does not initialize hkpSurfaceConstraintInfo** 6.0.0 Beta

All friction parameters for every constraint are set to 0.0f. hkSimplexSolver is used only for support checking.

Constraints

Bugs

HVK-4535 Fixed **Ragdoll cone limits are not adhered to after limits have been breached** 5.5.1 RC1

The cone limit of the hkpRagdollConstraint wouldn't work after it was breached once. This happened when using the default option of stabilizing the cone limit. This is fixed now.

Demos

Bugs

HVK-4382 Fixed **Public Wii Demo Crashes for 5.5** 6.0.0 Beta

Many demos were fixed on the Wii for Havok 5.5.0.

Documentation

Documentation Changes

HVK-3039	Implemented	**Build configurations are now explained in the manual** The differences between Release, Debug and Fulldebug are now explained in the quickstart guides in the "Linking with Havok" section.	6.0.0 Beta
HVK-3783	Implemented	**Document serialization registration requirements in the 'getting started' platform-specific guides.** A section has been added to the platform-specific guides which explains how to register Havok's classes with the serialization registry.	6.0.0 RC1
HVK-4489	Implemented	**No documentation for the new timestep argument used by hkpCharacterRigidBody::setLinearyVelocity** Documentation for the timestep argument used by hkpCharacterRigidBody::setLinearVelocity has been added to hkpCharacterRigidBody.h	6.0.0 Beta
HVK-4506	Implemented	**hkpSimulation does not appear in reference manual** hkpSimulation was added to the refman.	6.0.0 Beta

General SDK

Bugs

HVK-3972	Fixed	**hkConvexPenetrationUtil::tryToExpandPolytope() crashes if hkWingedEdge::findVisibleTriangles() recurses too often** We created a work around for a rare bug (assert 0x7910923c) in the convex collision detector (hkpWingedEdge::patchInBoundaryToPointOfViewExpansion()).	5.5.1 RC1

New Features

HVK-4179	Implemented	**Gracefully handle when objects are added to the world in a penetrating state** A body can easily cause 10's of TOIs when it's added to the world in a way that it penetrates multiple triangles of a landscape. To prevent that and the resulting huge CPU spikes, one can use the new hkpEasePenetrationAction.	6.0.0 Beta

Improvements

HVK-4188	Implemented	**Prevent strict-aliasing warnings from inlined Havok functions** Havok's headers and inline files no longer causes warnings when they are included in files which have strict aliasing warnings enabled. Note that Havok libraries must still be built without strict aliasing.	5.5.1 RC1

hkcollide

Bugs

HVK-3979	Fixed	**Capsule-capsule collisions can generate bad contact points** Fixed capsule-capsule collision numerical instability on xbox 360 caused by numerical precision of sqrtInverse() in closest point calculation. See more details in comments.	6.0.0 Beta
HVK-4412	Fixed	**hkpListAgent / hkpConvexListAgent fail to compile because of the missing 'const' while overriding calcStatistics** Compile error due to missing 'const' in calcStatistics() declarations. This has been fixed.	6.0.0 Beta
HVK-4509	Fixed	**Welding now works with new collection collection agent** We fixed the welding for the collection collection agent.	5.5.1 RC1
HVK-4526	Fixed	**hkpListShape::castRay implementation broken on SPU** Child shapes are now properly DMAed during raycasting. We have also replaced a virtual call to isCollisionEnabled() with a call to the new custom user collision filter callback (HVK-4533).	5.5.1 RC1

HVK-4533 Fixed **Usage of hkpGroupFilter is hard coded when raycasting on SPU** 6.0.0 Beta

We have added support for user-defined collision filter callbacks. You can register your own custom collision filter callback by using registerCollisionFilterFunctions().

Raycasting on SPU used to always assume that the provided collision filter was an hkpGroupFilter. The code will now call the new custom user callback function instead, giving the user control over what to do with the filter. We also provide a default implementation for this callback that will only query the filter if it is an hkpGroupFilter and return TRUE otherwise.

HVK-4538 Fixed **hkp3AxisSweep::hkpBpNode handle is uninitialized** 5.5.1 RC1

A member of the first hkpBpNode was uninitialized, which could cause an uninitialized AABB to appear when the Broadphase Viewer was enabled in the Visual Debugger.

HVK-4545 Fixed **hkExtendedMeshShape using tranformed or translated children uses non multithreaded safe reference counting** 6.0.0 Beta

We fixed a potentially serious bug, where the hkpExtendedMeshShape used non thread safe reference counting for transformed or translated convex children

HVK-4553 Fixed **Aliasing issue in GCC 4.1.1 breaks the broadphase** 5.5.1 RC1

A compiler bug in GCC 4.1.1 would corrupt the broadphase when adding or removing bodies from the world. We added a workaround for this problem.

HVK-4279 Fixed **hkpCollectionCollectionAgent3 is not used for hkconvexShape vs hkpListShape** 6.0.0 Beta

A new agent hkpCollectionAgent3 is added, and it replaces the old hkpListAgent3. The new agent handles collisions of convex shapes against collection shapes (i.e. collections without BvTree shapes around them). It performs an aabb overlap test to minimize the number of narrowphase collisions tested, just like the hkpCollectionCollectionAgent3 does. Note that the new agent only works with the 'streaming' technology and the old hkpListAgent (non -3) is still used elsewhere.

HVK-4419 Fixed **hkpStorageExtendedMeshShape non deterministic binary image (quick fix)** 6.0.0 Beta

Uninitialized variables (4th element in vertex array and indices between striding) are set to zero during m_meshstorage building.

HVK-4436 Fixed **hkpExtendedMeshShape should not allocate memory when loaded from a packfile** 6.0.0 Beta

This no longer causes an allocation. Subparts now only need to be aligned on a 16byte boundary.

HVK-4484 Fixed **Welded normals have non-unit length** 5.5.1 RC1

Welded normals for triangle shapes that have non-zero values in the .w components of their vertices were incorrectly normalized. These types of triangles are produced by the extended mesh shape. This has been fixed

HVK-4491 Fixed **hkpExtendedMeshShape destructor frees memory twice if loaded from packfile** 5.5.1 RC1

If an hkpExtendedMeshShape was loaded from a packfile and later destroyed, it would incorrectly try to free the child shape arrays of its subparts. This has been fixed. It now frees that memory only if the shape was created at runtime.

HVK-4544 Fixed **hkpExtendedMeshShape::getAaab not registered for integrate elf** 5.5.1 RC1

hkpExtendedMeshShape::getAaab now works on SPU.

HVK-4616 Fixed **hkpAgentNnEntry::m_agentIndexOnCollidable out of hkpLinkedCollidable::m_collisionEntries bounds** 5.5.1 RC1

This was a symptom of corruption in collision structures. The corruption was caused by unguarded access to entities in the fixed island by multiple threads in the split-island job. This is fixed now.

HVK-4626 Fixed **Incorrect normalize in hkpMeshWeldingUtility::calcAngleFromVertices** 5.5.1 RC1

hkpMeshWeldingUtility::calcAngleFromVertices was incorrectly using normalize4 on a vector instead of normalize 3, which could cause incorrect welding results if the w component was uninitialized. This has been fixed.

HVK-4635	Fixed	**hkpExtendedMeshShape::getChildShape uses hkVector4::load4 which may read out-of-bounds memory** Previously hkpExtendedMeshShape::getChildShape could read past the end of valid memory. This has been fixed.	5.5.1 RC1
HVK-3495	Fixed	**Linear casting outside the broadphase causes a crash.** Linear casting outside the world's extents will no longer cause a crash. Note that false misses may be reported for objects which are outside the broadphase; a warning will be triggered in this case.	6.0.0 RC1
HVK-4448	Fixed	**hkpListShape asserts for hkpListShape vs. hkpExtendedMeshShape (no mopp)** The assert 0xf031ed46 in hkpListAgent3::create() would fire wrongly. This is fixed now.	6.0.0 RC1
HVK-4551	Fixed	**hkpMoppCode::m_buildType is not saved when using hkpMoppCodeStreamer::writeMoppCodeToArchive** hkpMoppCodeStreamer now properly saves and loads hkpMoppCode::m_buildType. MOPPs that were saved from older versions of Havok can still be loaded, but should be re-exported so that the flag can be saved properly.	6.0.0 Beta
HVK-4555	Fixed	**hkpTriSampledHeightFieldBvTreeShape::queryAabb with AABB larger than height field extents returns no hits** Previously queryAabb would give incorrect results if the query's aabb was larger than the heightfield's. This has been fixed.	6.0.0 RC1
HVK-4627	Fixed	**hkAabbUint32 should use hkUint8's, not hkUchar's** Fixed hkAabbUint32 class member types, replace hkUchar with hkUint8.	6.0.0 RC1

HVK-4632	Fixed	**hkpConvexVerticesShape::castRay's use of hkSpu4WayCache sometimes gets incomplete DMA request** This was a specific case of COM-422 (hkSpu4WayCache doesn't check the size of DMA request when determining whether to fetch data) , and is fixed as a part of it.	6.0.0 RC1
COM-422			
HVK-4637	Fixed	**Incorrect memcpy in hkpExtendedMeshShape::expandOneTriangleSubparts** hkpExtendedMeshShape::expandOneTriangleSubparts didn't copy all of the subparts when resizing the array. This has been fixed. Note that this only affects users of the 6.0 Beta version.	6.0.0 RC1
HVK-4421	Fixed	**hkpListShape asserts for custom hkpShapeCollections** The assert 0xf031ed46 in hkpListAgent3::create() would fire wrongly. This is fixed now.	6.0.0 RC1
HVK-4580	Fixed	**Fix handling of hkCompressedSampledHeightfieldShape in hkpHavokSnapshot** hkCompressedSampledHeightfieldShape is now handled properly in hkpHavokSnapshot. This problem only affects users of Havok 6.0 Beta 1.	6.0.0 RC1
HVK-4583	Fixed	**hkpTriSampledHeightFieldCollection::getCollisionFilterInfo not implemented** hkpTriSampledHeightFieldCollection::getCollisionFilterInfo is now implemented on SPU.	5.5.1 RC1
HVK-4602	Fixed	**hkpAgentSectorHeader::allocate does not allocate through hkThreadMemory** This is fixed now. The function allocates through hkThreadMemory.	6.0.0 RC1

New Features

HVK-4179	Implemented	**Gracefully handle when objects are added to the world in a penetrating state** A body can easily cause 10's of TOIs when it's added to the world in a way that it penetrates multiple triangles of a landscape. To prevent that and the resulting huge CPU spikes, one can use the new hkpEasePenetrationAction.	6.0.0 Beta
HVK-4550 **Interface Change**	Implemented	**Full support for heightfields on SPU** Tri-sampled heightfields are now supported on the SPU out of the box. To use this feature, you must create an hkpStorageSampledHeightfieldShape or hkpCompressedSampledHeightFieldShape, and wrap it an hkpTriSampledHeightFieldCollection and an hkpTriSampledHeightFieldBvTreeShape. See the HeightfieldsOnSpu demo for an example. Note that the interface for heightfields has changed; the getHeightAt method is now getHeightAtImpl.	6.0.0 Beta
HVK-4365	Implemented	**Heightfield raycasting on SPU** SPU Raycasting against standard heightfield shapes (hkpStorageSampledHeightFieldShape and hkpCompressedSampledHeightFieldShape) is now supported.	6.0.0 Beta
HVK-4391 **Behavior Change** **Interface Change**	Implemented	**Support for welding with hkpTriSampledHeightFieldCollection** Welding information can now (optionally) be stored for hkpTriSampledHeightFieldCollections by calling hkpTriSampledHeightFieldCollection::computeWeldingInfo. Note that the hkpTriSampledHeightFieldCollection must also have an hkpTriSampledHeightFieldBvTreeShape.	6.0.0 RC1
HVK-4604	Implemented	**Handle hkpTransformShapes in AgentMachines and on SPU** hkpTransformShapes are now handled in Agent Machines (as part of streaming-agents technology). They are also handled on SPU's allowing for e.g. mopps of transformed mopps being handled on SPU. Note that on SPU you cannot have more than one hkpTransformShape placed consecutively in the shape hierarchy.	6.0.0 RC1

Improvements

HVK-4574	Implemented	**Remove hkpShapeType::HK_SHAPE_USER_SPU_COLLECTION_0 and add support for user shape collections on SPU.** The hkpShapeType HK_SHAPE_USER_SPU_COLLECTION_0 has been removed. To run custom hkpShapeCollection implementations on the SPU now, users can rebuild the elf and change the function pointers in hkpShapeCollect::s_ShapeCollectionFunctions[COLLECTION_USER] to point to their own functions.	5.5.1 RC1
HVK-4093	Implemented	**hkExtendedMeshShape::ShapesSubpart is too big - 128bytes. It can be shaved down to 64.** The size of hkExtendedMeshShape::ShapesSubpart has been reduced to 64 bytes.	6.0.0 RC1
HVK-2455	Implemented	**Test for identical vertices in hkCapsuleShape::setVertex** Capsule shape now has extra asserts to test for invalid input data.	6.0.0 Beta
HVK-3645	Implemented	**hkpTriSampledHeightFieldCollection has a getRadius method but no setRadius** We added hkpTriSampledHeightFieldCollection::setRadius.	6.0.0 RC1
HVK-3852	Implemented	**Improve asserts that check ray casts are inside the broadphase** Previously we would assert if the raycast went outside the broadphase. We changed the assert to a warning, and added a tolerance to the check.	6.0.0 Beta
HVK-4562	Implemented	**Better support for custom hkpBvTreeShapes on SPU** We refactored hkpBvTreeShape to have its own dispatch table for queryAabb functions. Users can rebuild the elf and change the function pointers in hkpBvTreeShape::s_bvTreeFunctions[BVTREE_USER] to point to their own functions. One of the overloaded queryAabb methods has been renamed to queryAabbImpl - see hkpBvTreeShape.h.	6.0.0 Beta

Interface Change

HVK-4596	Implemented	**Improve toggling of 32-bit broadphase.** Previously, in order to enable the 32-bit broadphase, the user would have to use the HK_ENABLE_32BIT_BROADPHASE macro. Now they should create an hkpBroadPhase::Enable32BitBroadphase before creating the world. See hkpBroadPhase.h for more information.	6.0.0 RC1

Interface Change

Demos

HVK-4603	Implemented	**Add a demo showing how to create scaled instances of meshes.** A demo has been added demomstrating how to create scaled instances of meshes (hkpExtendedMeshShapes) without duplicating the collision data (vertex data and MOPP code). The demo uses existing Havok classes, so the resulting shapes work without any changes to the serialization system, and can be simulated on the SPU without any additional work. Please see the MoppInstancing demo for details.	6.0.0 Beta

Optimizations

HVK-4567	Implemented	**Remove redundant call to hkpListShape::recalcAabbExtents in its constructor** hkpListShape's constructor now only calls recalcAabbExtents once.	6.0.0 Beta

hkdynamics

Bugs

HVK-4539	Fixed	**Assert 0xf056d145 hkLs_localizedSolveToi** hkpContinuousSimluation::simulateToi() could crash if the hkpPredGskCylinderAgent3 was registered. This is fixed now.	5.5.1 RC1
HVK-4536	Fixed	**Child shape AABB's deallocated when rigid body is loaded from packfile** When serializing hkpRigidBodies with compound shapes, some unused data was stored for the contained hkpCollidable::BoundingVolumeData::m_childShapeAabbs. This is not serialized anymore, and a warning is triggered when such data is found upon load.	6.0.0 Beta

HVK-4612 Fixed

Crash in hkpWorldOperationUtil::splitSimulationIsland() when critical operations are performed in callbacks.

5.5.1 RC1

We could get a crash in the hkpWorldOperationUtil::splitSimulationIsland() function, with hkUnionFind::addEdge() at the top of the callstack. This was caused by some critical operations being performed before island-merge requests were processed. The critical-operation queue was updated, and island-merge request are now always processed before any other operations. This fixes the bug.

HVK-4630 Fixed

The engine would crash with stepJacobians() at the top of the callstack.

5.5.1 RC1

The engine crashed on PlayStation3 when collision detection was performed on SPUs. The bug only happened when contacts with 2 or more extended user datas per point were collided on SPU. This could happend e.g. when colliding two rigid-body character proxies, or destructible bodies with breakOffUtil attached. This is fixed now.

HVK-4546 Fixed

Velocities not exported for bodies that were processed in a TOI but were not activated (in a TOI sense)

5.5.1 RC1

The bug showed when we collided 3 bodies in one TOI. The original TOI-event bodies would react properly, but the third body (only processed by the TOI solver) might seem to stay frozen/fixed/unaffected if it was not activated in the TOI sense. This is a very very rarely noticable bug. It is fixed now.

This results in a small difference in response in multi-body TOI collisions.

HVK-4543

HVK-4631 Fixed

hkpSimpleContactConstraintAtom::m_maxNumContactPoints is lesser than the actual max value when using extended user datas.

6.0.0 RC1

When using extended user data for contact points, the max number of contact points in a contact was reduced by too much, resulting in atoms being much smaller than their limit was. This if fixed now.

New Features

HVK-4481 Implemented

Disabling constraints has a better API - added hkpConstraintInstance::enable() / disable()

6.0.0 RC1

Some new functions were added to hkpConstraintInstance to make enabling and disabling constraints easier. See the reference manual for hkpConstraintInstance::enable(), disable(), setEnabled(bool), isEnabled().

Interface Change

Improvements

HVK-3878	Implemented	**checkPerformance() should return a boolean** hkpRigidBody::checkPerformance() and hkpCollidable::checkPerformance() now return a hkBool (where they used to return void). This allows them to be used for example during asset pipelining.	6.0.0 Beta
HVK-4486	Implemented	**Mass modifier has a better API - added hkpConstraintInstance::setVirtualMassInverse(float, float)** A new function was added to hkpConstraintInstance to make setting the virtual mass of a constraint easier. See the reference manual for hkpConstraintInstance::setVirtualMassInverse(). A new demo 'ConstraintVirtualMassDemo' was added.	6.0.0 RC1

Interface Change

HVK-4548	Implemented	**hkpConstraintInstance::getData() returns a const pointer to hkpConstraintData now** hkpConstraintInstance::getData() returns a const pointer to hkpConstraintData now. A new function getDataRw() has been added to get a non-const pointer.	6.0.0 Beta

Interface Change

Multithreading

Bugs

HVK-4649	Fixed	**hkpWorldLinearCastJob constructor checks hkpListShape::ChildInfo::m_shapeSize even when it doesn't need to** hkpWorldLinearCastJob now only checks for non-zero hkpListShape::ChildInfo::m_shapeSize on platforms where it needs to (i.e. Playstation(R)3).	6.0.0 RC1

Improvements

HVK-3313	Implemented	**Allow changing of monitor stream size in hkMultithreadingUtil without recompiling** The hkJobThreadPool classes, which replace the hkMultithreadingUtil, allow you to specify the buffer size (which defaults to 0) for monitors on construction.	6.0.0 Beta

HVK-3436	Implemented	**Need an option to set CreateThread stacksize** The thread stack size can now be set via hkCpuJobThreadPoolCinfo::m_stackSize. This defaults to the old values: 0 on Windows and Xbox 360, 256K on Playstation(R)3.	6.0.0 Beta

Platform specific

Bugs

HVK-4616	Fixed	**hkpAgentNnEntry::m_agentIndexOnCollidable out of hkpLinkedCollidable::m_collisionEntries bounds** This was a symptom of corruption in collision structures. The corruption was caused by unguarded access to entities in the fixed island by multiple threads in the split-island job. This is fixed now.	5.5.1 RC1

Improvements

HVK-4578	Implemented	**SPU shape registration should be more foolproof.** All SPU compatible shape types are now explicitly registered on the SPU. hkpShape::setShapeFunctions now take a function pointer as an argument, so that the ShapeFuncs structure doesn't get reused for multiple shapes.	6.0.0 RC1

PS3 Optimization

Bugs

HVK-4647	Fixed	**hkpWorldGetClosestPointsJob doesn't handle the collision filter correctly on SPU** hkpWorldGetClosestPointsJob could crash when processing an hkpListShape on the SPU, because the collision filter wasn't set up correctly. This has been fixed.	6.0.0 RC1

Serialization

Bugs

HVK-4644	Fixed	**hkpStiffSpringChainData finish constructor does not call finish constructor on m_infos** This has been fixed.	6.0.0 RC1

Tool Chain

Bugs

HVK-4610	Fixed	**Splitting rigid bodies into systems places the fixed rigid bodies in the unconstrained fixed system without taking into consideration the constraints.** A bug has been fixed where hkpPhysicsData::splitPhysicsSystems() would place constrained fixed rigid bodies in the system with the unconstrained fixed rigid bodies rather than grouping them with the other rigid bodies to which they are constrained.	6.0.0 Beta

Utilities

Improvements

HVK-4109	Implemented	**There should be an asynchronous version of applyHardKeyframe.** An asynchronous version of applyHardKeyframe has been added.	6.0.0 Beta