DI-Guy 13.1 Release Notes

This document provides the following release-specific information:

Document ID: DIG-13.1-8-150903

Systems Supported

Table 1 lists the platforms currently supported by MÄK DI-Guy 13.1. Application code must be built with the indicated compilers in order to link to DI-Guy libraries.

Table 1: Platforms supported

Platform	Compiler
Red Hat Enterprise Linux Workstation 6.0	Default compiler (64-bit)
PC with Vista/Windows 7/Windows 8	Microsoft Visual C++ 10.0 (64 bit) Microsoft Visual C++ 12.0 (64 bit)

You must be an administrator to install MÄK products on Windows Vista.

Renderer Support

DI-Guy 13.1 supports the following renderers:

- Windows 64 bit: OpenGL, OpenGL Graphics API, OSG Graphics API
- Linux: OpenGL, OpenGL Graphics API, OSG.

License Manager

To run the licensed version of DI-Guy, you must install license management software. DI-Guy 13.1 uses FLEXlm 11.13.0.2. If you are upgrading from a version of the DI-Guy that used an older version of FLEXlm, you must upgrade your license management files. You do not need a new license. Licenses are forward compatible.

The License Manager files are not part of the DI-Guy installer. You can download them at:

- Windows:
  ftp://ftp.mak.com/out/MAKLicenseManager-win-setup.exe (for 32-bit systems)
  ftp://ftp.mak.com/out/MAKLicenseManager-win64-setup.exe
- Linux: ftp://ftp.mak.com/out/MAKLicenseManager-linux-setup.tar.gz (for 32-bit systems)
  ftp://ftp.mak.com/out/MAKLicenseManager-linux64-setup.tar.gz

A license manager support is available at:

http://www.mak.com/support/license-support.html

Network Compatibility

HLA only

DI-Guy 13.1 was built against VR-Link 5.1.3 and is compliant with:

- RPR-FOM 1.0 and a subset of 2.0 (draft 6, 14, 17, and 20).
- MÄK RTI 4.1 or later .

DIS only

DI-Guy 13.1 supports DIS 4, 5, 6, and 7.

RTI Support

We recommend using the latest version of the MÄK RTI. However, DI-Guy should work with other RTIs that conform to the HLA 1.3 specification, the IEEE 1516 SISO DLC API or HLA Evolved and are built with the same compiler as DI-Guy.

When using the MÄK RTI, remember to make sure that DI-Guy can find your FED file, and optional *rid.mtl,* file by either putting them in the directory from which you are running, or by setting the environment variable to the directory that contains them.

FOM Support

DI-Guy 13.1 has built-in support for versions 0.5, 0.7, 0.8, 1.0, and 2.0, drafts 6, 14, 17, and 20, of the RPR FOM. It also supports VR-Link's ability to support alternative FOMs through the FOM Mapper. By default, DI-Guy 13.1 uses RPR FOM 1.0. It is recommended that you use the DI-Guy FOM Extensions. DI-Guy includes two custom FOMs, one for HLA 1.3 and HLA 1516, and one for HLA Evolved.

New Features and Updates

DI-Guy 13.1 has the following updated features:

- Improved support for collada mesh data exported from Maya.
- The Character Viewer now includes an appearance editing component that allows for real time editing of appearance and shape set configuration data.
- Support for Microsoft Visual C++ 2013 (VC 12).
- The VC 12 version uses Qt 4.8.7.
- Updated license manager software uses FLEXlm 11.13.0.2.

You must update your license management software to the latest version of MÄK License Manager. You do not have to update your license file if your maintenance agreement covers this release.

- New aircraft guide example scenario. Interactive scenario including different air vehicles and a missile launcher. Also demonstrates different ways to use camera. For details, please see Section 3.9.4, "Aircraft Guide," in *DI-Guy AI Users Guide*.
- New content.

  New Male Heads: 6 Caucasian, 1 Indian

  New Bodies:

  – sk_refinery_workers_1 & 2
  – sk_male_casuals_15 & 16
  – sk_female_child_1 (head)
  – sk_fire_fighters_1 & 2

  New Vehicles:

  – fire_truck_1 & 2
  – hmmwv_m1114_des_uah_1
  – safir_jeep
  – uh1y_rotors
  – toyota_tundra
  – sa_6
  – sa_13

  New Props, Animals and Sets:

  – bazaar_markets_1 – 3
  – sk_sheep_1 – 3
  – oil_refinery_1 & 2

– fuel_station

For a photo gallery, please see "New Content," on page 10.

Camera

- New camera control scheme, uses click-and-drag operations for pan, zoom, and rotate.
- Camera system now organized around base position, base orientation, position offset, and orientation offset. There is lots of flexibility in how these are applied. See class-level notes in diguyViewCamera.
- Camera can still dolly and track as it did before. Setting a dolly character causes base position and base orientation to be overridden.
- Behavior of camera in dollied state is now mathematically correct and consistent. If, for example, a camera is attached to the wing of an aircraft, it remains in a consistent place relative to the aircraft, no matter how it turns, pitches, and rolls.
- Restored camera behavior that existed in DI-Guy 12.5. In this version, when both a dolly and track character were set, the camera would move behind the dolly character to keep both in view. This behavior is now called "special track mode", but is turned off by default.

Particles/Appearance Effects:

- Any particle system in the current scenario can now be used as an appearance effect without modifying any configuration files.
- Appearance Effects can now be specified as part of appearance configuration data.
- Appearance Effects lists in decision beads are no longer hard coded.
- Removed legacy Appearance Effects system. Had been unused since DI-Guy 10.x.
- Removed independent particle system DLL; code moved into *libdiguy*.
- Added support for the rendering of particle systems to the graphics API. See *diguyParticleSystem* and *diguyOglParticleSystemRenderer*.
- Added support to the graphics API retrieving a list of active light sources, OpenGL simple_playback demonstrates.
- Added support for the mapping of DI-Guy particle systems to an external particle system renderer. See *diguyParticleSystem* for details.
- Particle system engine update is now multi-threaded.
- Added support for parsing lights from Collada DAE. Headlights and taillights are automatically hooked up to appropriate networking bits based on names.
- Particles associated with characters that are flagged as invisible won't draw.

Vehicle Improvements

- Added code for damaged and destroyed appearance functionality. Will now auto switch if character is killed, or if diguyCharacter::set_DIS_damage_state() is called. Will go over network, and secondary appearance should load at the same time as initial appearance. Added diguyCharacter::get_damage_state_appearance_name().
- Vehicle ground clamping system has been improved, uses 4 intersection points and avoids some ugly edge cases.
- Vehicles without articulated skeletons (that is, vehicles not vehicle_09s) ground footprint is calculated better.
- Vehicles with articulated skeletons can now be used as vehicles, wheels will rotate automatically. This uses a new physics simulation mode DIGUY_VEHICLE_SIMULATION_MODE_WHEELS.
- Vehicles can have up to 12 wheels that will automatically rotate. (Useful for tanks.)
- The vehicle_wheels character has been deprecated.

Performance Improvements

- Improved performance of link updates, animation variables are now required to have quaternion layouts (needs rw), allows simplifies code path through our update infrastructure. Custom character types may need to be updated.
- Removed recursive update of links - switched to a simpler linear array traversal.
- 4x4 Matrix multiplies now use SSE to speed up math.
- Multi-threaded update pipeline can now use the Intel Threaded Building Blocks parallel scheduler appears to 10% faster then Qt parallel for system. Currently we do not provide controls for how many threads it uses and we are using it's recommended defaults.
- Minimizing updates for culled characters: we've added new logic to DI-Guy to delay updates of culled characters, characters that are drawn the previous frame will get link matrix and shader matrices updates inside the multi-threaded pipeline, otherwise the update of links, shapes and shader matrices will be done on a as needed basis, for example, when character is being drawn or code calls diguyCharacter::get_link_position(). This guarantees that characters who are aiming or gazing at off screen entities still do the right thing. This provides massive speed boost for folks that are completely off screen without degrading the quality of the simulation. The following functions will need to be called to enable this system diguyScenario::pre_draw_cull_state_reset() diguyScenario::post_draw_cull_state_save() and in a retained mode render: diguyCharacter::set_is_culled(0)
- OSG reference sample has a new way of sending uniform arrays that is much faster.

Lighting/Shader Improvements

- Added support for emissivity maps as the alpha channel of spec/glos/ao maps. These default to on, and for vehicles will respond automatically to the power plant dis flag.

To enable this functionality you must update how emissivity is sent to the graphics card. It's now done on a per character basis, not per material.

- Point and emissivity lights are now balanced against the amount of ambient light in the primary directional light in the shader. This lowers blow out in daytime, but still has the desired impact at night.

Connection Point System

- New connection point system, allows adding extra locations to a model without any runtime cost. Connection points can be added to appearances or shape sets. Connection points can be marked with metadata (see *diguyConnectionPointType*) to help locate things like vehicle/prop entry and exit points. All functions that take link names can now also take connection point names.
- New Connection Point functions:
  - diguyCharacter::get_num_connection_points()
  - diguyCharacter::get_connection_point_name_at_index()
  - diguyCharacter::get_num_connection_points_with_type()
  - diguyCharacter::get_connection_point_name_with_type().

Networking

- Upgraded to VR-Link 5.1.3a.
- HLA Evolved supports new interactions: AI messaging, appearance effects, gestures, and reset command.
- Reflected characters are properly culled/unculled when they leave/enter the space being simulated locally.
- In HLA Evolved, networked characters are referenced by VR-Link's global ID string. In practice, this will contain a value such as 66_77_soldier07-0.
- Attempt is made to choose version-specific FOM file when connecting DI-Guy instances over the network. Will be named *DIGuyModule_1310.xml*. Also, the search for available FOM files covers both the DIGUY_CUSTOM folder and the base config folder.
- HLA 1516e FOM Mapper support: FOM Mapper DLLs may be placed in the networking folder alongside the associated FOM file. The FOM Mapper DLL name must match the FOM filename (excluding file extension, for example, *MyFOM.xml* and *MyFOM.dll*).

- Added Federate Name and Federate Type fields to network properties. These may be set from the Network Settings window, or via API calls to diguyNetInterface::set_federate_name/set_federate_type. Federate Name replaces the auto-generated federate name based on site/host ID.
- Added Aggregate Publishing: Crowds now have the option (default FALSE) to publish as an aggregate entity.
- Improvements to sending and receiving articulated vehicles and steering.
- Added basic support for visualizing laser designators.

API Changes

- Improved UID generation supports a larger number of characters with less performance impact.
- New input mode: DIGUY_SCENARIO_INPUT_MODE_POINT_SELECT_WITH_CAMERA.
- Improved RenderGizmo:draw_wedge_with_normal drawing of small-scale wedges.
- Improved calculation for shadow render frustum so that the edge of the shadow volume is not so easily visible in scene.
- Improved blocked-cell calculations for more reliable nav mesh generation.
- Procedurally created links from DAE models will now get properly destroyed when appearance is changed.
- More work on supporting alternative coordinate systems via diguyCharacter::set_local_to_world_orientation_matrix().
- Adding support for general emissive textures in the alpha channel of spec/gloss/ao map.
- Minimized the number of times shape build and unbuild is run.
- Reference shaders do a nicer job balancing emissive and point lights brightness with general scene ambient.
- Added bdiScenarioPluginCharacterFunction, this is now used for the plug-in mid_update function instead of bdiScenarioPluginFunction this is potentially called by multiple threads simultaneously.
- Update Loop changes:
  - diguyScenario core update loop has been cleaned up to try and yield more stable results when long dt's longer then the simulation frame rate are presented.
  - The diguyScenario::set_max_ticks_behind_until_drop() default value is now 0 in the DI-Guy SDK, In DI-Guy Scenario it is 2, this keeps our application deterministic but the general SDK will try to allow a bit less perfection in scenario playback in exchange for better performance.
- Many new and deprecated functions. For a complete list, see the change log in DI-Guy SDK Documentation: *./doc/diguy_sdk_reference/doxygen/changelog.html*

Graphics API Changes

- Reference code now uses memcopy to set up low level animation matrices in *diguyOsgGraphicsShape.h*.
- Modified examples to use diguyScenario::pre_draw_cull_state_reset() diguyScenario::post_draw_cull_state_save() and diguyCharacter::set_is_culled(0), hooks into new optimization system that limits updates on culled characters.

New Content

This section illustrates some of the new content in DI-Guy.

Figure 1-1. Vehicles

Figure 1-2. People and animals

Figure 1-3. Props

Figure 1-4. Structures

Bug Fixes

This release fixes the following problems:

- Fixed a bug in diguy_graphics_get_osg_defaults() where transpose_shader_matrices was incorrectly set to 0.
- Fixed soldier aiming jitters when targeting a point close to the gun muzzle.
- Fixed issue where diguyScenario::destroy_character did not always remove character from groups immediately.
- Many bug fixes to appearance effects system.
- Fixed a number of bugs with stopping appearance effects that had been specialized for specific character types.
- A single meshes with more then 65k verts should be handled better. Was blowing out ushort index buffer, now converted to uint.

Known Problems

DI-Guy 13.1 has the following known problems:

- The first time you load the IGuy_Interaction scenario, the terrain may disappear at certain camera angles. If you reload the scenario, the problem is no longer present.
- There are Z buffering issues when running the OGL in Linux.